TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

ACPR : *Adjacent Channel Power Ratio*
: Taux de Puissance du Canal Adjacent

ADC : *Analog to Digital Converter*
: Convertisseur Analogique Numérique

AM : *Amplitude Modulation*
: Modulation d'Amplitude

AM/AM : *Amplitude-Amplitude Conversion*
: Conversion Amplitude-Amplitude

AM/PM : *Amplitude-Phase Conversion*
: Conversion Amplitude-Phase

AP : *Amplificateur de Puissance*

ASIC : *Application Specific Integrated Circuit*
: Circuit intégré à application spécifique

B : *Bandwidth*
: Largeur de Bande

BER : *Bit Error Rate*
: Taux d'Erreurs sur les Bits

CAN : *Convertisseur Analogique Numérique*

CDMA : *Code Division Multiple Access*
: Accès Multiple par Répartition en Code

CNA : *Convertisseur Numérique Analogique*

CNR : *Carrier to Noise Ratio*
: Rapport Porteuse sur Bruit

CPFSK : *Continuous-Phase Frequency-Shift Keying*
: Modulation à Déplacement de Fréquence avec Continuité de Phase

CVTDNN : *Complex-Valued Time-Delay Neural Network*
: Réseau de Neurone à Valeurs Complexes Avec Retards

CVNN : *Complex-Valued Neural Network*
: Réseau de Neurone à Valeurs Complexes

DAC : *Digital to Analog Convertor*
: Convertisseur Numérique Analogique

DPD : *Digital Predistorter*
: Préditorteur Numérique

DSB-SC-AM : *Double Side Band-Suppressed Carrier-Amplitude Modulation*
: Modulation d'Amplitude à Double Bande Latérale avec Onde Porteuse Supprimée

DSP : *Digital Signal Processor*
: Processeur de Signaux Numériques

EDGE : *Enhanced Data Rates For GSM Evolution*
: GSM à Débit Amélioré

EER : *Envelope Elimination and Restauration*
: Elimination et Restauration d'Enveloppe

EVM : *Error Vector Magnitude*
: Amplitude du Vecteur d'Erreur

FET : *Field Effect Transistor*
: Transistor à Effet de Champ

FIR : *Finite Impulse Response*
: Réponse Impulsionnelle Finie

FPGA	: *Field Programmable Gate Array*
	: Réseau Pré-diffusé Programmable par l'Utilisateur
GMSK	: *Gaussian Minimum Shift Keying*
	: Modulation à Déplacement Minimal
	avec Filtrage Gaussien
GSM	: *Global System for Mobile Communications*
	: Système Mondial de Communications Mobiles
HBTs	: *Heterojunction Bipolar Transistors*
	: Transistors Bipolaires Hétérojonction
ISI	: *Inter-Symbol Interference*
	: Interférence Inter-Symbole
LTI	: *Linear Time Invariant*
	: Linéaire Invariant dans le Temps
LMS	: *Least Mean Square*
	: Le petit Moyen Carré
LVDS	: *Low Voltage Differential Signaling*
	: Signalisation Différentielle à Basse Tension
MIMO	: *Multiple Input Multiple Output*
	: Multi-Entrées Multi-Sorties
Mod./Dem.	: *Modulator/Demodulator*
	: Modulateur/Démodulateur
MOS	: *Metal-Oxide Semiconductor*
	:Semi Conducteur à Métal Oxidé
MLP	: *Multi Layer Perceptron*
	: Perceptron Multi-Couches
MSE	: *Mean Square Error*
	: Erreur Quadratique Moyenne
MSK	: *Minimum Shift Keying*
	: Modulation par Déplacement de Phase Minimum

M-QAM : *M-Array Quadrature Amplitude Modulation*
: Modulation Quadrature d'Amplitude d'ordre M

Nat-GD : *Natural Gradient Descent*
: Descente de Gradient Naturel

NN : *Neural Network*
: Réseaux de Neurones

OGD : *Ordinary Gradient Descent*
: Descente de Gradient Ordinaire

OFDM : *Orthogonally Frequency Division Multiplexing*
: Multiplexage par Répartition Orthogonale de la Fréquence

PA : *Power Amplifier*
: Amplificateur de puissance

PAPR : *Peak to Average Power Ratio*
: Rapport Puissance Crête sur Puissance Moyenne

PDC : *Personal Digital Cellular*
: Cellulaire Digital Personnel

PSD : *Power Spectral Density*
: Densité Spectrale de Puissance

QAM : *Quadrature Amplitude Modulation*
: Modulation d'Amplitude en Quadrature

QPSK : *Quadrature Phase Shift Keying*
: Modulation par Déplacement de Phase en Quadrature

RF : *Radio frequency*
: Fréquence Radio

ROM : *Read Only Memory*
: Mémoire à Lecture Seule

RVNN : *Real-Valued Neural Network*
: Réseaux de Neurones à Valeurs Réelles

RVTDNN : *Real-Valued Time-Delay Neural Network*
: Réseaux de Neurones à Valeurs Réelles
avec des Retards

SSPA : *Solide State Power Amplifier*
: Amplificateur de Puissance à État Solide

TD : *Total Degradation*
: Dégradation Totale

TWT : *Travelling-Wave Tube*
: Tube à Ondes Progressives

UMTS : *Universal Mobile Telecommunication System*
: Système Universel de Télécommunications avec les Mobiles

UWB : *Ultra Wide Band*
: Ultra Large Bande

VHDL : *Very High Speed Integrated Circuit*
Hardware Description Language
: Langage de Description des Circuits
Intégrés Matériels Très Rapides

WCDMA : *Wide Band Code Division Multiple Access*
: Accès Multiple par Répartition en Code à Large Bande

XSG : *Xilinx System Generator*
: Système Générateur de Xilinx

2D : *Two Dimensional*
: Deux Dimensions

3G : *Third Generation*
: Troisième Génération

$\pi/4$ DQPSK : *$\pi/4$ Differential Quadrature Phase Shift Keying*
: $\pi/4$ Modulation par Déplacement de Phase
en Quadrature Différentielle

CHAPITRE I

INTRODUCTION

1.1 Généralités

Le grand défi des systèmes modernes de communication sans fil est de transmettre, correctement et rapidement, de grandes quantités de données (par exemple des fichiers de données numériques, musique, vidéo, photos, etc.). L'émergence de ces systèmes provoque une pénurie de la bande utile des fréquences, qui se voit en expansion continue et qui nécessite forcément une exploitation judicieuse. Le respect des normes et standards de communications sans fil en vigueur est obligatoire afin de maximiser le nombre d'abonnés en service sans avoir recours à d'autres bandes de fréquence, qui nécessitent forcément des dispositifs micro-ondes et radio performants et plus coûteux (figure 1.1). Les techniques de modulation numérique linéaire (`M-QAM`, `QPSK`, etc.)[14] et d'accès multiple (`CDMA`, `WCDMA`, `CDMA2000`, `OFDM`, etc.) [14] sont utilisées dans les nouvelles générations de systèmes de communications, car elles sont plus appropriées pour réduire l'occupation spectrale (c.-à-d. caractérisées par une meilleure efficacité spectrale) et satisfaire les besoins en débits de communication. Pour une modulation linéaire M-array QAM, l'efficacité spectrale est définie comme étant $\Gamma = \frac{log_2 M}{1+\beta}$ exprimée en bit/s Hz^{-1}, où β est le facteur de décroissance (*roll-off factor*) du filtre de mise en forme [47] et M est

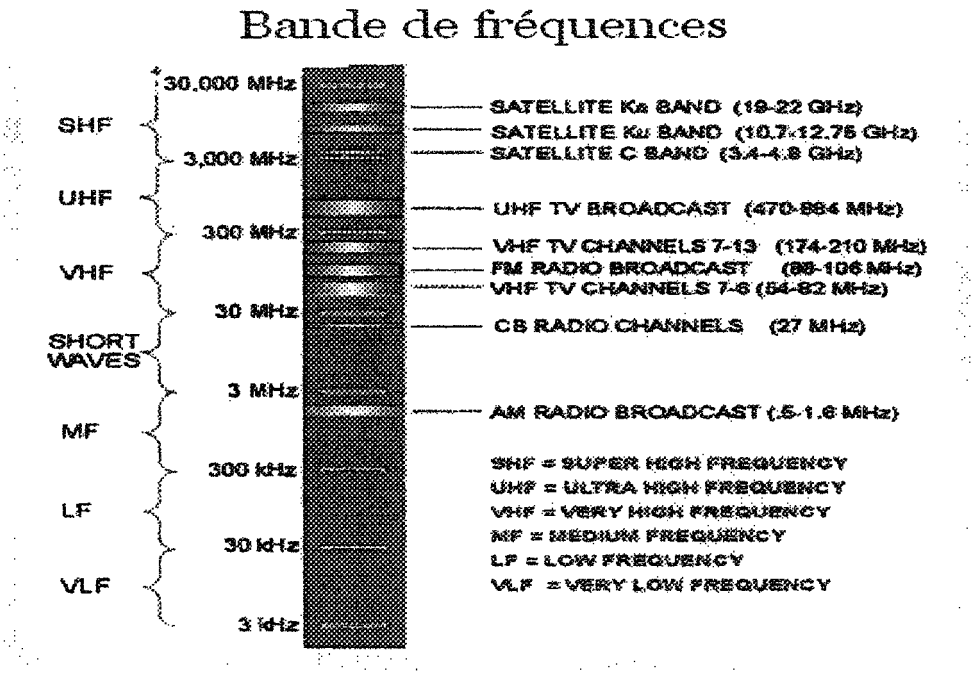


Figure 1.1: Répartition des bandes de fréquences des applications de communication sans fil [2].

le nombre de symbole. Cette efficacité est proportionnelle au nombre de bits par symbole m qui est défini par : $m = log_2(M)$. Par contre, ces méthodes de modulation engendrent des signaux à enveloppes variables avec un rapport puissance crête sur puissance moyenne PAPR considérable, p. ex., 9,70 dB pour la modulation CDMA2000[41] et 5,8 dB pour 16-QAM[46]. Si $r(t)$ est l'enveloppe modulée, le paramètre PAPR peut être calculé selon l'équation suivante :

$$PAPR = \frac{\max(|r(t)|^2)}{E(|r(t)|^2)} \tag{1.1}$$

où E(.) représente l'espérance mathématique.

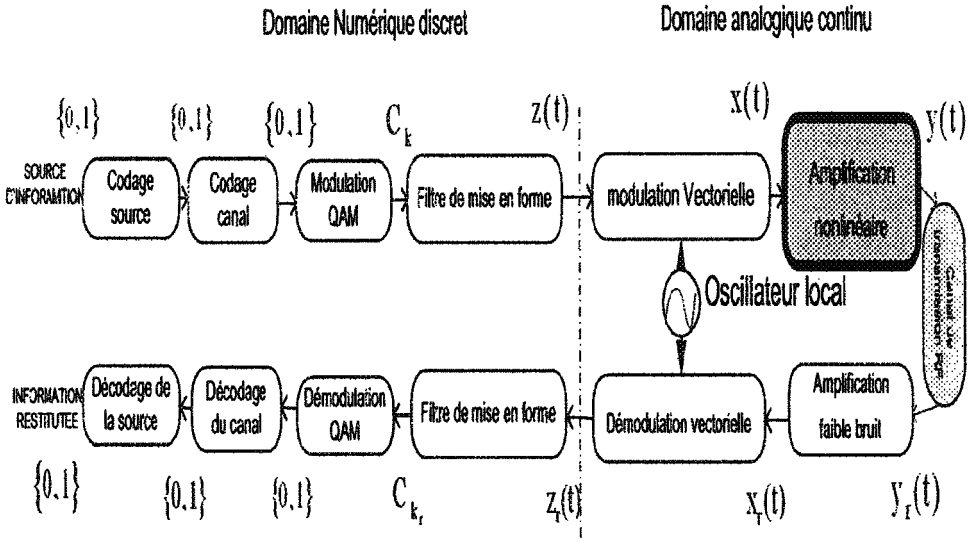


Figure 1.2: Parcours typique d'un signal dans une chaîne de communication sans fil numérique.

Le signal de la porteuse aura une puissance variable en fonction de la dynamique de l'enveloppe. C'est une forme d'onde qui présente de nouvelles contraintes par rapport à celles existantes dans les techniques de modulation non-linéaires classiques, principalement caractérisées par une faible efficacité spectrale.

Une des principales contraintes qui entrave l'utilisation et le développement de ces méthodes de modulation réside dans la non-linéarité de la trajectoire parcourue par le signal de fréquence radio (RF) (figure 1.2). Par analyse des éléments constituant le parcours du signal, il est possible de remarquer qu'à l'entrée du récepteur, le niveau du signal $y_r(t)$ est très faibe et ne peut pas introduire d'effets indésirables. Le canal RF est naturellement linéaire avec la présence d'effets dispersifs (dispersion du flux électromagnétique) et de trajets multiples qui peuvent être corrigés par le biais d'un égaliseur [25].

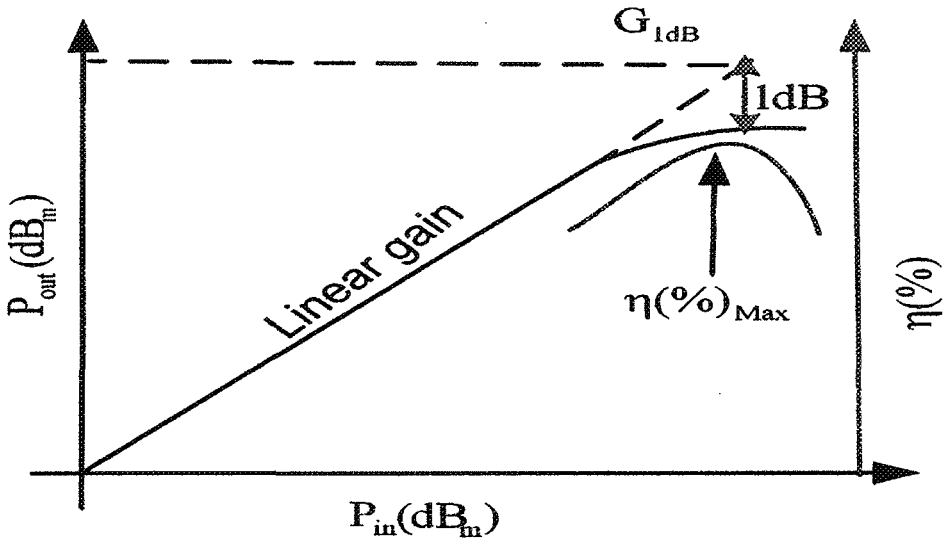


Figure 1.3: Puissance de sortie et rendement de l'élément AP en fonction du niveau de la puissance d'entrée [36].

En réalité, le seul élément de cette chaîne qui peut présenter des non-linéarités importantes est l'amplificateur de puissance (AP) qui réalise l'amplification du signal avant son envoi dans le canal de transmission. L'efficacité énergétique et la linéarité de cet élément amplificateur sont deux caractéristiques évoluant dans des sens opposés. Autrement dit, l'augmentation de l'efficacité énergétique entraîne une dégradation de la linéarité et vice versa. La relation entre ces deux paramètres et la puissance d'entrée est présentée à la figure 1.3, c'est une caractéristique physique typique qui peut être vérifiée expérimentalement. Elle a la même forme pour toutes les classes d'élément AP, sauf qu'elle change de degré de non-linéarité selon les classes.

Il est remarqué qu'en augmentant le niveau de la puissance d'entrée P_{in}, le rendement η atteint sa valeur maximale au voisinage du point où la linéarité de la caractéristique commence à se dégrader. Il est à noter que le rendement

de l'élément AP est défini par : $\eta = P_{out}/P_{dc}$, où P_{out} est la puissance de sortie et P_{dc} est la puissance absorbée à partir de la source continue.

Ainsi, l'élément le plus critique (le circuit AP) n'est pas capable de jumeler la possibilité d'amplifier fidèlement des signaux à enveloppe variable en travaillant le plus proche de sa zone de saturation (compression). Par conséquent, il est constaté que l'amplification du signal est accompagnée de remontées spectrales en dehors du canal utile (c.-à-d. ACPR), et de déformations de constellation des signaux modulés (c.-à-d. EVM)[36], ceci est du aux non-linéarités de cet élément. Le recul du niveau de puissance du signal d'entrée de quelques décibels («Back-off») est une approche simple à intégrer dans le système, sauf qu'elle réduit considérablement l'efficacité énergétique.

En résumé, un système de communication sans fil doit impérativement satisfaire les exigences suivantes qui sont étroitement liées à la linéarité de l'élément AP :

1. Exploitation judicieuse de la bande de fréquence afin d'éviter une saturation et une pénurie du spectre ;
2. Maximisation de l'efficacité énergétique de chaque dispositif de la chaîne pour réduire l'encombrement de l'équipement (poids et volume) et sa consommation en puissance ;
3. Réalisation d'un système pouvant assurer la transmission de l'intégrité du signal porteur avec une faible perte d'informations et un minimum d'influence sur les systèmes émettant dans les bandes voisines.

Les circuits AP sont catégorisés, selon leurs modes de fonctionnement, en classes nommées A, B, AB, C, D, E et F. Ces classes sont définies en fonction de l'angle de conduction du transistor dans la période du signal. L'équation

qui détermine la valeur maximale du rendement du drain des transistors MOS utilisés généralement dans les systèmes de communications est la suivante [59] :

$$\eta_D(\theta) = \frac{1}{4}\frac{\theta - sin(\theta)}{sin(\frac{\theta}{2}) - \frac{\theta}{2}cos(\frac{\theta}{2})}100\% \tag{1.2}$$

où θ est l'angle de conduction. p. ex., si le circuit AP est de classe A ($\theta = 2\pi \ \ rad$), il est trouvé que le rendement aura la valeur $\eta_D = 50\%$. Les rendements maximaux théoriques et les indices relatifs de la qualité de la linéarité pour les différentes classes d'amplificateurs de puissance sont présentés dans le tableau 1.1.

Tableau 1.1: Rendement maximal théorique et indice relatif de la qualité de la linéarité pour les différentes classes d'amplificateurs de puissance.

classes de fonctionnement	rendement maximal	qualité de la linéarité
A	50%	excellente
AB	50%-78,5%	bonne
B	78,5%	faible
C	100%	très faible
D	100%	très faible
E	100%	très faible
F	100%	très faible

Selon la classe du circuits AP utilisé, il existe deux axes de recherche touchant au développement des systèmes de communications sans fil afin de satisfaire les exigences citées précédemment. Le premier se concentre sur l'utilisation des circuits AP parfaitement linéaires (classe A), cherchant par la suite des méthodes d'amélioration de l'efficacité énergétique (p. ex., Doherty, élimination et restauration de l'enveloppe «EER») [36], alors que le second s'intéresse à l'amélioration de la linéarité des circuits AP des classes fortement non-linéaires, en utilisant des techniques de linéarisation. Les méthodes de linéarisation les

plus répondues sont [36] :

- Contre réaction (*Feedback*) ;
- Pré-compensation ou Prédistorsion (*Predistortion*) ;
- Post-compensation (*Feedforward*) ;
- Amplification linéaire avec des composants non-linéaires (*LINC*) ;
- Égalisation (*Equalization*).

La compensation des effets de non-linéarité, peut être effectuée sur des signaux numériques et analogiques, avec la possibilité de l'intégrer dans l'émetteur, en amont du circuit AP, cas de la prédistorsion, ou bien dans le récepteur, cas de l'égalisation. Pour des raisons de rentabilité économique, il est clair que la prédistorsion est la meilleure approche car la linéarisation est effectuée sur la station de base ou le satellite, au lieu de procéder à une égalisation dans chaque récepteur. La prédistorsion présente plusieurs avantages dont la simplicité d'implantation, un moindre impact sur la taille du transmetteur, et l'inexistence de problème d'instabilité. Avec l'emploi des techniques de modulation large bande (p. ex., WCDMA et CDMA2000), la considération et la modélisation des effets mémoire sont incontournables. Ces effets sont principalement dus aux facteurs suivants [63, 17, 9, 19] :

- Filtres d'émission et de mise en forme ;
- Phénomènes d'auto-chauffage ou effets thermiques ;
- Circuits de polarisation électrique et fréquence du signal ;
- Changements climatiques ;
- Vieillissement du transistor et des composants RF qui l'entourent.

La prédistorsion dynamique et adaptative constitue une solution qui permet de compenser ces effets mémoire. Parmi les méthodes de prédistorsion proposées dans la littérature, il est cité, les réseaux de neurones dynamiques

[54, 39, 28], les modèles non-linéaires avec mémoire (p. ex., modèle polynomial [15, 67] et les séries de Volterra [21, 75]). Les réseaux de neurones présentent un certain nombre d'avantages par rapport aux méthodes des tableaux de correspondance multi-dimensionnels [7, 8, 38], il est possible de citer [51] :

- L'inexistance d'effets relatifs aux tableaux des gains de quantifications ;
- La non nécessité d'un algorithme pour l'indexation du tableau des gains ;
- Un petit nombre d'échantillons de données d'apprentissage du prédistorteur est requis ;
- La simplicité de leurs réalisations pratiques.

Il est à noter que la prédistorsion de données en bande de base avant le filtre de mise en forme «*Data Predistortion*», peut compenser les distorsions à l'intérieur de la bande «*In-Band distortions* (EVM)», sans pouvoir compenser convenablement celles à l'extérieur de la bande «*Out-of-band distortions* (ACPR)». En plus qu'elle nécessite une adaptation du bloc de prédistorsion à chaque fois que la technique de modulation change, elle ne peut pas opérer avec des modulations complexes, p. ex., OFDM [65, 17, 36]. Avec le développement des technologies DSPs, FPGAs et CNA/CAN, la prédistorsion peut opérer en bande de base, après le filtre de mise en forme. Par conséquent, la bande de fréquence de correction va s'étaler et les deux types de distorsions (*In-band* et *out-of-band*) sont simultanément éliminés. La modélisation des circuits AP avec les réseaux de neurones [72, 41, 42] a été développée pour permettre de reproduire, fidèlement, les comportements sévèrement non-linéaires des circuits AP, en profitant de leurs potentiels à apprendre et à généraliser le comportements de ces circuits à partir des échantillons de signaux d'entrées et de sorties mesurés [24].

Présenté sous format mémoire par article, ce document se divise en trois

chapitres. Le premier chapitre contient une révision de la littérature pertinente, l'énoncé de la problématique, les hypothèses, la méthodologie et les contributions majeures de ce travail de recherche. Rédigé en anglais sous forme d'un article scientifique, le deuxième chapitre constitue le cœur du travail. Il commence par la section introduction du système de communication à étudier. Dans une deuxième section, il est présenté le modèle de l'amplificateur adopté pour tester les performances de la méthode de linéarisation proposée. Dans une troisième section, il est validé les capacités des réseaux de neurones (RVTDNNs) à modéliser les non-linéarités des circuits AP avec le type de la modulation adoptée. Dans une quatrième section, il est exposé la nouvelle architecture de linéarisation de données en ligne (*data on-line*) proposée. La dernière section, contient une brève présentation de la méthodologie d'implantation du réseau de neurones et l'algorithme de rétro-propagation du gradient avec le logiciel XSG (*Xilinx System Generator for DSP*) ; ce chapitre est clôturé par une conclusion générale. Le troisième et dernier chapitre est une conclusion générale en français. Des propositions pour des travaux futurs sont énumérées à la fin de ce chapitre.

Les travaux basés sur les réseaux de neurones pour la prédistorsion, sont résumés dans les points qui suivent. Il est choisi un échantillon présentant des résultats de telle manière que leurs comparaisons avec les résultats du présent travail sont faciles et significatives.

1. **H. Abdulkader *et al.*** [1]

   Les auteurs proposent une méthode DPD basée sur les réseaux NN en utilisant un nouvel algorithme (Nat-GD). Cette méthode est adaptative pour un modèle du circuit AP sans mémoire. Le signal de test est issu d'une

modulation 16-QAM. Les résultats sont obtenus avec des simulations, en comparant deux algorithmes (Nat-GD, OGD). Les performances de l'algorithme Nat-GD sont démontrés, et qui donnent un gain de 10^{-4} V^2 sur le facteur MSE par rapport a l'algorithme OGD .

2. **N. Benvenuto *et al.*** [9]

   N. Benvenuto *et al.* proposent une méthode DPD basée sur les réseaux CVTDNNs. La méthode DPD est appliquée au signal avant le filtrage de mise en forme. En utilisant des signaux de test : 16-QAM et 64-QAM, un nouvel algorithme et une nouvelle architecture sont proposées pour l'apprentissage du réseau NN. Le modèle de Saleh sans mémoire pour le circuit AP est utilisé. Le Contrôle du paramètre DSP en sortie du circuit AP est réalisé. Le travail est basé sur des résultats de simulations.

3. **W. Henghui *et al.*** [26]

   La méthode DPD proposée est basée sur les réseaux NN de type MLPs à valeurs complexes, en utilisant l'algorithme LMS. Dans ce travail, la caractéristique AM/PM est considérée parfaite. Un modèle sans mémoire d'un amplificateur SSPA est utilisé. Le signal de test est issu d'une modulation 16-QAM dans un système OFDM. La méthode d'apprentissage indirect est adoptée pour obtenir les résultats avec des simulations. Ce travail ne prend pas en compte les effets mémoire et une amélioration de 2 dB est réalisée sur le paramètre TD. Cette approche est valable pour les circuits APs de type SSPA seulement.

4. **H. Hwangbo *et al.*** [28]

   Les auteurs présentent la modélisation et la méthode DPD pour la linéarisation des circuits AP. Le travail est basé sur les réseaux RVTDNNs. Les données entrée/sortie du circuit AP sont obtenues à partir d'un circuit AP phy-

sique réel. La correction par la méthode DPD est appliquée aux composantes cartésiennes en bande de base. L'architecture d'apprentissage utilisée est indirect. Les signaux de test utilisés sont issus des modulations WCDMA et CDMA2000, la largeur de bande B du signal de test est 5 MHz. La méthode de linéarisation utilisée est adaptative et prend en compte les effets mémoire. Un excellent modèle est obtenu pour le circuit AP avec la méthode proposée. Les résultats de simulations démontrent des améliorations du paramètre ACPR :

- Sans mémoire (2 - 3 dB) pour la modulation WCDMA, (moins de 10 dB) pour la modulation CDMA2000;
- Avec mémoire 15 dB pour la modulation WCDMA, 20 dB pour la modulation CDMA2000.

5. **F. Langlet *et al.*** [39]

   Les auteurs proposent une méthode DPD basée sur les réseaux MLPs à valeurs réelles. Une implémentation mixte (analogique et digitale) du réseau NN est réalisée. Les modèles avec et sans mémoire pour le circuit SSPA sont adoptés. Une comparaison des algorithmes mentionnés dans le travail de H. Abdulkader *et al.* [1] est faite. Le signal 16-QAM avec une largeur de bande de 25 MHz est utilisé pour tester l'architecture adaptative proposée. Ils ont utilisé 4 architectures différentes pour les réseaux NNs. La prédistorsion du signal en bande de base est effectuée. Les résultats présentés sont expérimentaux et démontrent un gain de 25 dB sur le paramètre MSE en utilisant l'algorithme Nat-GD. Les calculs effectués par l'algorithme d'adaptation sont complexes et longs (2×10^6 itérations).

6. **T. Liu *et al.*** [41]

   Les auteurs proposent un travail introduisant pour la première fois les réseaux RVTDNNs pour la modélisation dynamique des éléments APs avec effets mémoire, dans des conditions de signaux large bande. Le travail a été présenté une deuxième fois dans la $1^{ère}$ partie du travail de H. Hwangbo *et al.* [28].

7. **N. Naskas et Y. Papananos** [51]

   Les auteurs utilisent la méthode DPD basée sur les réseaux MLPs à valeurs complexes. L'opération de PD du signal est effectuée en bande de base. La compensation est basée sur les caractéristiques AM/AM et AM/PM. Le signal de test est issu de la modulation $\pi/4$-DQPSK avec largeur de bande de 0,5 MHz. Un modèle sans mémoire est supposé pour l'élément AP. La méthode n'est pas adaptative. Le nombre de neurones par couche respective (couche d'entrée, $1^{ère}$ couche cachée, $2^{ème}$ couche cachée, couche de sortie) est 1-9-9-2. les amélioration du paramètre ACPR avec des simulations sont :

   - nombre de pas dans le vecteur de stimulus (10 pas) : très faible;
   - nombre de pas dans le vecteur de stimulus (20 pas) : 25 dB.

   Un examen des effets des imperfections des éléments Mod./Dem. est effectué. La méthode est valable pour n'importe quel type de circuit AP. Les effets des retards de la boucle sont traités. L'algorithme d'apprentissage est complexe et ne prend pas en considération les effets mémoire.

8. **H. Qian et G. T. Zhou** [54]

   La méthode DPD proposée est basée sur les réseaux CVTDNNs. Les modèles sans et avec mémoire sont utilisés pour le circuit AP. L'algorithme d'apprentissage avec des valeurs complexe est utilisé. La méthode DPD est

appliquée au signal complexe en bande de base. Les lignes de retards dans la couche d'entrée sont utilisées pour prendre en compte les effets mémoire. l'architecture d'apprentissage adaptative indirect est employée avec un signal de test issu d'une modulation 16-QAM sur une largeur de bande de 0,4 MHz. Les amélioration du paramètre ACPR en fonction de nombre de neurones par couche sont :

- Réseau 1-5-1 pour le cas de circuit AP sans mémoire : suppression quasi-totale;
- Réseau 1-24-1 pour le cas de modèle polynomial avec mémoire : suppression quasi-totale;
- Réseau 1-25-1 pour la cas de modèle de Wiener de $2^{ème}$ ordre : suppression quasi totale;

Les résultats sont obtenues avec des simulations. L'algorithme utilisé présente des risques de convergence vers des minimums locaux et les effets des bruits sont pris en compte. La méthode est valable pour n'importe quel type de circuit AP.

9. **B. E. Watkins et R. North** [68]

Les auteurs proposent une méthode DPD adaptative basée sur deux réseaux MLPs à valeurs réelles, un pour la caractéristiques AM/AM et l'autre pour AM/PM. Un modèle sans effets mémoire est adopté pour présenter l'élément AP. La prédistorsion est effectuée sur le signal en bande de base. Le signal de test utilisé est issu d'une modulation 64-QAM. Les résultats sont obtenus par des simulations. La méthode demande moins de complexité de calculs avec des risques de convergence vers des minimums locaux et elle ne compense pas les effets mémoire à court-terme.

1.2 Problématique

Après avoir présenté les défis à relever par les systèmes modernes de communications sans fil, il est normale de lier la réalisation de leurs intégrité à l'élimination des effets indésirables engendrés par les non-linéarités des circuits AP. L'élément AP se trouve contraint d'amplifier des signaux à envelopes variables issus de modulations linéaires avec un paramètre PAPR important. Les non-linéarités et les effets mémoire provoquent des distorsions qui vont augmenter le facteur BER et au même temps provoquer des signaux parasites dans les bandes de fréquence voisines. Les stations de base de $3^{ème}$ et $4^{ème}$ génération (3G et 4G) utilisent des techniques de modulation avancées, s'étalant sur des largeurs de bande plus importantes. Par conséquent, les effets mémoire engendrent davantage d'effets indésirables, qui sont également engendrés par le circuit AP.

Dans ce travail, les efforts sont concentrés sur les moyens qui permettent de corriger efficacement les distorsions dues au circuit AP et de compenser les effets mémoire avec le même dispositif. Ce dispositif doit être simple et sans grands impacts sur la taille et le fonctionnement du transmetteur sans fil.

L'intégration du dispositif de linéarisation dans le transmetteur doit générer des corrections sur une largeur de bande considérable, donnant des débits d'information de l'ordre de ceux donnés par des modulations numériques non-linéaires. p. ex., dans le présent cas, la modulation 16-QAM peut atteindre pour un facteur de décroissance du filtre de mise en forme $\beta = 0,3$, un débit de 1 Mb/s sur une largeur de bande de 1,3 MHz. Ainsi, la compensation doit être à l'intérieur et à l'extérieur de la bande des fréquences du signal, afin de minimiser le facteur BER et réduire les effets d'interférence sur les systèmes fonction-

nant dans les bandes adjacentes. La stabilité du transmetteur et la convergence rapide de l'algorithme adopté pour la correction des non-linéarités avec données en ligne (*data on-line*) est un critère primordial. La problématique de ce travail de recherche considère l'amélioration des travaux similaires pour trouver un moyen plus simple et robuste permettant de remédier aux problèmes des effets indésirables des circuits AP.

1.3 Hypothèses

Les résultats de ce travail de recherche sont obtenus en supposant que les imperfections des modulateurs et démodulateurs, telles que le déséquilibre de l'amplitude, le déséquilibre de la phase, le déséquilibre des composantes I_p/Q_p et les fuites de l'oscillateur local [51] sont négligeables. Quoique, en pratique, elles sont corrigées automatiquement par la prédistorsion du fait qu'elles font partie de la boucle de correction. Ces imperfections vont diminuer impérativement les performances du prédistorteur. Il est aussi supposé, que les délais dans la boucle sont négligeables par rapport à la période d'échantillonnage du signal. D'autre part, il est opté pour une structure largement parallèle (aucun multiplexage n'est utilisé) pour le bloc de prédistorsion. Il est supposé que la fréquence du processeur numérique (FPGA), servant comme plate-forme d'implantation du prédistorteur, est supérieure que la fréquence d'échantillonnage du signal en bande de base. Il est supposé également que la surface du silicium est suffisante pour fournir toutes les ressources nécessaires à l'implantation des différentes parties de l'architecture neuronale telles que les mémoires, les tableaux de correspondance, les multiplicateurs et les additionneurs.

1.4 Méthodologie retenue

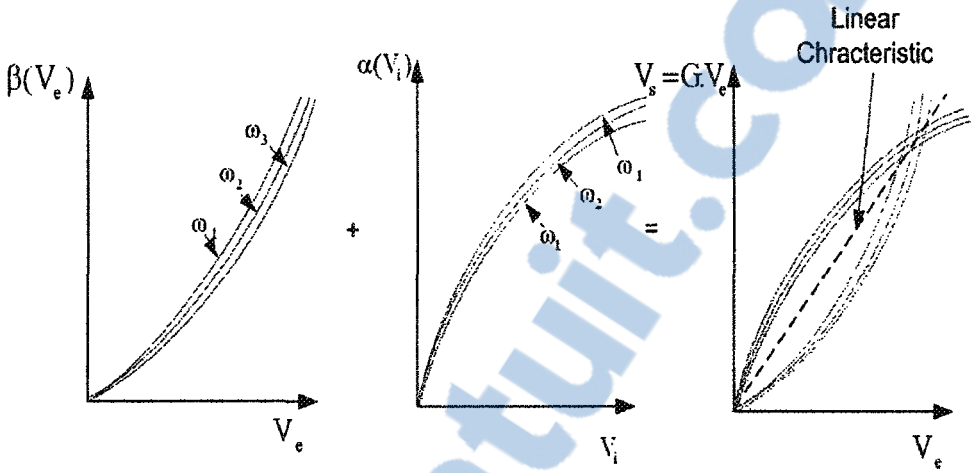


Figure 1.4: Principe général de la méthode de linéarisation par prédistorsion du signal en considérant les effets mémoire engendrés par les fréquences du signal d'entrée.

Le système de communication adopté pour l'évaluation et le test des performances de l'architecture de linéarisation proposée, est un système opérant avec la modulation linéaire `16-QAM` pour constater le vrai comportement de l'élément `AP` dans le cas de ce type de modulation. En utilisant un circuit `AP` de type `TWT` avec des effets mémoire, il est possible de modéliser le comportement non-linéaire de l'élément `AP` avec considération des effets mémoire. Le modèle de Saleh [57] est adopté pour introduire la non-linéarité sans effets mémoire et un filtre linéaire `LTI` placé en amont de l'élément `AP` est utilisé pour introduire les effets mémoire [54, 19, 5]. Un seul signal modulé est présenté à l'entrée de l'élément `AP` sans utiliser aucun multiplexage, ceci est du à la structure de l'élément de prédistorsion qui est choisie largement parallèle, pour augmenter la fréquence du travail du transmetteur.

Le principe de la technique de linéarisation par prédistorsion du signal en bande de base (*Baseband signal predistortion*) est de prédistordre le signal à la sortie du filtre de mise en forme ($h_g(t)$), avant de l'amplifier par le biais de l'élément AP non-linéaire, afin de compenser les distorsions et les déformations ultérieures [36]. Cette méthode peut être employée dans des systèmes utilisant des techniques de modulations linéaires, complexes à large bande (CDMA et OFDM), sans être obligé d'adapter la structure [65]. Cette approche se prête bien à l'intégration avec un impact faible sur la complexité et la taille du système au complet. Le schéma synoptique de la configuration de cette méthode est présenté sur la figure 1.4. Il consiste généralement en deux fonctions juxtaposées dépendantes de la fréquence du signal (ω_j) : une fonction de prédistorsion $\beta_{\omega_j}(V_e)$ et une fonction d'amplification non-linéaire $\alpha_{\omega_j}(V_i)$. La combinaison des deux fonctions donne une sortie $V_s = \alpha_{\omega_j}(\beta_{\omega_j}(V_e)) = G.V_e$ proportionnelle à l'entrée, où G est le gain linéaire désiré du circuit AP qui doit être indépendant de la fréquence du signal.

Dans le présent cas, le bloc de prédistorsion est composé d'une seule structure largement parallèle et qui ne comporte aucun multiplexgae, afin de pouvoir traiter des signaux d'entrée à grande fréquence. Cette structure est basée sur les réseaux de neurones et implantable sur des processeurs numériques (FPGAs) performant et présentant des avantages de rapidité, portabilité et moindre encombrement. En principe, le gain du prédistorteur augmente quand celui du circuit AP diminue et le déphasage prend la valeur négative de celle provoquée par l'élément AP. La mise en cascade des deux éléments peut donner naissance à une caractéristique plus linéaire. Avec la présence des effets mémoire, il est constaté une caractéristique non-linéaire pour chaque fréquence présente dans le signal d'entrée (figure 1.4). Dans ce cas, l'ensemble des courbes non-linéaires

juxtaposées nécessite d'être linéarisées en même temps en compensant les effets mémoire. Par conséquent, le bloc de prédistorsion sera impérativement de même nature, c.-à-d., avec mémoire.

Pour mettre en place cette méthode, la première étape consiste à déterminer la fonction de prédistorsion capable de mémoriser la fonction inverse du circuit AP. L'architecture neuronale des RVTDNNs, proposée par Liu *et al.* [41] pour modéliser les circuits APs avec des signaux large bande, est une approche performante capable de reproduire le fonctionnement du circuit AP en tenant compte des effets mémoire. Les réseaux de neurones RVTDNNs sont des réseaux MLPs qui ne comportent pas les inconvénients inerrant à d'autres formes de présentation de signaux d'entrée et de sortie (forme polaire ou rectangulaire) parce qu'ils traitent les données de manière séparées durant le flux de l'information dans la boucle direct et la boucle de retour. En fait, lorsque les paramètres du réseau NN (poids et biais) prennent des valeurs complexes [9, 54], ceci conduit à un algorithme d'apprentissage complexe qui présente des problèmes de convergence et un temps de calcul important [41]. Dans le cas du présent travail, la détermination de la fonction inverse de l'amplificateur est basée sur un réseau RVTDNNS de structure identique à celle proposée par Liu *et al.* [41]. Ce réseau est implanté avec le logiciel XSG dans l'environnement Matlab/Simulink et son fonctionnement est validé avec les signaux (16-QAM) pour la modélisation du circuit AP.

Pour la deuxième étape, c. à d., la linéarisation, l'idée maîtresse de l'architecture proposée provient du travail de Benvenuto *et al.* [9], dans lequel les modèles approximatifs de l'élément AP et les Mod./Dem. ont été intégrés avec le réseau NN ; l'erreur minimisée par l'algorithme de rétropropagation du gradient est calculée à la sortie démodulée du circuit AP. Dans le travail de Benvenuto *et*

al. [9], un seul bloc processeur est suffisant pour obtenir la fonction inverse de l'élément AP. Dans le présent travail de recherche, il est proposé une méthode quasi similaire à celle de Benvenuto *et al.* [9], dans laquelle il est développé une nouvelle approche pour la propagation du signal de l'erreur à travers le réseau RVTDNNs, à partir d'un point de décision situé à la sortie démodulée de l'élément AP.

1.5 Contributions apportées

En adoptant une méthode de linéarisation par préditorsion adaptative du signal en bande de base, les améliorations apportées concernent essentiellement la nouvelle architecture (figure 1.5) de linéarisation par prédistorsion adaptative du circuit AP. La compensation des effets mémoire à court et à long terme est assurée en même temps que la correction des distorsions à l'intérieur et à l'extérieur de la bande utile.

Il est adopté, l'architecture neuronale des réseaux RVTDNNs proposée par Liu *et al.* [41] car elle est capable de corriger, en même temps, les non-linéarités et de compenser les effets mémoire. Afin de pouvoir corriger les distorsions à l'intérieur et à l'extérieur de la bande, il est opté pour la prédistorsion du signal en bande de base en prédistordant le signal échantillonné après être traité par le filtre de mise en forme (*pulse shaping filter*). L'architecture adaptative proposée est capable de compenser les variations de la caractéristique de l'élément AP dues aux effets mémoire à long terme. Afin d'éviter les convergences vers des minimums locaux et de réduire la complexité de l'algorithme de convergence, l'algorithme de rétropropagation du gradient de l'erreur est choisi, qui est efficace dans le cas des réseaux NN à valeurs réelles ayant la possibilité

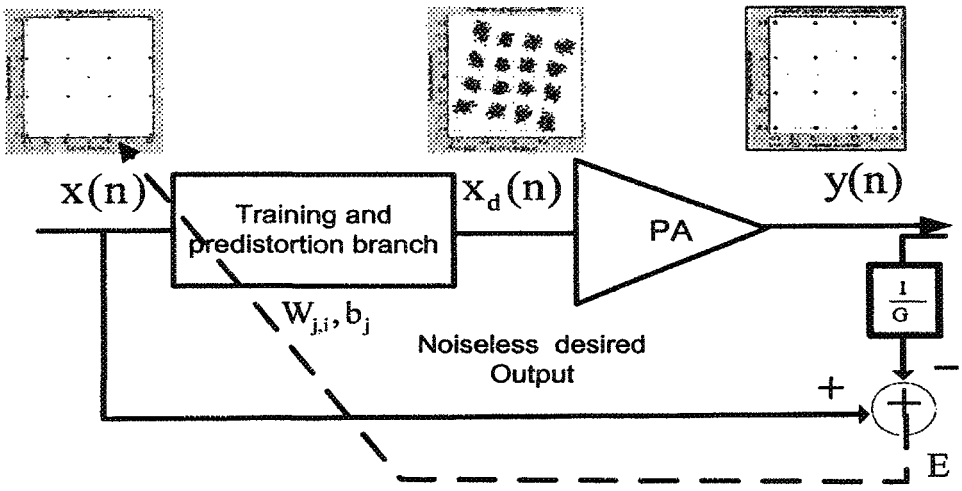


Figure 1.5: Principe l'architecture directe proposée.

d'adaptation aux fonctionnement en temps réel (données en ligne).

La contribution majeure de ce travail est l'architecture de prédistorsion directe en temps réel, avec un seul bloc processeur, qui se base sur une nouvelle approche pour la propagation de l'erreur à travers le réseau de neurones. La figure 1.5 montre le principe général de l'approche proposée. A la connaissance de l'auteur, cette approche est unique. Aussi, l'architecture est programmée avec le logiciel XSG pour la première pour des applications de ce domaine de recherche. Cet outil donne la possibilité d'implanter le réseau NN sur des circuits programmables de type FPGAs performants pour un fonctionnement en temps réel. L'approche d'implantation se base dans plusieurs cas sur le travail de Minhui [49], a partir de lequel, il est amélioré quelques blocs et proposé de nouveaux blocs (par exemple : calcul de MSE, bloc de décision). Les résultats de simulation obtenus démontrent l'efficacité de la méthode proposée et prouvent de bonnes performances réalisées sur la qualité de la linéarité du transmetteur.

CHAPITRE II

A NEURAL NETWORK APPROACH FOR THE LINEARIZATION OF RADIO-FREQUENCY POWER AMPLIFIERS WITH ADAPTIVE PREDISTORTION

2.1 Abstract

In this paper, we present a data on-line (real-time) modeling and linearizing techniques, for radio-frequency (RF) power amplifiers (PA). First, we present a dynamic adaptive modeling approach, based on real-valued time-delayed neural networks (RVTDNNs), to obtain dynamically the AM/AM and AM/PM characteristics of power amplifiers PA. Secondly, this model characteristics are adapted for baseband signal predistortion of PA. The proposed architecture is suitable for adaptive linearizing of PAs with memory effects. Instead of using indirect learning architecture, we propose a data on-line adaptive predistortion to ensure a continuous adaptation without interrupting the transmitting process. The modulator, demodulator and ADC/DAC imperfections are automatically compensated, because they are considered in the corrected imperfections loop. We adopte a baseband signal predistortion, it has the ability to simultaneously correct, the in-band-distortions (EVM) and the out-of-band distortions (ACPR). In case of the indirect learning architecture, two processors are needed at the same time, one for predistortion and the other for obtaining the neural network

NN parameters. The last approach suffers from noise at their inputs and desired outputs. In the proposed architecture, we need only one processor unit based on noiseless inputs and desired outputs. We demonstrate, by computer simulation, that almost 25 dB in ACPR is permanently suppressed after convergence at 350 kHz frequency offset, with a fast elimination of EVM. The proposed NN architecture is implemented with Xilinx system generator for DSP (XSG) in the Matlab/Simulink environment.

2.2 Introduction

The great challenge of modern wireless communication systems is to transmit, suitably and rapidly, a great quantity of data, which requires high flow communication channels. However, penury of frequency bandwidth obliges to use new spectral efficient linear digital modulation techniques, e.g. M-QAM, M-PSK and multiple access techniques, e.g. CDMA, WCDMA, CDMA2000 [14].

Unfortunately, the above techniques give a modulated complex signal with varying envelope characterized by a high peak to average power ratio factor (PAPR) like 9.7 dB for CDMA2000 [41] and 5.8 dB for 16-QAM[46]. The carrier signal will have a varying power with a dynamic envelope, it presents new constraints compared to non-linear modulation techniques, e.g. AM, FM.

One of the major constraints hindering the use of these modulation techniques is the linearity of the pathway travelled by the RF signal (Figure 2.1). By analyzing the components of the signal pathway, we can note that the receiver is devoid of nonlinear effects, because the level of its input signal $y_r(t)$ is very weak. The transmission channel is characterized by a dispersive frequency response and multi-path effects that can be corrected by an equalizer[25].

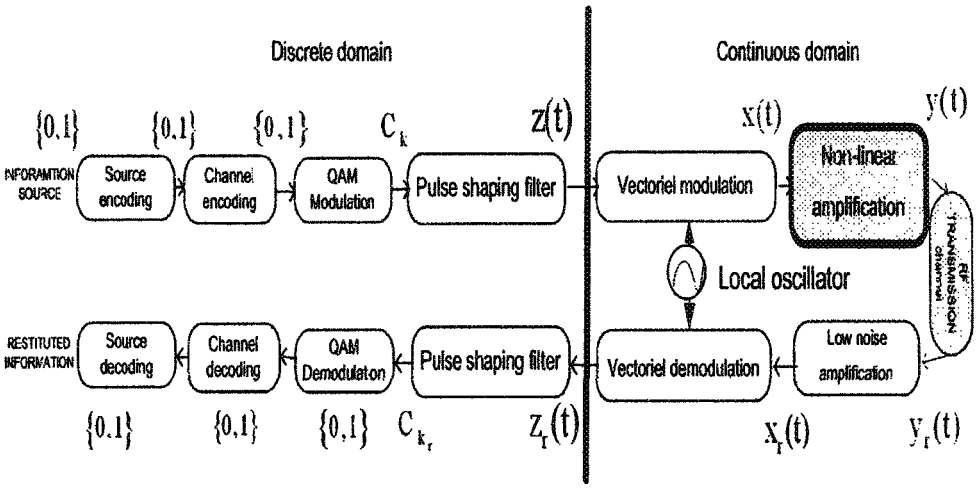


Figure 2.1: The pathway travelled by an useful signal in a wireless communication channel (discrete domain and continuous domain).

Actually, the only element of the communication channel that can present more important nonlinearities is the PA; it requires a high energy efficiency and a very linear characteristic at the same time. The PA is not able to operate near the saturation zone and at the same time provides a faithful amplified version of RF signal. These nonlinearities have very harmful consequences on the quality of the communication channel. The PA output signals is accompanied with spectral regrowth (i.e. ACPR) and deformations of the modulated signal constellation (i.e. EVM). In this condition, the first solution in hand is to drop the input power level «Back-off» ; however, it degrades the energy efficiency. Linearizing methods, such as feedback, feedforward, predistortion and post-compensation (equalization) [36, 59] are used to improve the linearity of PAs with high energy efficiency classes (e.g. class E and class F). However, techniques, such as Doherty, envelope elimination and restauration «EER» [36] attempt to improve PAs power efficiency of high linear classes (e.g. class A and

class AB). The predistortion approach drew the attention of many research teams this last decade. It is promoted by the development of digital signal processors (DSP) used in transmitters to implement filtering, coding and modulation/demodulation. This method is advantageous in term of simplicity and stability.

With the employment of complex wide-band modulation techniques, the memory effects consideration are ineluctable. They are caused by the pulse shaping filter in the transmitter [19, 17], the self-heating phenomena [63], the bias circuit, the modulated signal frequencies, the climatic changes and the PA obsolescence. Thus, the PA linearization process must be dynamic to compensate memory effects. To perform that, the dynamic adaptive predistortion is the best solution : it does not need hard synchronization operations and stability requirements. It can be cited the digital predistortion based on, look-up tables [8, 38], PA inverse models (e.g. polynomial model [15, 67], Volterra series [21, 75] and neural networks [9, 28]. The neural networks methods are better compared to the other cited methods : 1) the non-existence of gain table quantization effects, 2) the gain table indexing algorithm is not required, 3) a fairly small number of samples is sufficient for the predistorter training, and 4) the low complexity of their practical realization [51]. Data predistortion operates with the signal before being filtered with the pulse shaping filter [17], and analog (signal) predistortion operates with the signal after being filtered with the pulse shaping filter. The signal predistortion is retained because data predistortion can compensate only for the in-band distortions (EVM) and does not intentionally eliminate the out-of-band distortions (ACPR)[36, 65, 17]. Moreover, it depends on the type of the modulation technique and it must be adapted every time the modulation technique is changed[65].

Real-valued time-delay neural networks RVTDNNs are multi-layer perceptron neural networks proposed by Liu *et al.* [41], to overcome the complexity of the training algorithm and the convergence problems encountered in the previous NNs based techniques [9, 54, 1]. Unlike the use of two separate real NNs or one complex NN depending on the input/output representations [9, 54], Liu *et al.* [41] proposed the use of one real valued NN, that can process at the same time the signal with its two cartesian components.

In this work, is adopted a digital baseband signal predistorter based on RVTDNNs architecture [41]. This NN architecture is implemented with XSG software [70] in Matlab/Simulink environment. All parts of the NN (multipliers, activation functions, etc.) are implemented in a modular hardware components that run in FPGA circuit. We assume that just undesirable effects of the PA are present, and that all other devices imperfections [51] in the transmitter are not considered. The proposed architecture for adaptive data on-line linearization requires a simpler and more efficient configuration compared to the indirect learning architecture (figure 2.2).

This paper is organized as follow : firstly, some generalities about the linear modulated test signals are presented ; secondly, the realization of the PA memory model is exposed ; thirdly, is detailed the proposed NN architecture theory, and the results of the memory PA dynamical modeling process ; fourthly, The unique architecture of data on-line linearization and the obtained results of the proposed approach are described. Finally, RVTDNN architecture implementation with XSG software are briefly presented and the paper is completed with a general conclusion and propositions of future works.

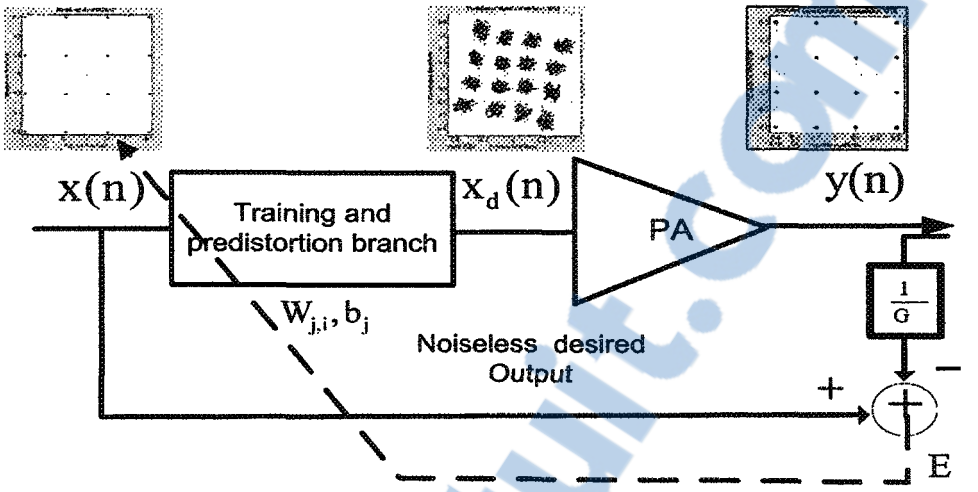


Figure 2.2: Proposed architecture Principle.

2.3 Linear modulated test signals

Modern communication systems are continuously developed to reach higher data rate. Nowadays, digital communications are widely used, taking advantages of digital signal filtering and modulation progression, which offers a greater data rate and a judicious exploitation of the frequency band. Figure 2.3 shows a digital transmitter general components realized in Simulink for the predistortion experiments.

To make the data transmission more resistant to the transmission channel disturbances, extra data bits for encoding, commonly called redundancy bits, are used. Source encoding purpose is to ensure a data transmission more efficient [18] by realization of data compression. These characteristics and operations make the amplified and transmitted source of information very redundant; it is a specification that will be advantageous in realizing the proposed data on-line modeling and predistortion architecture. Linear modulation tech-

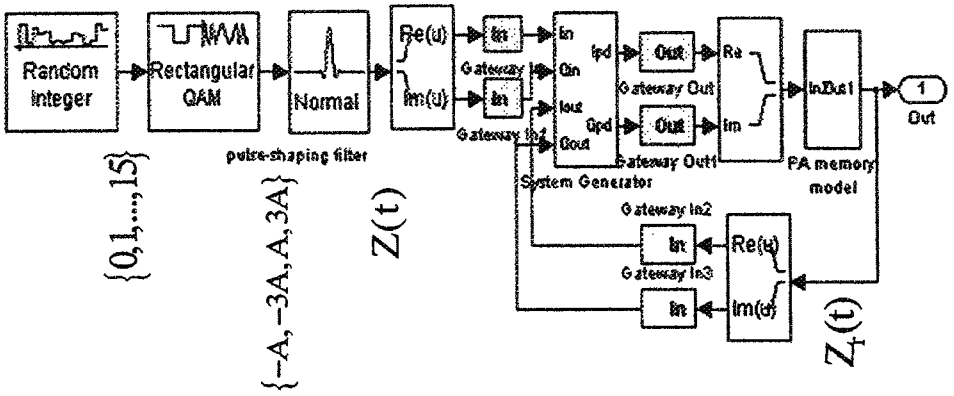


Figure 2.3: Digital transmitter model realized in Simulink to test the proposed architecture performances without interrupting the transmission process.

niques are widely employed, for their high data rate using a relatively narrow frequency band. In this section, it will be presented briefly the linear modulation techniques tested in the simulations of the present work.

Binary symbol encoding generates the symbol C_k from input digital signal; it can be complex or real. The symbol rate measured in «*baud*» is given by the number of symbols transmitted in unit of time. In the case where the symbol C_k is complex, this value must be coded in amplitude and phase. If a constant amplitude A of the signal is retained, we talke about the phase shift keying (PSK) modulation, m is the number of states in the signal and the number of possible symbols can be obtained from this relation :

$$C_k \in \{A\exp(\frac{j2k\pi}{m}),\ k = 0, ..., m-1\} \tag{2.1}$$

It can be also constructed an amplitude modulation (ASK) with 2^n states, where each symbol encodes n bits. The combination of both the ASK and the

PSK gives a quadrature amplitude modulated signal (M-QAM) and the information are encoded at the same time in the real and imaginary parts of the symbol C_k. For example, 16-QAM uses two (02) amplitude-modulations (one for I and the other for Q part) with four states ($s = 04$) in each part, here $M = 16 = 2^s = 2^4$:

$$C_k = I_{p.,k} + jQ_{p.,k} \qquad I_{p.,k}, Q_{p.,k} \in \{-3A, -A, A, 3A\} \tag{2.2}$$

where $I_{p.,k}$ and $Q_{p.,k}$ are the real (in-phase) and imaginary (quadrature-phase) part of the complex symbol C_k.

The transmission channel is a continuous medium : before being able to transmit the symbols, they must have a continuous waveform representation. This continuous waveform is obtained by interpolation. Symbols are transmitted at the frequency of $F_s = 1/T_s$, where T_s is the symbol duration. The waveform, which can interpolate this discrete signal, is a nonzero function $h(t)$ on the interval $[0, T_s]$. The waveform of the continuous signal takes the following form [25] :

$$x_c(t) = \sum_{k=o}^{\infty} C_k h(t - kT_s) \tag{2.3}$$

where $h(t)$ is a rectangular function waveform.

$$h(t) = \begin{cases} 1 & 0 \leq t < T_s \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

This signal has an infinite spectrum ; it is necessary to adapt its spectrum

for the band limited transmission channel and for keeping control of inter-symbol interference (ISI) effect as the modulation rate is increased. For this purpose, it should be filtered with a pulse shaping filter $h(t)$ [48, 25], which consists of a low pass filter serving to limit and determine the spectrum of the transmitted signal. Nyquist ISI criterion is commonly used for evaluating these types of filters : it relates the transmitted signal frequency spectrum to the ISI. For this purpose, the pulse shaping filter $h(t)$ commonly used for digital modulation systems is the raised cosine-filter, it can be a normal raised cosine or a square root raised cosine FIR filter. The impulse response of a square root raised cosine filter convolved with itself is approximately equal to the impulse response of a normal raised cosine filter. The impulse response of normal raised cosine filter is defined as [25] :

$$h_g(t) = \frac{sin(\frac{\pi t}{T_s})}{\frac{\pi t}{T_s}} \cdot \frac{cos(\frac{\pi \beta t}{T_s})}{1 - \frac{4\beta^2 t^2}{T_s^2}} \tag{2.5}$$

For a square root raised cosine filter, the impulse response is [47] :

$$h_g(t) = 4\beta \frac{cos(\frac{(1+\beta)\pi t}{T_s}) + \frac{sin(\frac{(1-\beta)\pi t}{T_s})}{(\frac{4\beta t}{T_s})}}{\pi\sqrt{T_s}(1 - (\frac{4\beta t}{T_s})^2)} \tag{2.6}$$

where $\beta \in [0, 1]$ is the roll-off factor determining the smoothness of the RF signal, and T_s is the symbol duration. The roll-off factor measures the excess bandwidth (Δf) of the filter.

$$\beta = \frac{\Delta f}{\frac{1}{T_s}} = \frac{\Delta f}{F_s} \tag{2.7}$$

The bandwidth occupied by the carrier is then defined as [43] :

$$B = \frac{1}{T_s}(1+\beta) = F_s(1+\beta) \tag{2.8}$$

The spectral efficiency Γ is defined as [43] :

$$\Gamma = \frac{R_c}{B} = \frac{R_c T_s}{1+\beta} = \frac{log_2 M}{1+\beta} \tag{2.9}$$

where $m = log_2 M$ is the number of bits per symbol and R_c is the bit rate at the source encoder output. For a `16-QAM` modulation, is found that $\Gamma = 3.07\, bit/sHz^{-1}$ with a roll-off factor $\beta = 0.3$.

The output of the pulse shaping filter gives a baseband complex signal $z(t) = r(t)e^{j\phi_0(t)} = I_p(t) + jQ_p(t)$, where $I_p(t) = r(t)cos(\phi_0(t))$ and $Q_p(t) = r(t)sin(\phi_0(t))$ are the in-phase and quadrature-phase parts respectively, and $r(t)$ and $\phi_0(t)$ are the instantaneous modulated envelope and phase, respectively.

After digital to analog conversion and low pass filtering, the In-phase component «I_p» and the Quadrature-phase component «Q_p» modulate two carriers phase-shifted of 90^0. The spectral distribution of the complex signal is not symmetric and, consequently, the information quantity transmitted in the same frequency bandwidth is doubled compared to that of a real-valued signal. The modulated signal takes the form :

$$x(t) = I_p(t)cos(2\pi f_c t) - Q_p(t)sin(2\pi f_c t) \tag{2.10}$$

where f_c is the carrier frequency.

The resulting modulated signal may be written in the form :

$$x(t) = Re[z(t)e^{j\omega_c t}] = r(t)cos(\omega_c t + \phi_0(t)) \tag{2.11}$$

where $\omega_c = 2\pi f_c$.

In this research work, is generated the 16-QAM and 8-PSK baseband modulated signals from the Matlab/Simulink blockset to test the PA model.

2.4 Power amplifier original model

2.4.1 Memoryless nonlinear subsystem

A nonlinear travelling-wave tube (TWT) amplifier can be modeled with a memoryless nonlinearity by presenting the static AM/AM and AM/PM conversion curves. These relations are based on numerical models, validated by experiments, and proposed in the Saleh's work [57] :

$$A[r(t)] = \frac{\alpha_a r(t)}{1 + \beta_a r^2(t)} \tag{2.12}$$

$$\Phi[r(t)] = \frac{\alpha_\Phi r^2(t)}{1 + \beta_\Phi r^2(t)} \tag{2.13}$$

where $r(t)$ is the instantaneous modulated envelope of the input signal. Typical normalized amplifier parameter values are [57] : $\alpha_a = 2.1587$, $\beta_a = 1.1517$, $\alpha_\Phi = 4.0033$ and $\beta_\Phi = 9.104$. In this condition, the gain of the amplifier is given by :

$$G[r(t)] = \frac{A[r(t)]}{r(t)} = \frac{\alpha_a}{1 + \beta_a r^2(t)} \tag{2.14}$$

With the in-phase and the quadrature-phase components, the memoryless nonlinearities are given by Saleh [57] :

$$I_p[r(t)] = \frac{\alpha_{I_p} r(t)}{(1+\beta_{I_p} r^2(t))} = A[r(t)]cos[\Phi[r(t)]] \tag{2.15}$$

$$Q_p[r(t)] = \frac{\alpha_{Q_p} r^3(t)}{(1+\beta_{Q_p} r^2(t))^2} = A[r(t)]sin[\Phi[r(t)]] \tag{2.16}$$

where the normalized values are : $\alpha_{I_p} = 2.0922$, $\beta_{I_p} = 1.2466$, $\alpha_{Q_p} = 5.529$, $\beta_{Q_p} = 2.7088$. In these conditions, the output distorted baseband signal y_{bb} takes the form :

$$\begin{aligned} y_{bb}(t) &= A[r(t)]e^{j(\Phi[r(t)]+\phi_0(t))} \\ &= A[r(t)](cos(\Phi[r(t)]+\phi_0(t)) + jsin(\Phi[r(t)]+\phi_0(t))) \end{aligned} \tag{2.17}$$

The cartesian components of the output baseband signal are distorted in function of the input signal amplitude. It can be noted that for very large values of `r(t)`, both $I[r(t)]$ and $Q[r(t)]$ are inversely proportional to $r(t)$ and they are odd functions of $r(t)$ [57]. It has been chosen normalized parameters because **NNs** operate generally with normalized input values (-1 to +1). The modulated output signal of the **PA**, can be written in the form :

$$y(t) = Re[y_{bb}(t)e^{j\omega_c t}] = A[r(t)]cos(w_c t + \phi_0(t) + \Phi[r(t)]) \tag{2.18}$$

2.4.2 Memory linear subsystem

The short term memory effects are modeled with a linear time-invariant dynamic **LTI** filter that is based on a finite impulse response (**FIR**) filter. This

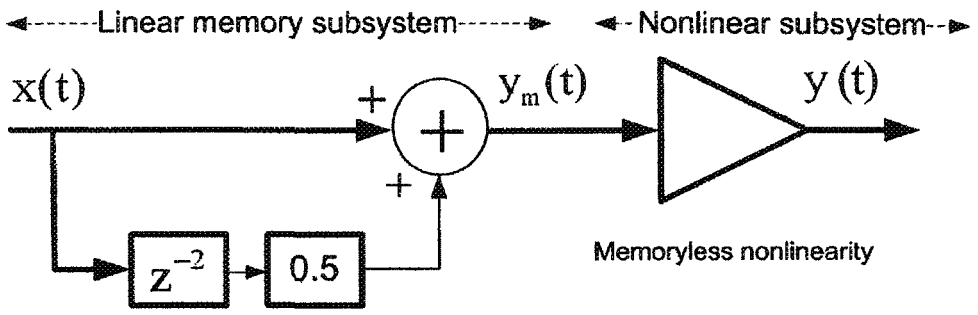


Figure 2.4: Block diagram of the modeled memory nonlinear subsystem to present the RF amplifier.

filter is used to provide an output signal depending on the input signal variation in time. The subsystem containing the LTI filter is placed upstream of the memoryless nonlinearity according to the Wiener model [54, 27]. Short-term memory effects can then be entirely introduced by the LTI filter effects observed on the output signal. The depth of the filter memory effects is then $q = 2$, the filter output can be written as done in the work [54] in the form :

$$y_m(n) = x(n) + 0.5x(n-2) \tag{2.19}$$

The output signal $y_m(n)$ is then a function of the input signal $x(n)$ at the instants n and $n-1$.

2.4.3 Memory nonlinearity system

The overall memory nonlinear system is composed of a memory linearity subsystem followed by a memoryless nonlinearity model. This gives a memory nonlinear model of the PA according to the Wiener model [54, 27]. Figure 2.4 illustrates the block diagram of this model.

Long-term memory effects are introduced by a small variation in the PA parameters $(\alpha_a, \alpha_\phi, \beta_a, \alpha_\phi)$ during the simulation process. This dynamic model is behaviorally equivalent to the polynomial model of RF power amplifier [54, 28]. Which is defined by :

$$y(n) = \sum_{k=0}^{K} \sum_{q=0}^{Q} c_{(2k+1)q} x(n-q) \mid x(n-q) \mid^{2k} \tag{2.20}$$

where $c_{(2k+1)q}$ are the parameters of the PA model, K and Q are respectively the order of the PA nonlinearity and the memory depth. A Simulink realization of this model is shown on the appendix 3.1

2.5 Linearization performance criteria

The main parameters used to evaluate the proposed linearization technique are :

1. Out-of-band distortion components : they can be evaluated with the Adjacent Channel Power Ratio (ACPR). It is a measurement of the amount of interference, or power, in the adjacent frequency channel [55]; it is calculated with the following equation :

$$ACPR = \frac{\int_{f_0-0.5B_{adj}}^{f_0+0.5B_{adj}} S(f)df}{\int_{-0.5B_{ch}}^{0.5B_{ch}} S(f)df} \tag{2.21}$$

   where $S(f)$ is the power spectral density function of the signal, B_{adj} is the specified adjacent channel bandwidth at a given frequency offset f_0, and B_{ch} is the bandwidth of the useful signal.

2. The In-band distortion components : they degrade the bit error rate pa-

rameter BER of the system. The error vector magnitude (EVM) is used to measure the difference between the reference waveform and the measured waveform at the PA output. EVM parameter measures the normalized difference between reference cartesian components and measured cartesian components. Denote by $I_{p,out}$ and $Q_{p,out}$ the measured cartesian components at the PA output. And by $I_{p,d}$ and $Q_{p,d}$ the desired correct reference cartesian components. We define the EVM over N samples of the input signal, using the MSE of the vector $\vec{\epsilon} = (I_{p,d}(n) - I_{p,out}(n)) + j(Q_{p,d}(n) - Q_{p,out}(n))$ [10] :

$$\begin{aligned} \text{EVM} &= \sqrt{\frac{1}{N}\sum_{n=1}^{N}|\vec{\epsilon}|^2} \\ &= \sqrt{\frac{1}{N}\sum_{n=1}^{N}(I_{p,d}(n) - I_{p,out}(n))^2 + (Q_{p,d}(n) - Q_{p,out}(n))^2} \end{aligned} \tag{2.22}$$

N is the number of samples, it can be considered as the number of samples per epoch to train the NN. The EVM is the image of the MSE presenting the cost function of the NN training algorithm.

3. Convergence rate : we define the convergence rate C_{rate} of the NN to evaluate its convergence speed. It is calculated as follows :

$$C_{rate} = \frac{|\text{MSE}(t_{end}) - \text{MSE}(t_0)|}{t_{end} - t_0} \tag{2.23}$$

where t_{end} is the instant of the stopping criterion satisfaction and t_0 is the instant of the stopping criterion dissatisfaction detection (the starting correction instance).

2.6 Memory power amplifier data on-line modeling architecture

2.6.1 Neural network theory

The modeling of microwave PAs is one of the key subject in wireless communication systems. This is mainly driven by the need of precise model to be used for behavioral study and linearization of PAs. Through evolution of technology in DSPs, FPGAs, ADC and DAC, the PA is modeled and dynamically characterized on a wide bandwidth with excellent performances. With the improvement of modulation techniques, the dynamic behavioral modeling is inescapable. In this section, it will be validated, by simulation, with the test signal (16-QAM) the results given in several works [41, 28]. A RVTDNNs architecture will be realized with XSG software in the Matlab/Simulink environment.

The proposed model is based on the sampled baseband signal, after being filtered by the pulse shaping filter. The model based on data before the pulse shaping filter can not well model the output-of-band behavioral, because it can not compensate the out-of-band distortions when linearizing (section 2.2) [36, 65, 17]. This model is able to reproduce at the same time out-of-band (ACPR) and in-band (EVM) distortions. As shown in figure 2.5, signals $I_{p.in}$ and $Q_{p.in}$ are modulated to be injected by a 90^0-coupler in the PA, at the output, the RF signal is passed through a 90^0-coupler and is demodulated to get the distorted $I_{p.out}$ and $Q_{p.out}$ signals. The flow of the information signal in the forward direction of the RVTDNN can be summarized in the following equations :

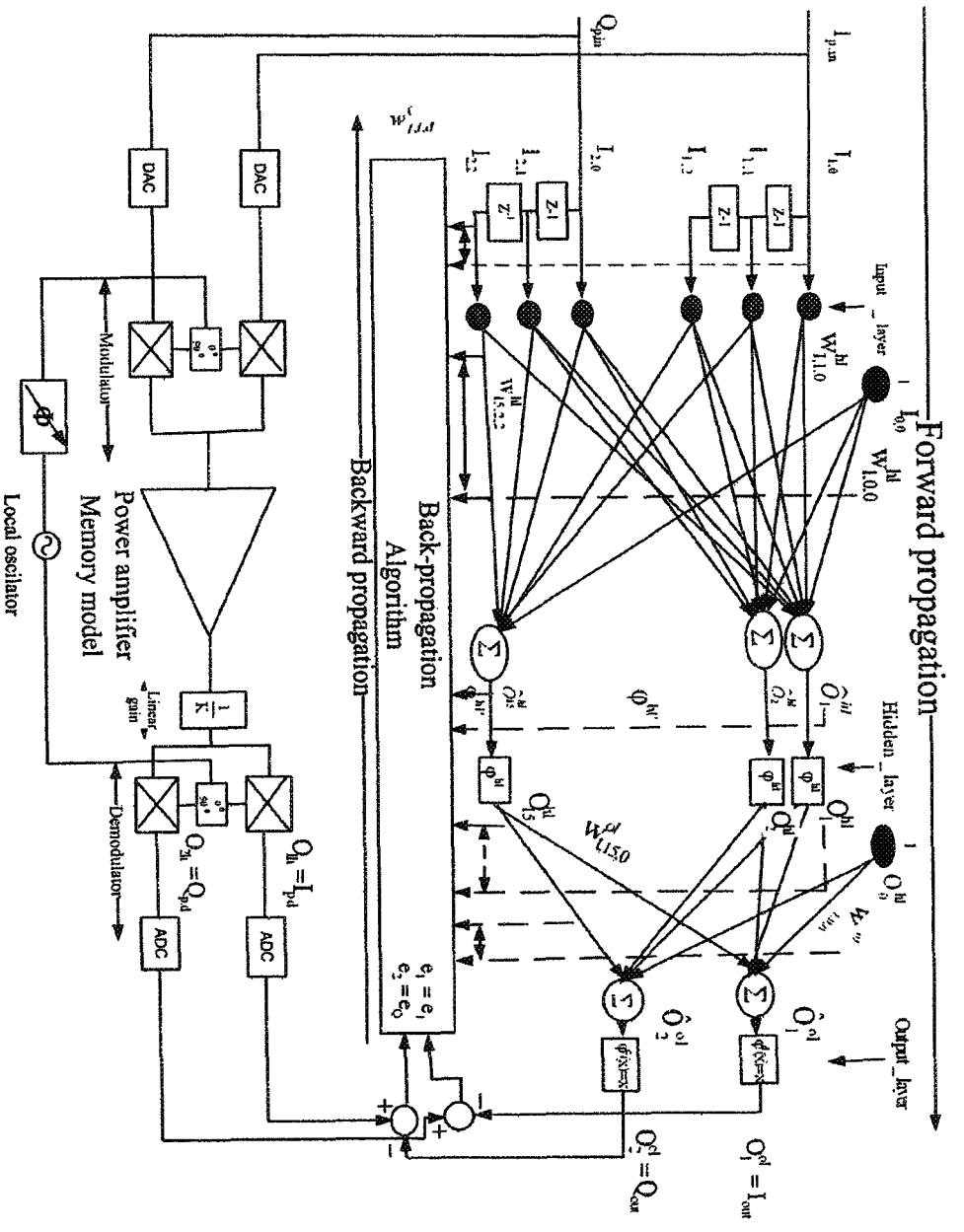


Figure 2.5: Proposed NN modeling structure : the top part is the NN architecture and the bottom is the PA surrounded by its accessories (Mod./Dem. and CNA/CAN).

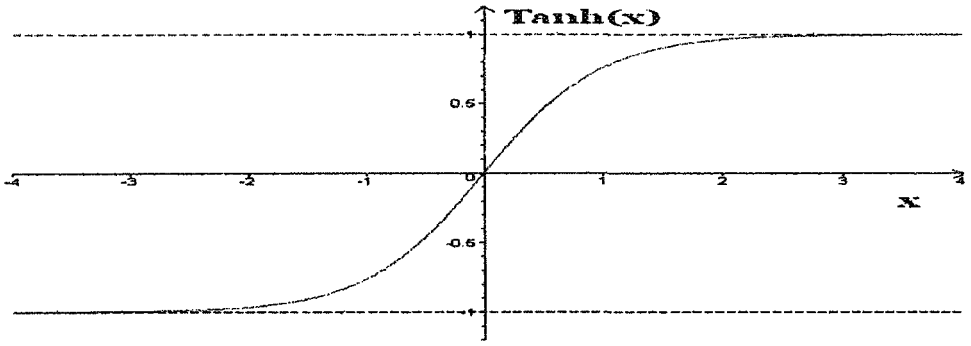


Figure 2.6: The nonlinear odd form of the activation function tanh(x).

$$I_{p.out}(n) = O_1^{ol}(n) = \varphi^{ol}[\sum_{j=1}^{N_{hl}} w_{1,j,0}^{ol} \varphi^{hl}\{\sum_{d=0}^{q}(w_{j,1,d}^{hl}(I_{p.in}(n-d) + w_{j,2,d}^{hl} Q_{p.in}(n-d)) + w_{j,0,0}^{hl}\} + w_{1,0,0}^{ol}] \quad (2.24)$$

$$Q_{p.out}(n) = O_2^{ol}(n) = \varphi^{ol}[\sum_{j=1}^{N_{hl}} w_{2,j,0}^{ol} \varphi^{hl}\{\sum_{d=0}^{q} w_{j,1,d}^{hl}(I_{p.in}(n-d) + w_{j,2,d}^{hl} Q_{p.in}(n-d)) + w_{j,0,0}^{hl}\} + w_{2,0,0}^{ol}] \quad (2.25)$$

where $w_{j,i,d}^{ol}$ is the connection weight between neuron i of the hidden layer to the neuron j of the output layer and $w_{j,0,0}^{ol} = b_j^{ol}$ is the biases of the output layer neuron j. Because there are no delays in the hidden layer neurons, d is always set to 0 in the relating synaptic weights ($w_{1,j,0}^{ol}$, q=2 is a constant value referred to the memory depth of the PA). There are only two neurons in the NN output and there are two equations without mentioning N_{ol}. The result of the seconde summation is applied to the hidden layer nonlinear function φ^{hl}, after that, it is calculated within the first summation by varying j, i and d.

The variable $w_{j,i,d}^{hl}$ is the connection weight between neuron i of the in-

put layer delayed of d samples period and the neuron i of the hidden layer. Parameters N_{hl} and N_{ol} are the numbers of neurons in the hidden and the output layers, respectively, and q is the memory depth. The function $\varphi^{hl}(.)$ is the nonlinear activation function of the hidden layer neurons and $\varphi^{ol}(.)$ is the activation function of the output layer neurons, which is chosen to be purely linear $\varphi^{ol}(x) = x$ because it will not affect the approximation task of the NN and it will optimize the memory and Silicon surface. A linear function does not need any FPGA ressources to be realized, a simple linear link is sufficient. The activation function of the hidden layer neurons is an hyperbolic tangent function (see figure 2.6) :

$$\varphi^{hl}(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.26}$$

It has been opted for this function despite of the Log-sigmoid because the PA model is based on cartesian components, that their based nonlinearities representations are odd functions of $r(t)$, not like polar components based models, where only AM/AM conversion is an odd function of $r(t)$ [57]. Back-propagation algorithm uses the steepest descent method in synaptic weights change and it can be done in sequential mode or batch mode [24]. The sequential mode performs weights updating after the presentation of each sample of the learning set, the forward and backward computations are performed at each learning sample and the weights and biases are adjusted. Further adjustments are performed at every learning sample presentation until satisfying the stopping criterion. However, in batch mode the weight updating is performed after the presentation of all learning samples that constitute one epoch.

It has been opted for the sequential mode updating process. This mode is

advantageous because it can ensure a data on-line modeling and linearizing process [24]. Also, this mode takes a low convergence speed with redundant information sources. This mode of training is known for its less-requirements of the data storage and the stochastic nature of the search in the weight space. It can also prevent back-propagation algorithm to get stuck at a local minimum. For our work, samples in the learning set are presented at any time of the convergence process are redundant for either In-phase and Quadrature-phase components, and encoding operation of the canal gives a great redundancy to the transmitted symbols.

2.6.2 Back-propagation algorithm

2.6.2.1 Output layer neurons

The error signal at each neuron in the output layer is the first available information that can be observed, that can be obtained from :

$$e_j(n) = (O_{jh}(n) - O_j^{ol}(n)) \tag{2.27}$$

where, $O_{jh}(n)$ is the desired NN output and is also the actual output of the PA model to be reproduced, $O_j^{ol}(n)$ is the actual output of the neuron j in the NN output layer. The total error energy $E(n)$ is obtained by mean summing the squares of the errors of equation 2.27 at the output layer neurons :

$$E(n) = \frac{1}{2}\sum_{j=1}^{2}(e_j(n))^2 \tag{2.28}$$

The mean square error MSE is the cost function that it is searched to mini-

mize, it represents the stopping criterion for the learning algorithm of the NN. It gives an information on the EVM variation defined in the equation 2.22. The MSE is equal to :

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N} E(n) \tag{2.29}$$

According to equation 2.22, the MSE can be written in function of the EVM as :

$$\mathrm{MSE} = \frac{\mathrm{EVM}^2}{2} \tag{2.30}$$

where N is the number of samples in the learning epoch. By using back-propagation, the weights change $\Delta \mathbf{w}_{j,i,d}(n)$ can be calculated with the same manner of the LMS algorithm [24]. It can be expressed the gradient according to the chain rule at the output layer neurons as :

$$\frac{\partial E(n)}{\partial \mathbf{w}^{ol}_{j,i,0}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial O^{ol}_j(n)} \frac{\partial O^{ol}_j(n))}{\partial \hat{O}^{ol}_j(n)} \frac{\partial \hat{O}^{ol}_j(n)}{\partial \mathbf{w}^{ol}_{j,i,0}(n)} \tag{2.31}$$

This partial derivative equation represents a sensitivity factor, determining the direction of search in the weight space for the corresponding synaptic weight $\mathbf{w}^{ol}_{j,i,0}$. We use the chain rule to evaluate every part of equation 2.31 as follows :

From equation 2.28 it can be written :

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \tag{2.32}$$

from equation 2.27 and because φ^{ol} is linear, we find that :

$$\frac{\partial e_j(n)}{\partial O_j^{ol}(n)} = -1 \quad and \quad \frac{\partial O_j^{ol}(n))}{\partial \hat{O}_j^{ol}(n)} = 1 \tag{2.33}$$

Also, from equations 2.24 and 2.25 :

$$\frac{\partial \hat{O}_j^{ol}(n)}{\partial w_{j,i,0}^{ol}(n)} = O_i^{hl}(n) \tag{2.34}$$

Defining the local gradient of the output layer neurons as :

$$\delta_j^{ol}(n) = -\frac{\partial E(n)}{\partial \hat{O}_j^{ol}(n)} = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial O_j^{ol}(n)} \frac{\partial O_j^{ol}(n)}{\partial \hat{O}_j^{ol}(n)} = e_j(n). \tag{2.35}$$

The equation 2.31 can be written as :

$$\frac{\partial E(n)}{\partial w_{j,i,0}^{ol}(n)} = -e_j(n) O_i(n)^{hl} = -\delta_j^{ol}(n) O_i^{hl}(n) \tag{2.36}$$

2.6.2.2 Hidden layer neurons

The hidden layer neurons do not have a specific desired output value, that the calculation is more complex relating to that of the output layer. So, the error signal for these neurons has to be determined recursively in terms of errors of all neurons to which every single hidden layer neuron is directly connected [24]. The local gradient of the neuron j in the hidden layer may be defined :

$$\delta_j^{hl}(n) = -\frac{\partial E(n)}{\partial O_j^{hl}(n)} \frac{\partial O_j^{hl}(n)}{\partial \hat{O}_j^{hl}(n)} = -\frac{\partial E(n)}{\partial O_j^{hl}(n)} \varphi^{hl'}(\hat{O}_j^{hl}(n)) \tag{2.37}$$

$\varphi^{hl'}(.)$ is the derivative of the activation function applied to its input.

$$\varphi^{hl'}(x) = 1 - (\varphi^{hl}(x))^2 \tag{2.38}$$

From equation 2.28, it can be written that :

$$\frac{\partial E(n)}{\partial O_j^{hl}(n)} = \sum_{k=1}^{N_{ol}} e_k(n) \frac{\partial e_k(n)}{\partial O_j^{hl}(n)} \tag{2.39}$$

For notation requirements, we note with $e_k(n) = (O_{kh}(n) - O_k^{ol}(n))$ the error related to the neuron k of the output layer at the n^{th} sample time. Using the chain rule, the above equation can be written as :

$$\frac{\partial E(n)}{\partial O_j^{hl}(n)} = \sum_{k=1}^{N_{ol}} e_k(n) \frac{\partial e_k(n)}{\partial O_k^{ol}(n)} \frac{\partial O_j^{ol}(n))}{\partial \hat{O}_j^{ol}(n)} \frac{\partial \hat{O}_k^{ol}(n)}{\partial O_j^{hl}(n)} \tag{2.40}$$

From equation 2.33 and because $\hat{O}_k^{ol}(n) = \sum_{j=0}^{N_{hl}} \mathtt{w}_{k,j,0}^{ol}(n) O_j^{hl}(n)$, where $\mathtt{w}_{k,0,0}^{ol}(n) = b_k^{ol}$ is the bias of the output layer k^{th} neuron ($O_0^{hl}(n) = 1$), it can be written :

$$\frac{\partial \hat{O}_k^{ol}(n)}{\partial O_j^{hl}(n)} = \mathtt{w}_{k,j,0}^{ol}(n) \tag{2.41}$$

Then equation 2.40 can be written as :

$$\frac{\partial E(n)}{\partial O_j^{hl}(n)} = -\sum_{k=1}^{N_{ol}} e_k(n).\mathtt{w}_{k,j,0}^{ol}(n) = -\sum_{k=1}^{N_{ol}} \delta_k^{ol}(n).\mathtt{w}_{k,j,0}^{ol}(n) \tag{2.42}$$

So, the local gradient for the hidden layer neurons (equation 2.37) can be

obtained from :

$$\delta_j^{hl}(n) = \varphi^{hl'}(\hat{O}_j^{hl}(n)) \sum_{k=1}^{N_{ol}} \delta_k^{ol}(n) w_{k,j,0}^{ol}(n) \tag{2.43}$$

Finally, the updating value of the weights of the output layer neurons is :

$$\Delta w_{j,i,0}^{ol}(n) = -\mu \frac{\partial E(n)}{\partial w_{j,i,0}^{ol}(n)} = \mu \delta_j^{ol}(n) O_i^{hl}(n) \tag{2.44}$$

and that of the hidden layer neurons is :

$$\Delta w_{j,i,d}^{hl}(n) = \mu \delta_j^{hl}(n) I_{i,d}(n) \tag{2.45}$$

where μ is the learning rate parameter of the back-propagation algorithm, it determines the speed of the NN convergence. There is no specific rule to choose the value of μ, its value is obtained experimentally. If the learning rate is very small, the NN learning needs more time. However, if the learning rate is very large, the NN will oscillate and will compromise the convergence. Also, sometimes, if the NN becomes saturated, the outputs will be very large. The solution to this problem is to choose a variable learning rate initialized to a large value near unity [24], and reduced during the back propagation algorithm, to a minimum positive value fixed beforehand. In the present work case, fast convergence is needed only when the PA characteristics change (long-term memory effects) during the adaptation process. By experiments, we retain a very small learning rate ($\mu = 2^{-10}$) and we chose random small initial weights. It is easy to realize a multiplication in two's complement fixed point format, when the multiplied number has the form of ($\mu = 2^{-10}$), by simply using shifters. It will simplify the training algorithm implementation and ensure a smooth

convergence trajectory in the weight space without oscillations.

The new values of the weights and biases at each instant of the training epoch are obtained according to the following equation :

$$\mathtt{w}^{\ell}_{j,i,d}(n+1) = \mathtt{w}^{\ell}_{j,i,d}(n) + \Delta\mathtt{w}^{\ell}_{j,i,d}(n) \tag{2.46}$$

ℓ can be an output layer *ol* or a hidden layer *hl*.

2.6.2.3 Stopping criterion

Stopping criterion is a test operation realized at each learning epoch, it can be decided whether or not to stop or not the learning process; we make the comparison between the calculated $\mathtt{MSE}_{cal}$ and the acceptable $\mathtt{MSE}_{acc}$ defined beforehand according to inequality 2.47. This value is obtained in function of precision and exactitude of the possible desired results.

$$\mathrm{MSE}_{cal} \leq \mathrm{MSE}_{acc} \tag{2.47}$$

The following steps are realized during the learning process of the **NN** :

1. Initialize the synaptic weights $\mathtt{w}_{j,i,d}$ at random values (generally between -1 and +1);
2. Initialize an accumulator of instantaneous errors energy to zero $E(0) = 0$;
3. Present a sample of the input signal, and calculate the outputs of all output layer neurons using the equations 2.24 and 2.25;
4. Specify the desired output and evaluate the local gradient for all neurons using equations 2.35 and 2.37;

5. Adjust synaptic weights according to equations 2.44, 2.45 and 2.46 ;
6. Calculate the error energy from equation 2.28 and add it to accumulator ;
7. Repeat steps 3 through 6 for each instant of the learning epoch ;
8. Calculate the mean square error MSE according to equation 2.29 ;
9. If $\text{MSE}_{cal} \leq \text{MSE}_{acc}$ then go to 10, else go to 2 ;
10. Save weights and biases of the NN and end.

In this algorithm procedure, steps 2 to 8 represent a learning epoch.

2.6.3 Simulation results

As shown in figure 2.5, it has been used a NN consisting of an input layer with 2 neurons and 2 tapped lines in each input cartesian component, a hidden layer with 15 neurons and an output layer with 2 neurons. There is no general rule for determining the structure of the network (the number of hidden layers and the number of neurons per hidden layer) [24]. For microwave applications, one hidden layer is sufficient since the main concern is the generalization capability of the network [58]. The hidden layer serves for detecting and memorizing the PA features with the learning process.

The experimental procedure of determining the number of neurons in the hidden layer consists of calculating the MSE and the ΔMSE for various numbers of neurons in the hidden layer, we test the NN with different number of neurons in the hidden layer with NN toolbox, learning vectors are obtained from the cartesian inputs and outputs of the PA. Table 2.1 shows a small and marginal variation of the MSE after having 15 neurons in the hidden layer. The number of the tapped delay lines is fixed to ($q = 2$) according to the memory depth assumed to the memory PA model (figure 2.4).

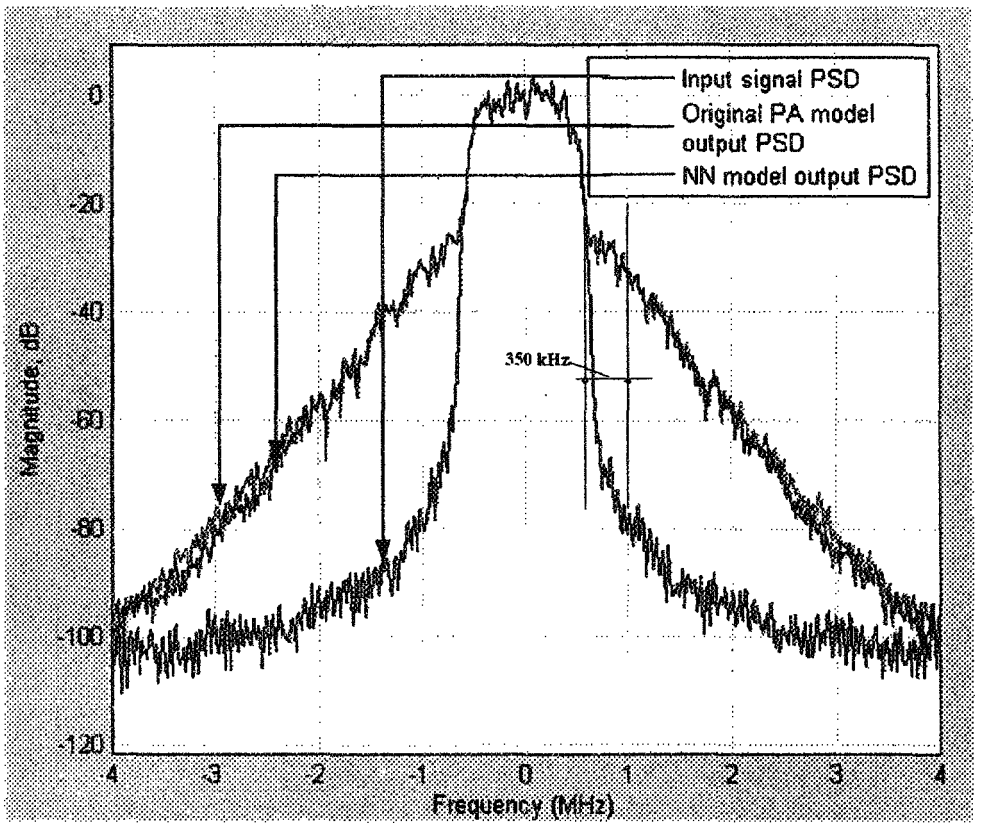


Figure 2.7: Power spectral density of the input signal, the original PA model output and the NN model output.

The 16-QAM modulated signal generated with Matlab/simulink is used to test the PA modeling ability of the RVTDNNs. The resulting power spectral densities of the input signal, the original PA model output, and the NN model output are shown in figure 2.7. We note that with baseband signal based modeling, the out-of-band distortions of the memory PA are well modeled at 350 kHz frequency offset.

Figure 2.8 shows the $I_{p.}$ and $Q_{p.}$ components of the output signal of original

Table 2.1: The MSE improvement in function of the number of neurons in the hidden layer.

N^0 of neurons in the hl	5	10	15	20	25
MSE($x10^{-5}$)	1.13464	0.205959	0.0149397	0.00879056	0.0070926
ΔMSE($x10^{-5}$)	–	0.928681	0.1910193	0.00614914	0.00169796

amplifier model, and those of the output signal of the NN model. It can be noted that the signals obtained with the original PA model and the NN model are almost superposed.

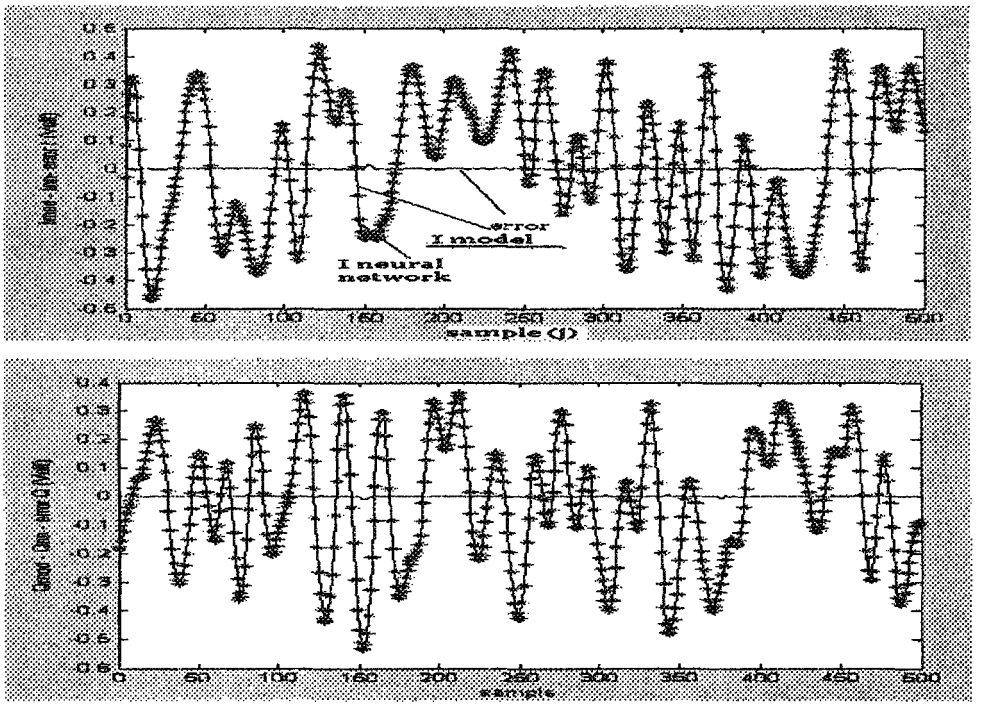


Figure 2.8: Cartesian components signals of the original PA model output (-) and the NN model output(*), I_p component (top), Q_p component (bottom).

The resulting constellation mapping in the complex plan obtained by the two approaches are presented in figure 2.9.

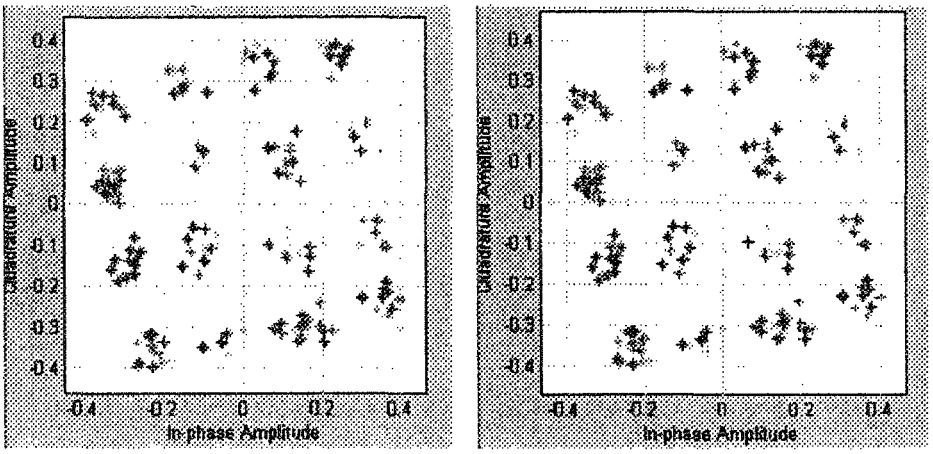


Figure 2.9: Signal constellation of original amplifier model (left), **NN** model (right).

Figure 2.10 shows the **AM/AM** and **AM/PM** conversions curves obtained by the original model and those obtained by the **NN** model. It is clear that with the **NN** architecture, both nonlinear and memory effects are well modeled at the same time.

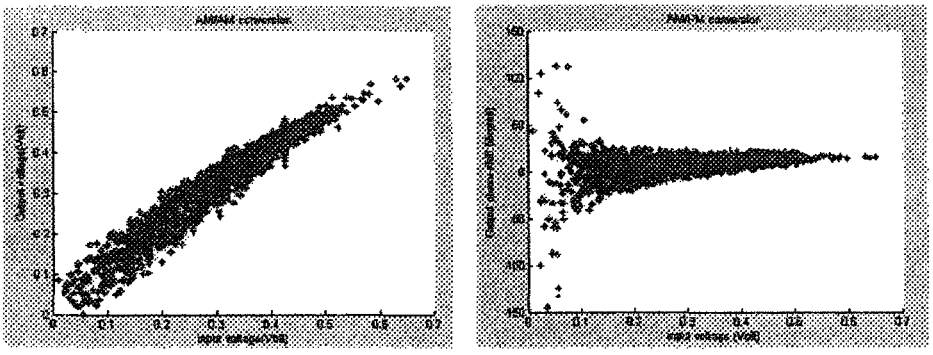


Figure 2.10: Comparison of **AM/AM** (top) and **AM/PM** (bottom) curves for the original amplifier model (red) and **NN** model with memory (blue).

2.6.4 Discussions and conclusion

In this section, we demonstrate with the same general architecture as shown in the works [41, 42, 28], the ability of the RVTDNN to model a PA with memory effect, using linear modulated test signal (16-QAM). We propose a NN modeling using a vector signal analyser (VSA) that can be used only in laboratory; this VSA based modeling test-bed needs a very expensive and advanced VSA that can not be installed in wireless communications base-stations on sites. Also when there are changes in environment such as climatic conditions and PA temperature, it is necessary to model again the PA to get new characteristics (AM/AM and AM/PM). That means that this model works only in conditions were measurements have been realized. But when using an FPGA based model, we instantaneously get the PA characteristics when it is needed. The RVTDNNs model is not dependent on the input signal modulation techniques and the PA type. So the proposed approach is universal for any PA type, class or modulated signal without changing hardware or software for obtaining dynamic AM/AM and AM/PM behavior.

The proposed NN based PA modeling method is suitable for base-stations using linear digital modulation. This approach is also valid according to several other works [41, 42, 28] for third generation (3G) base-station PAs that have very high peak to average power ratio (PAPR) and that are using digital modulated signal such as CDMA2000 and W-CDMA. We prove that modeling can be done with a simple robust adaptation algorithm with only one performant FPGA board. We demonstrate almost the same AM/AM and AM/PM results obtained by using the proposed NN based model. This obtained model is essential to correct the nonlinearity by using various kinds of linearizers. The dynamical NN can

be implemented on FPGAs systems to obtain data on-line AM/AM and AM/PM responses.

2.7 Memory power amplifier data on-line predistortion architecture

2.7.1 Back-propagation algorithm modification

Neural networks are used in linearizing PAs because of their capability of modeling and fitting nonlinearities. Using that, this characteristic can be generalized to memorize severe nonlinearities with memory. Linearization with RVTDNNs is reported in the work of Hwangbo *et al.* [28] by using the indirect learning architecture. It was proved that the trained NN is able to operate with (3G) base-station signals such as WCDMA and CDMA2000[41]. The proposed NN for linearization consists of a similar architecture (RVTDNNs) used for PAs modeling with the back-propagation algorithm in the adaptation process.

Figure 2.11 shows a proposed practical architecture for implementation of the global linearized transmitter, the input and output cartesian signals are compared to train the NN implemented on the Xilinx FPGA, low-pass filters are used to filter the signals after ADC and DAC operations done by the circuit Memec p160. The proposed NN predistortion architecture is based on sampled baseband signals, the Mod/Dem operations are achieved before using the signals by the FPGA board. This architecture can be applied for any type or class of PAs. It can be achieved with complex signals like CDMA and OFDM without modification of the NN general structure[65]. The $\frac{1}{G}$ block is used to compensate the PA linear gain in the backward information flow. The data on-line adapta-

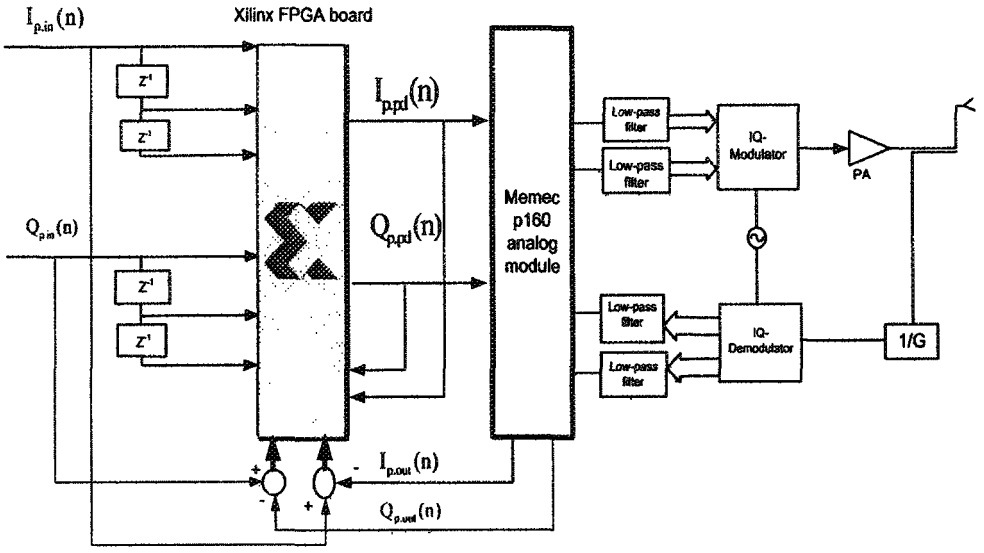


Figure 2.11: Proposed NN predistortion architecture for data on-line hardware implementation on FPGA

tion is needed only when the PA characteristics change with long-term memory effect caused mainly by phenomena of self heating and aging of the PA. In our case, a very small learning rate μ ($\mu = 2^{-10} = 9.7 \times 10^{-4}$) as done in the modeling section) is sufficient to have a smooth approximation to the trajectory in the weight space during the adaptation process [24]. In these conditions, the proposed linearizer can avoid problems of oscillations phenomena caused mainly by the bad choice of the learning rate μ in backward signal flow. The value of μ that cause oscillations depends on the signal frequency (learning data speed), it is obtained experimentally.

When adapting the NN parameters (weights and biases), it is necessary to propagate the error variation information with respect of these parameters ($\frac{\partial E(n)}{\partial w_{j,i,0}^{ol}(n)}$) from the demodulated outputs ($I_{p,out}$ and $Q_{p,out}$) of the PA with memory effects, which are considered as decision nodes. The PA long-term

memory effects must be taken into account during the backward propagation of the error gradient during the adaptation process. By analyzing the work of Benvenuto *et al.* [9], it was needed to have approximate models integrated with the NN, for the PA and the modulator and demodulator pulse responses; these models served only in the adaptation process to accelerate the convergence speed of the algorithm. The error to be minimized by the back-propagation algorithm is calculated at the PA output after being demodulated and divided by the desired linear gain. In this case, one digital processor performant block is sufficient to linearize the PA. Basing our work on that of Benvenuto *et al.*[9], we propose a quasi-similar method of linearization, where we develop our own approach to propagate the error information from the decision point at the demodulated PA output through the RVTDNN parameters. We note that the authors have adopted a predistortion placed before pulse shaping filter (data predistortion scheme) and it was seen that this predistortion architecture is not able to cancel out-of-band distortions (ACPR) well and suffer from memory effect added by the pulse shaping filters [36, 65]. Also, this architecture, based on CVNNs networks, is suffering from cumbersome training algorithm and convergence problems [41]. Conversely the proposed predistortion is performed with a pulse shaped baseband signal; it is able to minimize out-of-band distortions (ACPR) and in-band distortion (EVM) at the same time. The used NN is real-valued, so, relatively, it does not need complex calculations, and does not suffer from convergence problems.

We note an efficient stability during the adaptation process, by assuming the derivative of the nonlinear function *tanh* as the relation between PA output and input cartesian components in the backward information flow. Furthermore, this assumption has improved stability and dynamic in the adaptation

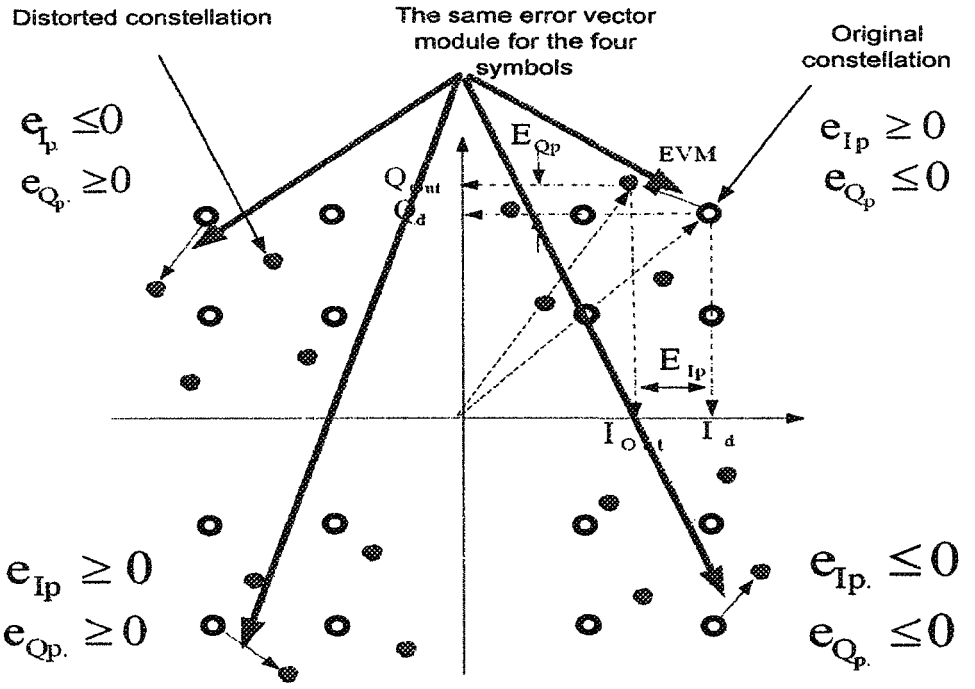


Figure 2.12: Data samples redundancy specification when using RVTDNNs with 16-QAM modulation.

process of compensation of the long-term memory effects. In other words, the error signal which will be used to calculate the gradient of the error in the output nodes is multiplied by the derivative of the function *tanh* (equation 2.38) applied to the NN output neurons, to calculate the local gradient of the related node (equation 2.50). This is an even function and does not affect the principle of the back-propagation algorithm. The *tanh* function is indirectly assumed to present the general form of the relation between the PA inputs and outputs in the forward signal flow, it is an odd function like the input/output relations presenting the memoryless nonlinearity (equations 2.15 and 2.16).

The errors, calculated at the four parts of the complex plan for symmetric

symbols constellations, will give the same weight changes after calculation with equations 2.44 and 2.45 (in 16-QAM, there are 4 different errors information). This is a redundancy that participates to accelerate the convergence process. Ideally, it can be said that a well trained RVTDNN to correct the errors just in the first part of the complex plan can generalize to the three (03) other parts (figure 2.12 explains this specificity). On the other hand, when using amplitude and phase based NN, each symbol has its own phase from 0 rad to 2π rad, with this approach there is no redundancy in the training set. Finally, the added assumption does not affect the general functionality of the back-propagation algorithm. It keeps the sign of the calculated errors at each decision point.

When there is a small PA input signal amplitude, the error value is relatively small, and it will be compensated faster than when the input signal amplitude is large, because after being multiplied by the derivative function applied to output layer neurons, it will present relatively great local gradient information. Also, to prevent oscillations, the network forces the errors of large amplitude symbols to be compensated slowly. This algorithm specificity will also participate in the smoothness of the trajectory in weight space. Moreover, when the NN input signal is zero, the error signal will be large and negative at the decision node, and the assumed derivative function will limit the NN parameters to diverge to large values. At the worst situation, this assumption will mainly affect NN convergence speed during the adaptation process against PA characteristic changes. Figure 2.13 shows the general flow of information during the predistortion and the adaptation process for the proposed RVTDNN linearization architecture. The input and output of this NN are respectively the cartesian components of the input signal and those of the predistorted signal.

In the present work, a data on-line predistortion architecture is used, which

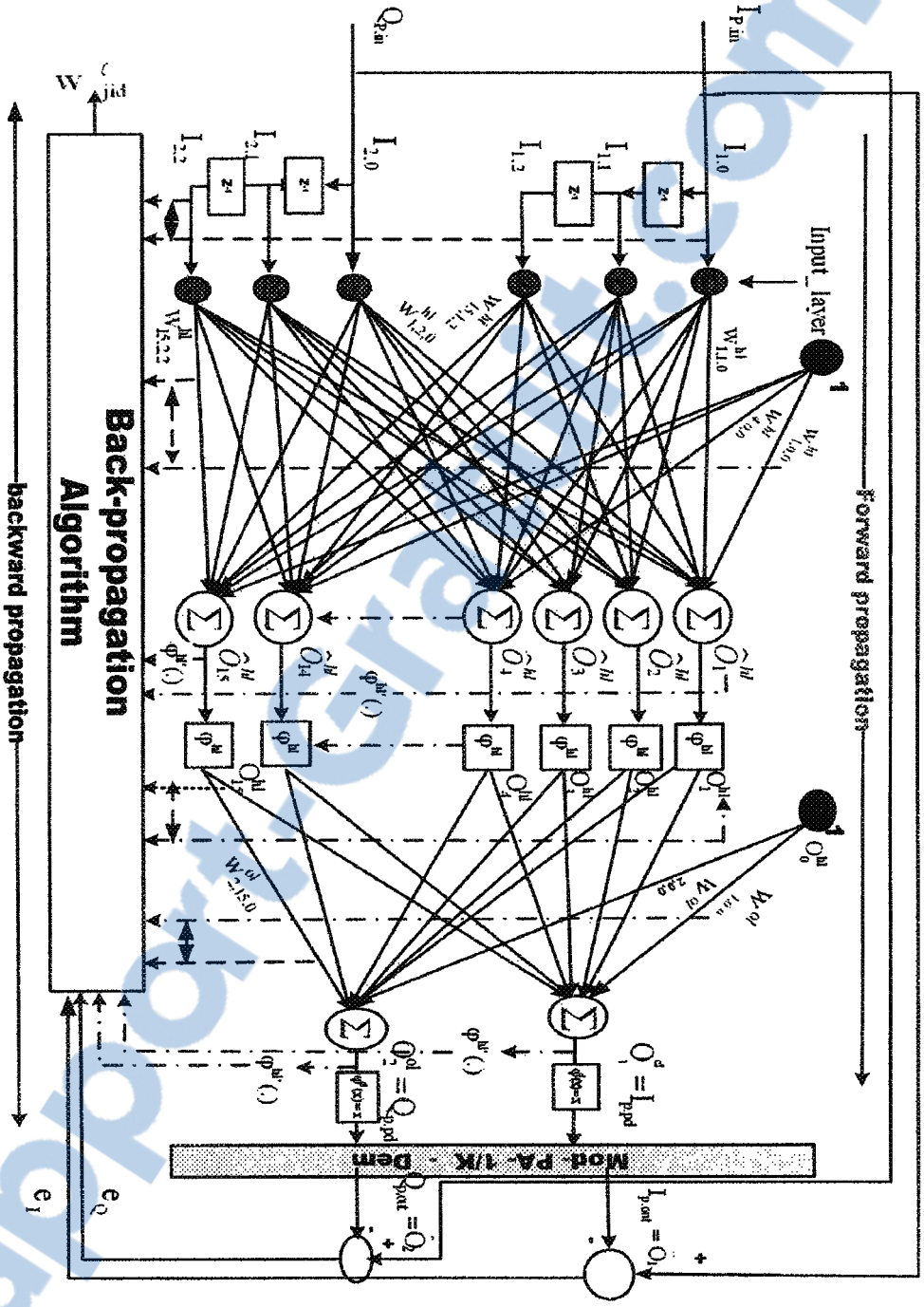


Figure 2.13: Proposed RVTDNN predistortion structure : the DPD is upstream the PA and its accessories

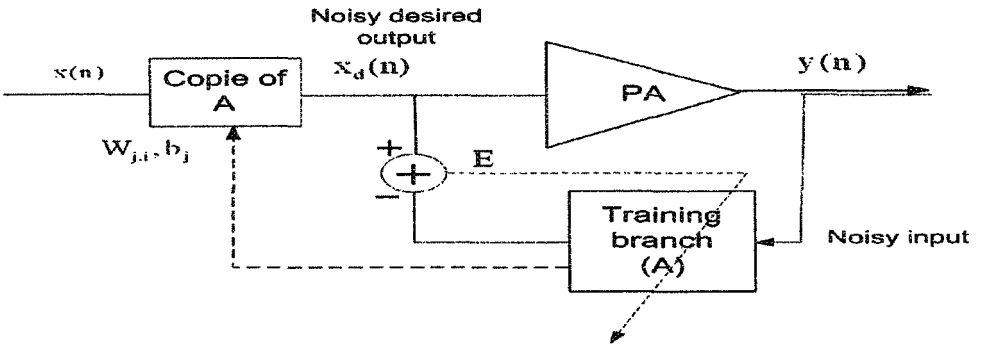


Figure 2.14: Indirect learning architecture principle

can be categorized as a direct learning architecture, because it is not necessary to identify the inverse function of the PA before starting the emission process [73] like in the case of the indirect learning architecture [28]. In the indirect learning architecture, data on-line predistorter identification is a hard task and obliges to use data off-line adaptation process to avoid complexity.

Moreover, as shown in figure 2.14, the measurement of the PA output and the desired output of the predistorter could be noisy, which can result in a convergence of the algorithm to biased values. In fact, placing a predistorter copy obtained with a post-identification structure upstream of the PA does not guarantee a good PA inverse model [73].

It has been retained the same NN architecture used in the PA modeling, the flow of information signal of figure 2.13 can be summarized, like in section 2.6, with equations 2.24 and 2.25. The specification to be added here is that the NN outputs are no longer decision points for the calculation of the error signal but the output of the whole system. The NN output layer neurons provide the predistorted version of the signal $I_{p,pd}(n)$, and $Q_{p,pd}(n)$ that corresponds to equations 2.24 and 2.25 respectively. By applying the back-propagation

algorithm, we search to minimize the following cost function :

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(\sum_{j=1}^{2}(I_{j,0}(n) - \ddot{O}_j(n))^2) \tag{2.48}$$

Equation 2.31 is now evaluated like this :

$$\frac{\partial E(n)}{\partial w_{j,i,0}^{ol}(n)} = \frac{\partial E(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial \ddot{O}_j(n)}\underbrace{\frac{\partial \ddot{O}j(n)}{\partial O_j^{ol}(n)}}_{\xi}\frac{\partial O_j^{ol}(n)}{\partial w_{j,i,0}^{ol}(n)} \tag{2.49}$$

All parts of equation 2.49 are evaluated in the same way as in section 2.6 except for the critical part ξ. We take the derivative of $\varphi(.)$=tanh(.) as explain previously (this derivative function is detailed in equation 2.38). The remaining steps are identical to those of the modeling process. The local gradient calculated at the output layer neurons is :

$$\delta_j^{ol}(n) = -\frac{\partial E(n)}{\partial \ddot{O}_j^{ol}(n)} = -\frac{\partial E(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial \ddot{O}_j^{ol}(n)}\frac{\partial \ddot{O}_j^{ol}(n)}{\partial O_j^{ol}(n)} = e_j(n).\varphi'(O_j^{ol}(n)) \tag{2.50}$$

2.7.2 Simulation results using 16-QAM test signal

The input and output signal constellations mapping in the complex plan and the resulting **AM/AM** and **AM/PM** conversions of the **PA** without predistortion block, are the same as in figures 2.9 and 2.10. Figure 2.15 presents these conversions and signal mappings after linearization with the **RVTDNN**. It is clear that with this predistorter, both nonlinear and memory effects are well compensated at the same time. **AM/AM** conversion curve of the linearized **PA** approaches a

linear curve, and the AM/PM conversion curve approaches zero. This is true for all the amplitude range, except for weak values of the input signal, for which the signal amplitude is small and the phase shift is more significant than those of large amplitude signal, as shown in figure 2.15 (AM/PM conversion curve after linearization). This phase shift will not affect the BER of the system [56].

The PA linearization is operated with an input back-off of about -6 dB from the normalized value of the input power. The symbol rate from the random source is 1 Msymbol/s. These symbols are pulse shaped with a normal pulse shaping filter with a roll-off factor $\beta = 0.3$. Because PC's simulations are done, signals are upsampled with an up-sampling factor $S_{factor} = 8$ before be pulse shaped. According to equation 2.8, the modulated signal bandwidth is B=1.3 MHz.

Memory predistorter provides a predistorted signal version to the PA, according to the memory nonlinearity learned to the NN during the training process, which compensates the PA undesirable effects. The linear gain of the PA is not presented on the signal constellation mapping diagrams because it value is compensated voluntary for results presentation tasks.

Figure 2.16 presents the PA output power spectral density. We demonstrate an almost complete cancelation of in-band distortion (EVM) and about 25 dB improvement, in spectral spreading suppression (ACPR), is achieved at a frequency offset of 350 kHz from the upper side frequency of the useful bandwidth. This ACPR suppression can be improved by adding neurons in the hidden layer. The inverse NN model of the PA is obtained with the NN realized in XSG software. We also mention the linearization results obtained with RVNN where the tapped lines are omitted; this kind of NN can not compensate memory effects

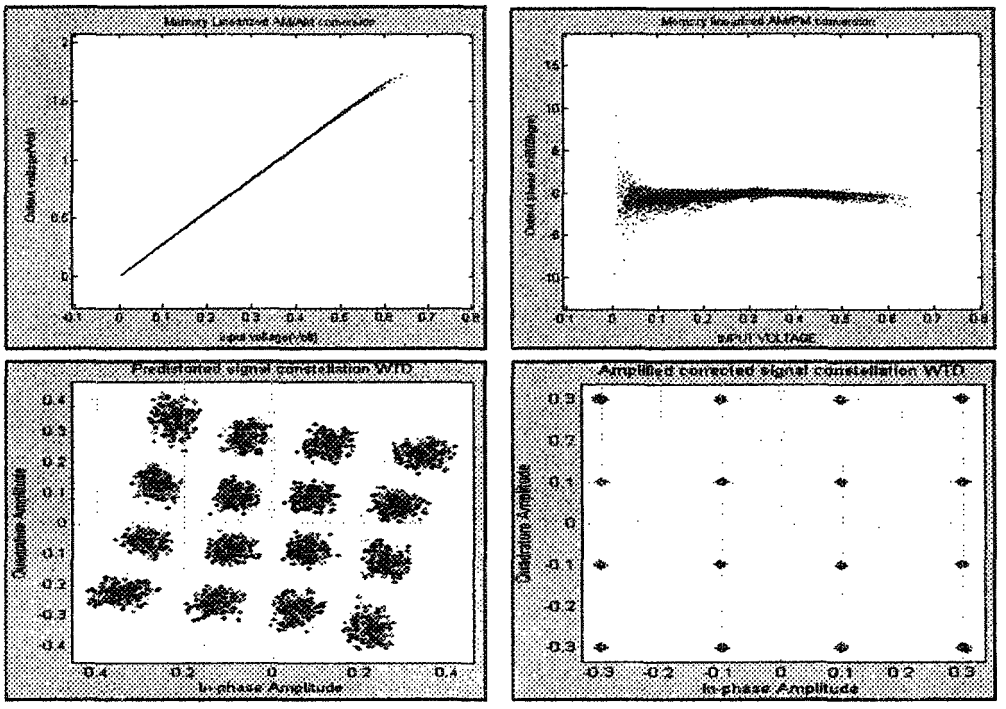


Figure 2.15: PA AM/AM and AM/PM conversions with time delayed (WTD) NN linearization (Top). PA input and output signal constellation with time delayed (WTD) NN linearization (Bottom).

and we can see that the RVTDNN achieve a better ACPR suppression.

When MSE convergence curves approach a small value, they necessarily oscillate more and more as they get smaller. This specificity is mainly a consequence of the choice of the learning rate μ [50]. Also, in the proposed work, the learning data samples are obtained from a random source, the MSE of each learning epoch is not identical to that of other epoch ; it is the reason why we noted in the convergence curves in figure 2.17 that the MSE in the $(n+1)^{th}$ epoch can be greater than the MSE of the $(n)^{th}$ epoch.

Unlike the indirect learning architecture that requires two (02) processors,

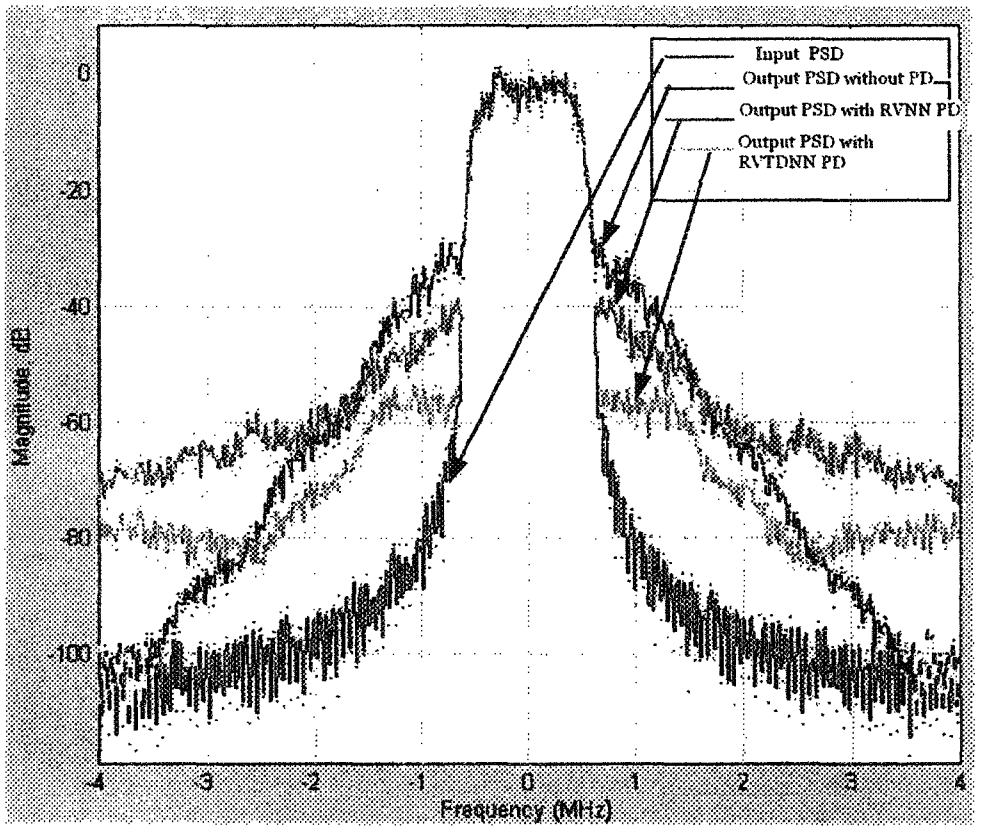


Figure 2.16: Power spectral density (PSD) of the PA input, PA output, without linearization, with RVNN and with RVTDNN predistorter

this approach needs only one processing unit (FPGA). For the convergence speed evaluation of the proposed NN architecture, it has been also obtained the inverse function of the PA with the indirect learning architecture like some other works [28, 54]. When the C_{rate} according to equation 2.23 is calculated, we find that with the proposed approach $C_{rate} = \frac{|2\times10^{-5}-6\times10^{-4}|}{0.03-0.0005} = 0.0196\ \ \mathrm{V}^2.s^{-1}$, and with the indirect learning architecture $C_{rate} = \frac{|2\times10^{-5}-5\times10^{-4}|}{0.03-0.0005} = 0.0160\ \ \mathrm{V}^2.s^{-1}$. The convergence speed seems ultra rapid, but, the period of the signal is ($1.25 \times 10^{-7}s$), and we find that the data sample number is great $N_s = \frac{5\times10^{-2}}{1.25\times10^{-7}} =$

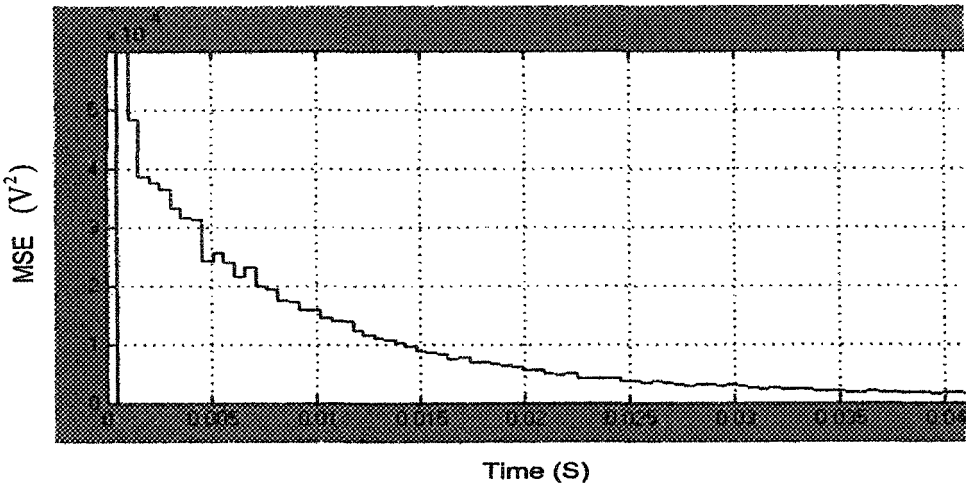


Figure 2.17: MSE convergence curve of the proposed architecture.

400000 sample, and the number of the training epochs is $N_{te} = \frac{400000}{4096} \simeq 97$ epochs (4096 is the number of the training samples in each training epoch). It is clear that our approach can minimize the MSE without significant effects on the training rate. This convergence rate is not an exact characteristic, because the source of information is random and the initial MSE is not the same. Figure 2.18 shows the convergence curves when using the indirect learning architecture to linearize the PA in conditions of our work.

In figure 2.19, we present instantaneous diagrams of the output signal constellation and the power spectral density taken during the adaptation process after perturbing the parameters α_a and α_ϕ of the memoryless nonlinearity subsystem with 5%. This data is taken at different moments (at : t=0 s, t= 0.2 s, t=0.5 s) and after meeting the stopping criterion at $t = 1s$.

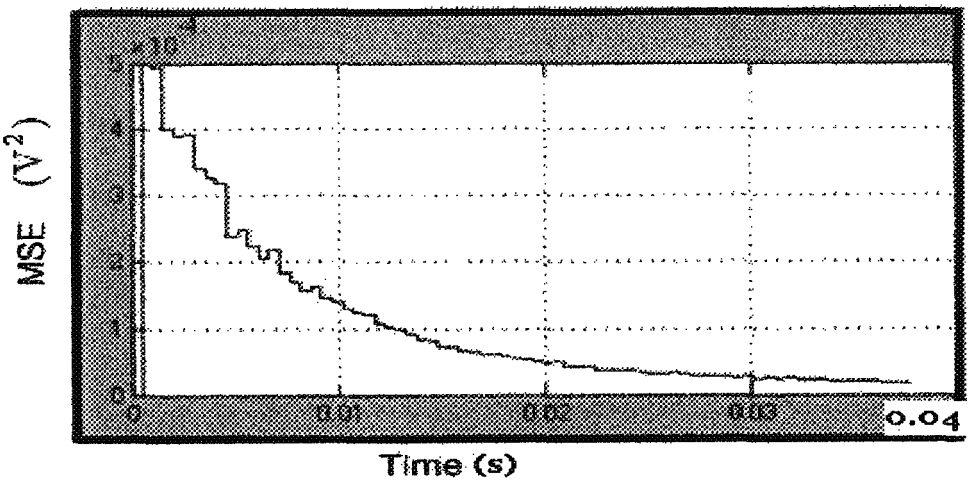


Figure 2.18: Convergence speed of the indirect learning architecture used in the conditions of our work

2.7.3 Test of the architecture with other PA model and modulated signal

To demonstrate that the proposed architecture can operate with other PA type or modulated signal without changing the general architecture, it has been tested with a solide state power amplifier (SSPA) model with 16-QAM modulation. An other simulation with 8-PSK modulated signal is achieved. SSPA model assumes that $\Phi[r(t)] = 0$ in equation 2.13 [26], that the output phase is not affected by the PA nonlinearity.

The resulting input and output signal constellation before and after linearization using a 16-QAM modulated signal, are shown in figure 2.20. With a 8-PSK signal, the results before and after linearization are shown in figure 2.21.

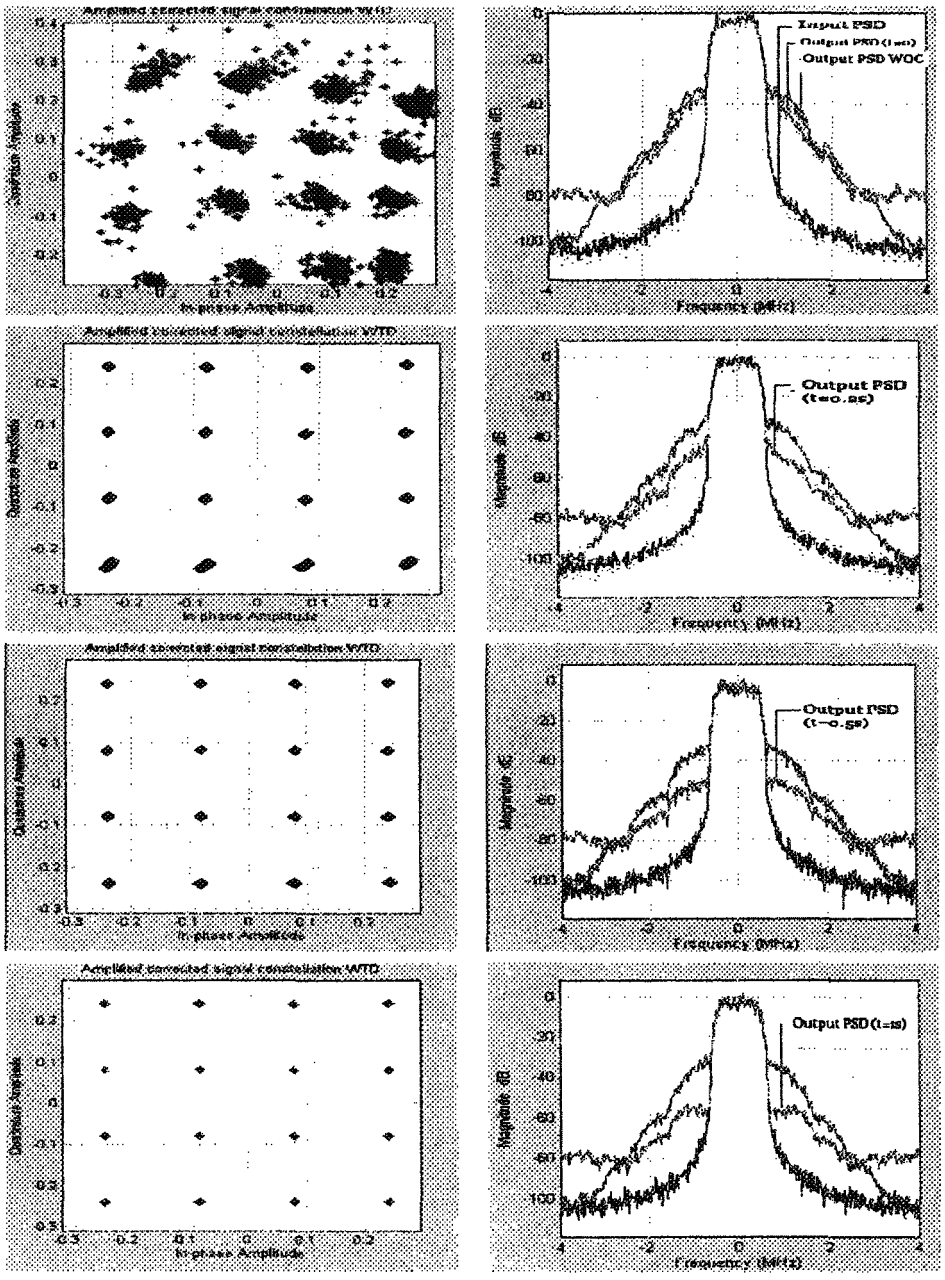


Figure 2.19: Output signal constellation and power spectral density during adaptation process (t=0 s (Top), t= 0.2 s and t=0.5 s (middles), and t=1 s (Bottom)).

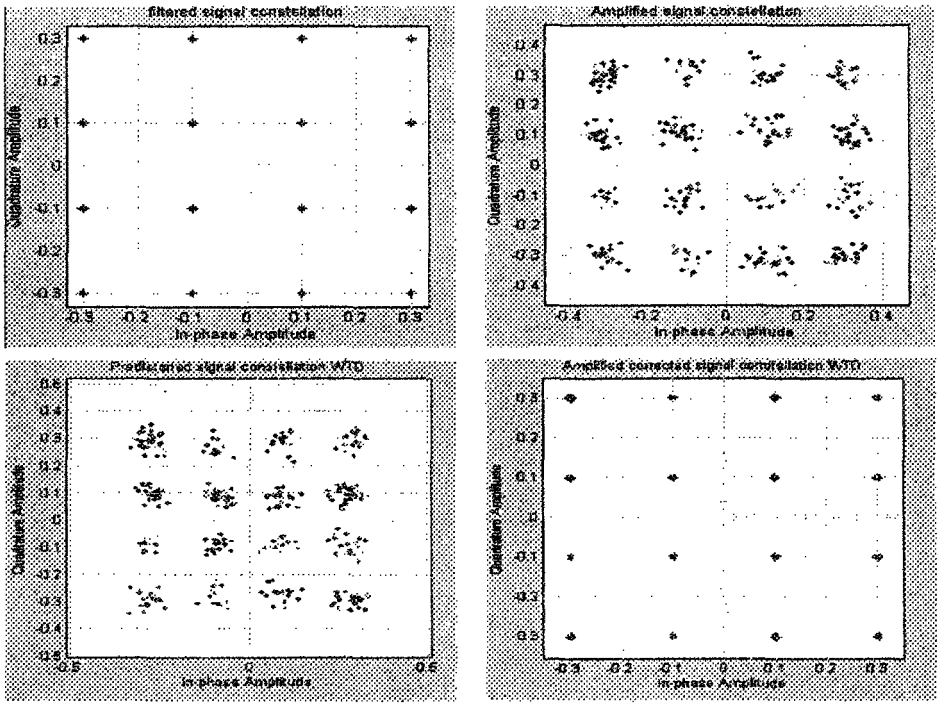


Figure 2.20: SSPA input and output 16-QAM signal constellation without predistortion (Top) and with predistortion (Bottom).

2.7.4 Discussions and conclusion

First, the proposed method is compared with most resembling works in term of assumptions of work and results presentation. The proposed DPD was not tested for 3G and 4G realistic signals, but in some works [28, 41, 42], it was used and it gave a satisfactory results (about 30 dB cancelation in ACPR). The work of Qian and Zhou [54] has treated the linearization based on CVTDNNs that suffers from local minimum convergence problems and cumbersome calculations. For the memory predistorter, the input signal is at about -8 dB input back-off with a bandwidth B=0.4 MHz, it was achieved a quasi-total suppres-

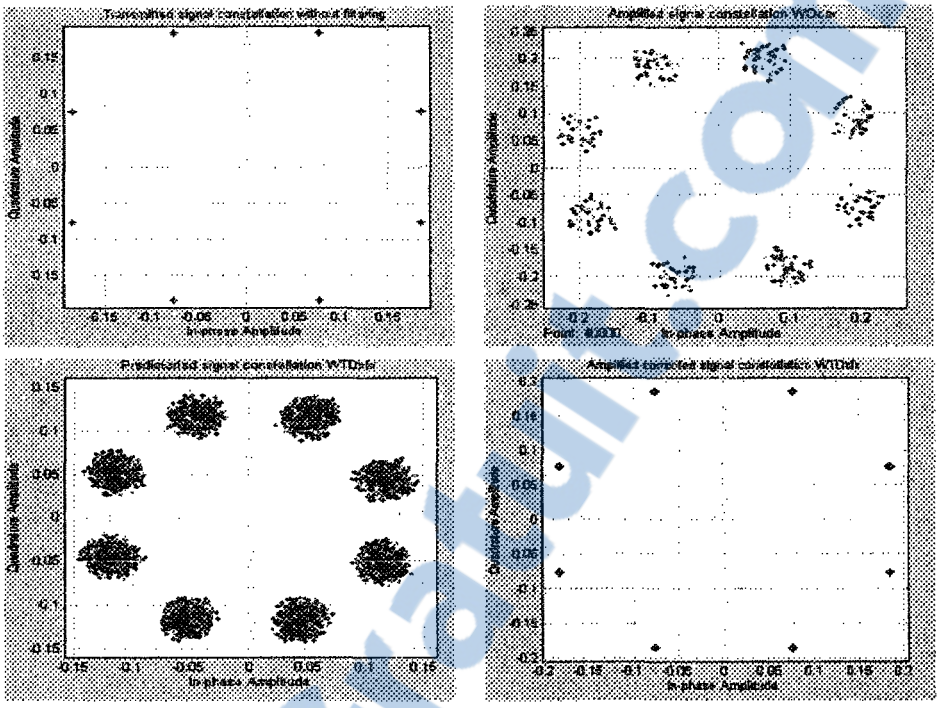


Figure 2.21: PA input and output 8-PSK signal constellation without predistortion (Top) and with predistortion (Bottom)).

sion (about 35 dB suppression in ACPR) of non-linearity effects with 25 neurons in the hidden layer, and it was noted that the memory-less predistorter can achieve only about 10 dB with aggravation of the spectral regrowth in the alternate channel. The obtained results show that the proposed approach gives practically the same performances. When speaking about the cost function MSE minimisation, the proposed NN can suppress about $1 \times 10^{-6}\ V^2$ from MSE after ($\frac{1s \times 8 \times 10^6}{4096}$=2000 training epochs), where 4096 is the number of samples in the training set. Comparing that with the Nat-GD proposed by Abdulkader *et al.* [1], it can achieve for a memory channel an MSE suppression of $1 \times 10^{-6}\ V^2$ after about 20,000 iterations (training epochs) and more than 450×10^4 iterations

with the implemented exemple cited by Langlet *et al.* [39]. Both cited architectures that are based on two independent **NNs**, one for the compensation of the **AM/AM** and the other for the **AM/PM**. These architectures are suffering from the same problems and constraints cited previously.

This work proposed method of linearization does not need to change the architecture of the **NN** when changing the signal type and when changing the **PA** type as shown in subsection 2.7.3.

The present work constitutes a contribution that improves the use of **RVTDNNs** architectures to linearize **PAs**, where their main performances are already demonstrated in several works [28, 41, 42]. A linearizing architecture like the proposed one is able to compensate for in-band and out-of-band distortions, it can also compensate memory effects on wide-band modulated signals. The frequency band of the linearization is dependent on the digital processor performances.

To our best knowledge, it is the first time that a similar architecture is proposed : it uses only one processor block, in contrast of the indirect learning architecture which needs two processors. Achieving a continuous correction without interrupting the transmission process with the indirect architecture is a hard task compared to the proposed approach, and the risk of local minimum convergence is high. Also, with the proposed approach, the stopping criterion of convergence is verified every epoch. If the time constant of long term memory effects, which requires an adaptive correction, is over hours, the adaptation process begins to operate in the next epoch after detecting that the **MSE** exceeds the specified acceptable value $\mathbf{MSE}_{acc}$. In a manner, the initial error to be propagated through the **NN** after dissatisfaction of the stopping criterion

is very small and a short-time is sufficient to get newly the convergence point. Consequently, at the receiver, the number of erroneous bits in the restituted information will be minimal. The MSE convergence curves can represent the EVM correction speed, and we note that this value is rapidly minimized in time, so the BER is obligatory very small.

The convergence of this algorithm is demonstrated with quasi-real conditions, such as simple point precision digital representation, real multipliers, adders and block of memory ressources. The XSG software serves to implement this ressources on appropriate FPGAs in term of chip area and computation time.

2.8 Neural network architecture implementation with XSG software

In this section, we present the proposed NN architecture implementation with XSG software which will be implemented on FPGA board in future research work. This architecture is designed for applications with Xilinx FPGA circuits, and this software gives the opportunity to realize applications with very high integration level. It has been adopted as a nonlinear activation function of the hidden layer neurons the tangent hyperbolic function `tanh(.)`. In digital systems, there are many kinds of representation for data, values of synaptic weights, biases. Input and output signals are real-valued and they can be presented even in digital or analog forms. When using digital representation, signal values can be presented on floating or fixed-point format and they can be serial or parallel. Ideally, design with floating point precision is more flexible and easy for manipulation. However, this advantage has a price in term

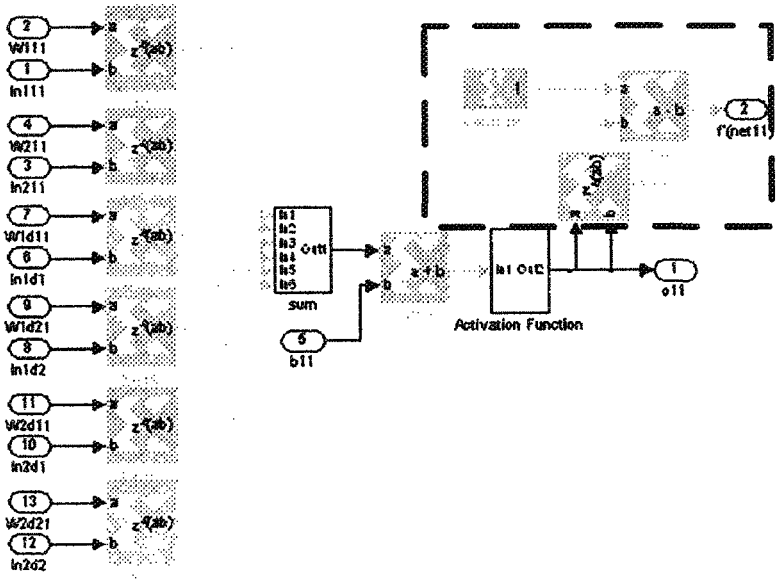


Figure 2.22: Hidden layer neuron realization with XSG software.

of silicon surface and number of input/output pins. For hardware implementation and because of the high price of arithmetic with floating point precision, it is necessary to use the fixed point simple precision presentation with two's complement Fix[30-28] format which can provide a precision of about 4×10^{-9} on the parameter variations and data are between -1 and 1. The work frequency of the FPGA board depends on the signal frequency and the rapidity of the ADC/DAC circuits. Because Matlab uses double precision presentation, the XSG software provides the use of blocks Gateway-in and Gateway-out like communication point between the FPGA and Simulink parts. The general presentation of data with simple fixed point precision consists of two parts : signed or unsigned numbers. Two's Complement Fixed Point Format can even represents negative and positive numbers [71].

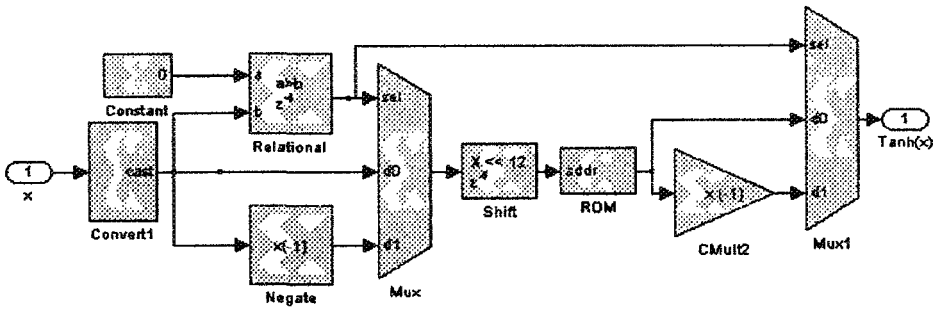


Figure 2.23: The activation function implementation with XSG software.

It will be explained all realized modules and the principle of operation of each module. The NN is realized in modular configuration. There are four main modules : forward propagation, local gradients calculation, decision and update module. The general view of the NN final architecture is reported in the appendix 3.2. White blocks represent neurons. Yellow block gives the instantaneous gradient of the output layer neurons and the signal of the convergence decision. The orange blocks are the updating block of the output layer neurons, and the cyan one is the updating block of the hidden layer neurons.

2.8.1 Forward propagation module

Equations 2.24 and 2.25 are programmed using the XSG software. We need to implement multipliers, adders and a subsystem for the nonlinear activation function and its derivative with XSG software. Figure 2.22 shows a neuron of the hidden layer and the framed part represents how to simply obtain the activation function derivative of the neurone output according to equation 2.38.

The activation function implementation can be done using the schema proposed by Peter *et al.* [53], but it seems a very complex approach. the realized

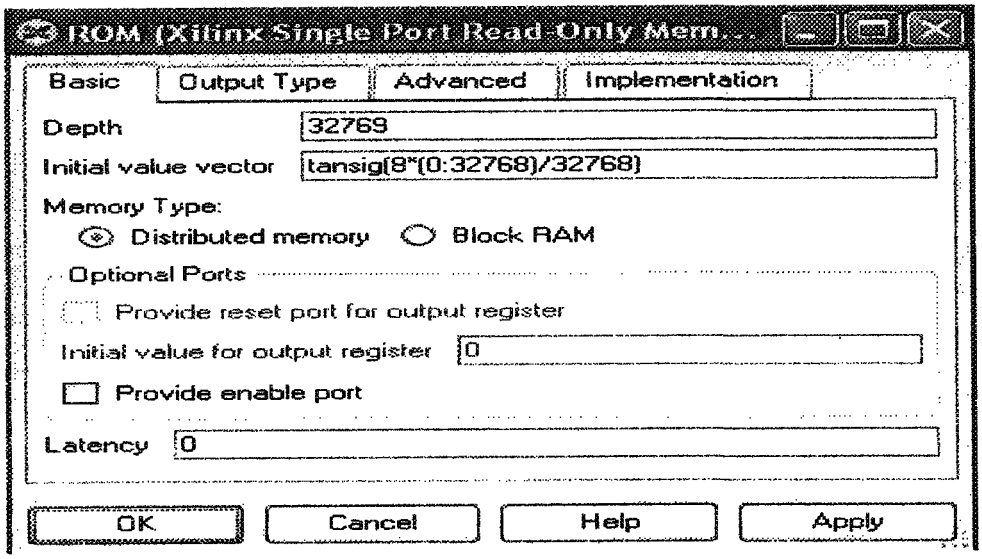


Figure 2.24: The ROM Block command window.

activation function is presented in figure 2.23. This function implementation was inspired from that of the *logsig*(.) function realized by Bastos *et al.* [6]. It is improved by using a shifter instead of a multiplier, because shifters are free sources and they can perform a left or right shift on the input signal for multipliers or dividers implementation [49].

This scheme is using the advantage from the symmetric characteristic of this function to realize only the positive part and the other part is obtained indirectly to avoid excessive use of memory in the FPGA board. It has been realized $32769 = 2^{15} + 1$ addressed points (including zero) in the ROM block to obtain sufficient precision. The block convert serves for converting the input sample to a number of desired arithmetic types and the multiplex block serves to choose whether the input signal is positive or negative to select the specified output value. Figure 2.24 shows the command window of the ROM block.

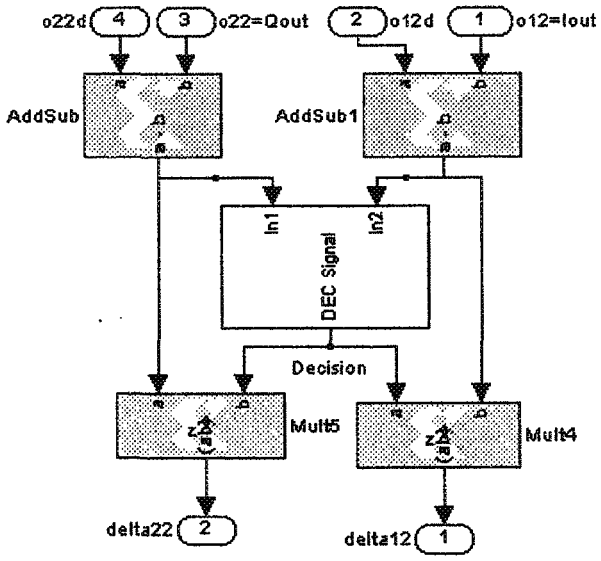


Figure 2.25: Output layer neurons local gradient for modeling

2.8.2 Backward propagation module

In the back-propagation flow of information there are three blocks to be realized : local gradient of the output layer neurons calculation, hidden layer neurons local gradient calculation, and the weights and biases parameters updating blocks. For the local gradient of the output layer neurons, equation 2.35 is considered for modeling and its implementation with XSG software is shown in figure 2.25.

Equation 2.50 serves to obtain the local gradient of the output layer neurons when linearizing and its implementation is shown in figure 2.26.

For the hidden layer neurons local gradient calculation, we refer to equation 2.43 and we proceed carefully to implement this equation. After calculating all

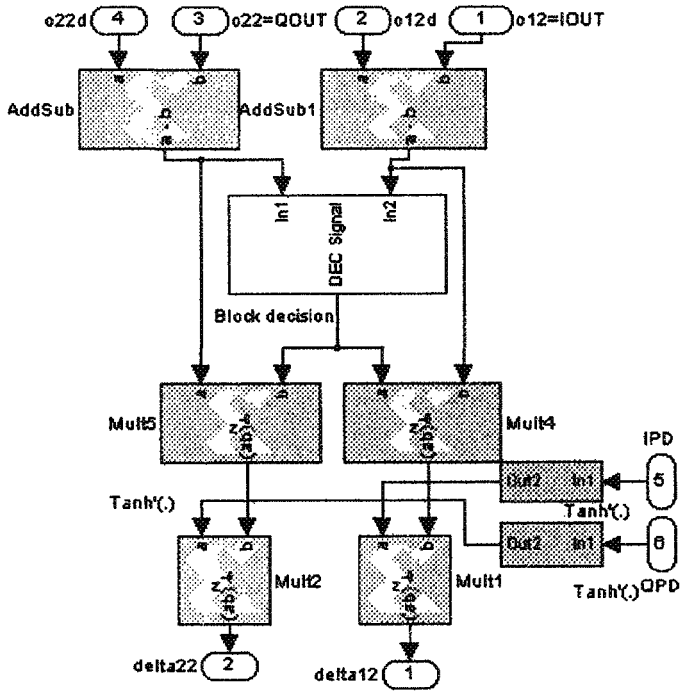


Figure 2.26: Output layer neurons local gradient for linearizing

neuron local gradients in the NN, it can be now proceeded to calculate the weight and biases changes through equations 2.44 and 2.45. Figure 2.27 shows the parameter's changes block of the hidden layer first neuron, it is like done in the work [49]. The shifter represents the optimal retained value to the learning rate $\mu = 2^{-10}$, this value is chosen because with superior values, NNs parameters are over valued and the NN diverges.

Then, the new weights and biases values are updated according to equation 2.46, and the implementation is shown in figure 2.28 [49].

The needed derivative of the nonlinear activation function $\varphi'(.)$ is given by equation 2.38 and its implementation is shown in the figure 2.29.

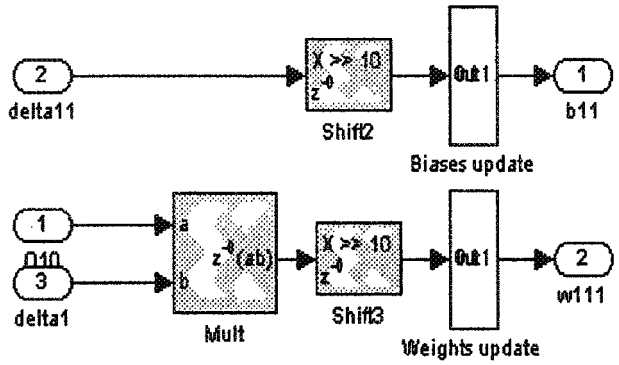


Figure 2.27: Biases changes block (Top) and weights changes block (Bottom) of the input layer first neuron

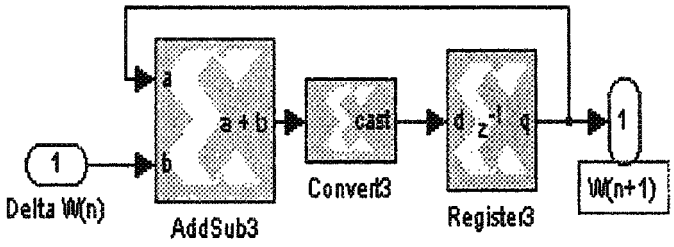


Figure 2.28: Weights and biases updating block

2.8.3 Decision module

The decision module is the source of the command that will finish the back-propagation after convergence. It is based on equations 2.28 and 2.29 or 2.48. Figure 2.30 shows the block to calculate the MSE of the NN. After every epoch training, a command signal will be sent to test the state of inequality 2.47.

In figure 2.31, we present the block that verifies whether the inequality is satisfied or not. Zero is sent to the local gradient block of the output layer neurons if the stopping criterion is satisfied; if not, 1 is sent and back-propagation

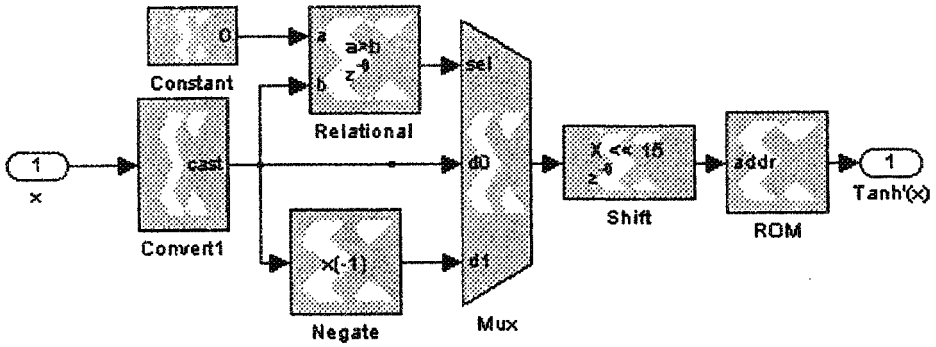


Figure 2.29: Implementation of the nonlinear activation function derivative.

continues to propagate the error signal.

2.8.4 Discussions and conclusion

Various techniques have been used to implement NNs on FPGAs. Some designers implement directly the full NN, including the nonlinear activation function, multipliers, etc., to profit from FPGA high speed. This technique needs very large silicon surface. As a second solution, they use time division multiplexing and share adders multipliers between several neurons [53]. In the case of our work, objectives are to get wide-band operation for the baseband signal. To do that, it has been maintained a fully parallel structure.

The NN architecture implementation using XSG software was necessary when evaluating the performances of our method of linearization. It is interesting, for future research, to optimize and implement the proposed linearization method on FPGA boards. The most recent researcher who succeeded to implement a NN architecture with on-chip training is Minhui [49], in 2006, but with

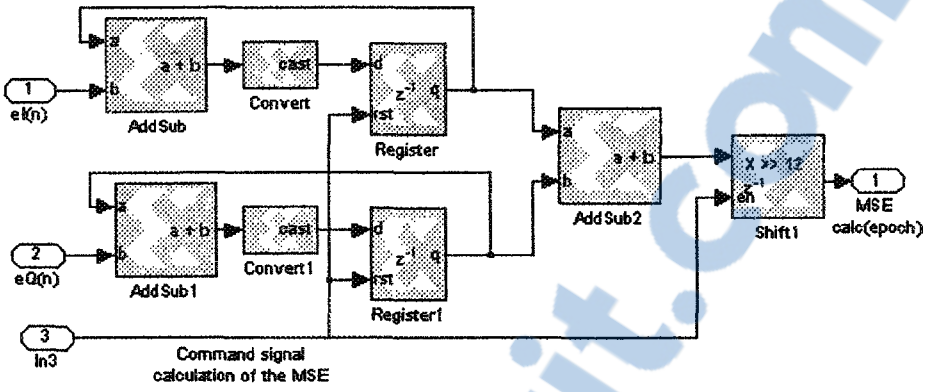


Figure 2.30: MSE calculation block.

a relatively simple architecture (6 inputs, 6 neurons in the hidden layer and 3 outputs). Our NN is more complex (2-15-2), with 2 tapped lines at each input; this NN is also designed for wireless communication systems applications (high frequency), requiring more advanced FPGA boards.

2.9 General conclusion and future works

The major constraint in the modern wireless communication systems are due to the PA imperfections. PAs can not achieve high power efficiency (70-90%), with acceptable linearity (10-15 dB in ACPR emission). Furthermore, with wide-band modulation techniques, memory effects degrade the quality of the service and introduce more spectral regrowth and higher ISI. Predistortion needs just one block predistorter at the transmitter side, not like equalization where all receiver systems must have their own equalization block, resulting in a hard task for synchronization and parameters ajustements. Predistortion does not have stability and synchronization problems, it is promoted by the large

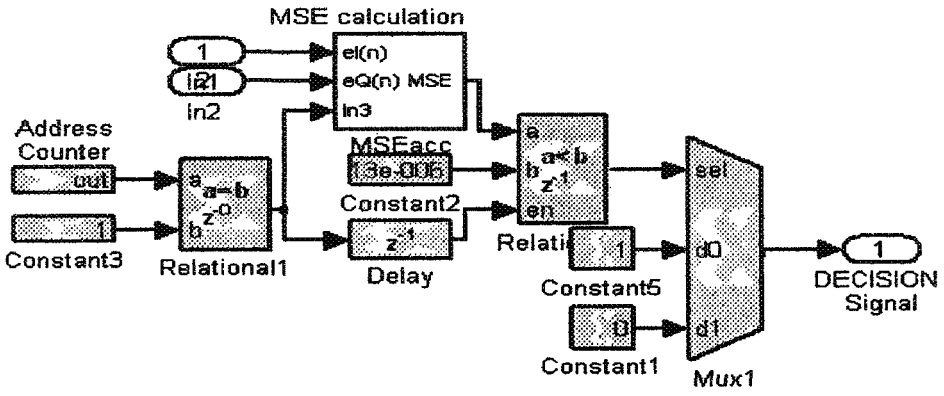


Figure 2.31: Inequality 2.47 verification principle to decide on the algorithm convergence. Block in white represents the MSE calculation of figure 2.30.

use of digital processors at the transmitter, to achieve filtering, modulation, etc.

In this work, we propose a novel architecture that can achieve, at the same time, the correction of nonlinearity effects and compensate short term memory effects. Adaptive linearization can compensate long term memory effects. Only one processor block is sufficient with the proposed DPD and it is effectively based on a pre-identified version of the PA inverse function, unlike the indirect learning architecture that needs two separate processors and is based on a post-identified inverse function. Moreover, the indirect learning architecture has two main drawbacks affecting its performance. First, the measurement of the PA output and the desired output of NN could be noisy, which can involve a convergence of the algorithm to biased values ; second, the placing of a copy of the predistorter structure in upstream of the PA does not guarantee a well inverse model obtention [73].

Artificial neural networks are high technological tools with good perfor-

mances, that promise advanced solutions in the future. The proposed linearization method is able to compensate out-of-band and in-band distortions at the same time and it is suitable for any kind of PA or base station using digital modulated signal such as CDMA2000 and W-CDMA [41, 28]. Moreover, because Mod./Dem. and ADC/DAC are in the corrected loop of the proposed architecture, their imperfections will be automatically taken into account. Also, our architecture provides a signal predistortion that operates with baseband signal; then, it is easy to up-convert it to any frequency range simply with quadrature (vectoriel) modulators. Our predistorter can work in any frequency band in function of the information symbol flow source, the working frequency of the ADC/DAC and the DSP adopted. The obtained results demonstrate a 25 dB continuous cancelation in ACPR at 350 kHz frequency offset with a 16-QAM test signal. The adopted PA model operates at about 6 dB input back-off. A quasi-total compensation of memory effects is achieved at the same time and the adaptive process operates with acceptable speed.

The future development of the proposed architecture must take into account the following considerations :

- The imperfection effects of the Mod./Dem. on the predistortion;
- The local oscillator leakage effects;
- The noise effects;
- The implementation of the NN on FPGA;
- The development and analysis of the training algorithm (using Nat-Gr);
- The augmentation of the number of neurons in the hidden layer to minimize the ACPR;
- The experiments with other type of modulation and PA models;
- The experiments of the influence of the modulation and filtering para-

meters such as the bandwidth and the effect of the roll-off factor ;

- The experiments for memoryless nonlinearities, with a low number of neurons at the beginning, then a linearizer with memory effects can be developed ;
- The evaluation of the effects of the FPGAs resources precision like multipliers, adders, and the sensitivity for weight errors [61] ;
- The use of distributed time delays NNs (Matlab Neural Networks toolbox), where the tapped delay lines are distributed through the NN ;
- The realization of a static linearizer by using NNs without multipliers to reduce FPGA ressources requirements [45].

CHAPITRE III

CONCLUSION GÉNÉRALE

Les systèmes modernes de communication subissent une contrainte majeure due aux imperfections et aux effets indésirables causés par les amplificateurs de puissance RF. Les circuits AP n'ont pas la possibilité d'avoir simultanément une grande efficacité spectrale (70-90%) et une linéarité acceptable (10-15 dB de ACPR). Avec l'emploi des techniques de modulation à large bande, les effets mémoire dégradent la qualité du service. Pour remédier à ce problème, il est opté pour la linéarisation par la prédistorsion adaptative en bande de base, nécessitant un seul bloc de prédistorsion intégré dans le transmetteur, au lieu d'une égalisation dans chaque récepteur.

Dans ce travail, il est proposé une nouvelle architecture capable de corriger simultanément les effets des non-linéarités et de compenser les effets mémoire à court-terme. La prédistorsion adaptative peut compenser les effets mémoire à long-terme. Un seul bloc processeur est suffisant pour l'approche proposée dans ce travail, qui est basée effectivement sur une version pré-identifiée de la fonction inverse du circuit AP. Par opposition, l'architecture d'apprentissage indirect nécessite deux blocs processeurs et elle est basée sur des versions post-identifiées de la fonction inverse du circuit PA. Ainsi, l'architecture indirect a deux inconvénients majeurs [73] : 1) la mesure de la sortie du circuit PA et la

sortie désirée de l'élément AP peuvent être bruitées, ce qui peut provoquer la convergence de l'algorithme à une valeur polarisée, 2) les prédistorteurs non-linéaires ne peuvent pas être permutés, c. à d., le prédistorteur identifié est en réalité un prédistorteur à post identification. Alors, l'utilisation d'une copie de sa structure, en amont du circuit PA, ne garantie pas une fonction inverse performante.

Les réseaux de neurones artificiels constituent un outil performant d'une technologie versatile, donnant des grandes promesses sur leur apport dans le domaine des linéarisateurs. La méthode de linéarisation proposée est capable de compenser les distorsions simultanément à l'intérieur et à l'extérieur de la bande utile du signal. Elle est valable pour n'importe quel type de circuits AP ou stations de base qui utilise des signaux modulés comme CDMA2000 et W-CDMA [41, 28]. Aussi, parce que les circuits Mod./Dem. et les circuits CNA/CAN font partie de la boucle de correction de l'architecture proposée, leurs imperfections sont automatiquement prises en compte. L'architecture de prédistorsion proposée opère avec des signaux en bande de base, qui seront modulés pour être transmis dans n'importe quelle bande de fréquence. La méthode de prédistorsion proposée peut alors opérer avec n'importe quelle fréquence en fonction du débit de la source d'information à transmettre, la fréquence de travail des circuits CNA/CAN et la fréquence de travail du processeur numérique. Les résultats obtenus avec un signal de test 16-QAM démontrent une suppression de 25 dB sur le paramètre ACPR à 350 kHz d'offset en fréquence à partir de la limite de la bande de fréquence utile avec le circuit AP fonctionnant à 6 dB de recul de puissance d'entrée. Une compensation quasi-totale des effets mémoire est réalisée en même temps. Le processus d'adaptatbilité fonctionne avec une qualité de convergence acceptable. Les résultats obtenus montrent que les réseaux de

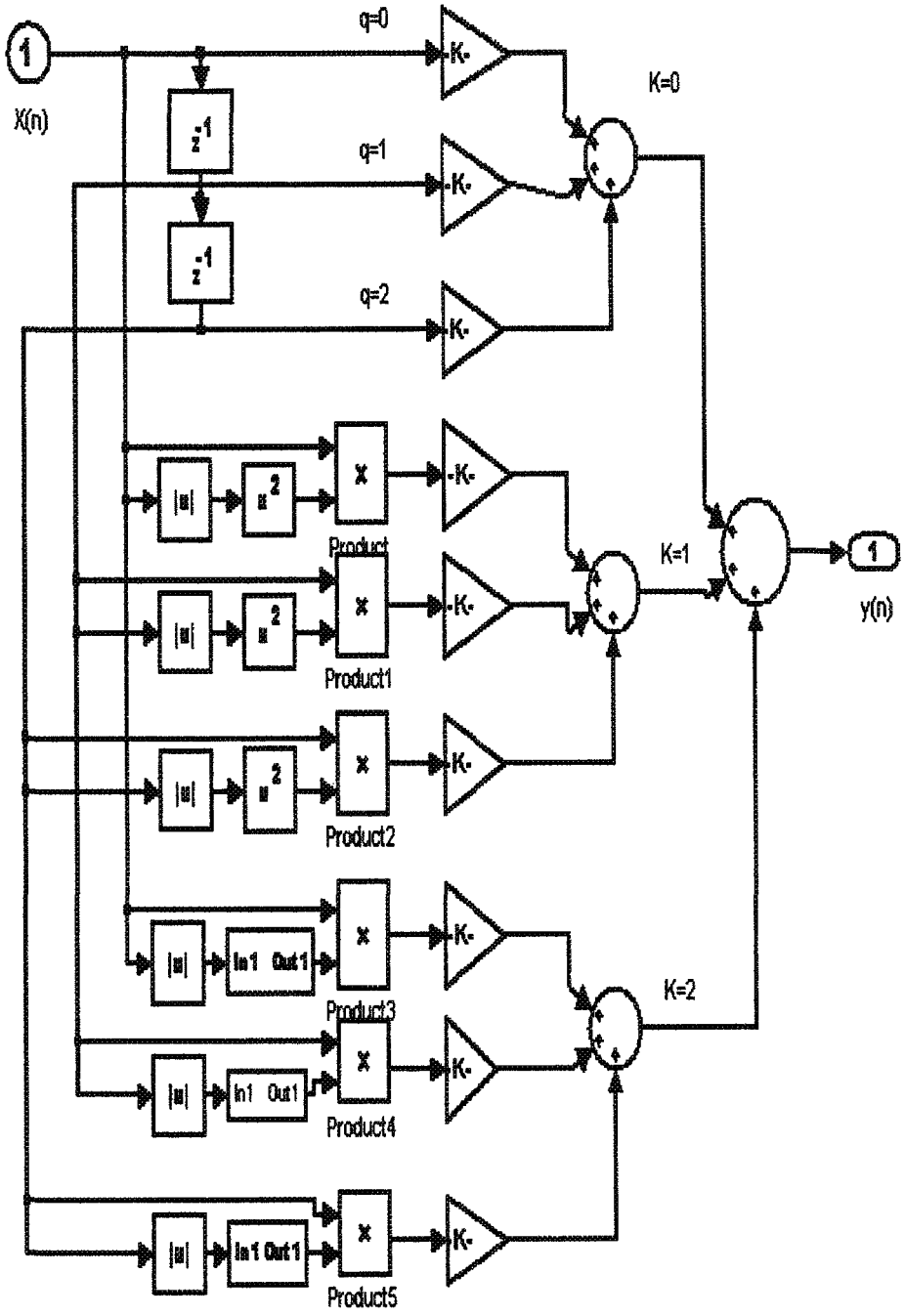
neurones sont très prometteurs dans le domaine de la linéarisation des circuits AP.

Le développement futur de l'architecture proposée devrait tenir compte des considérations suivantes :

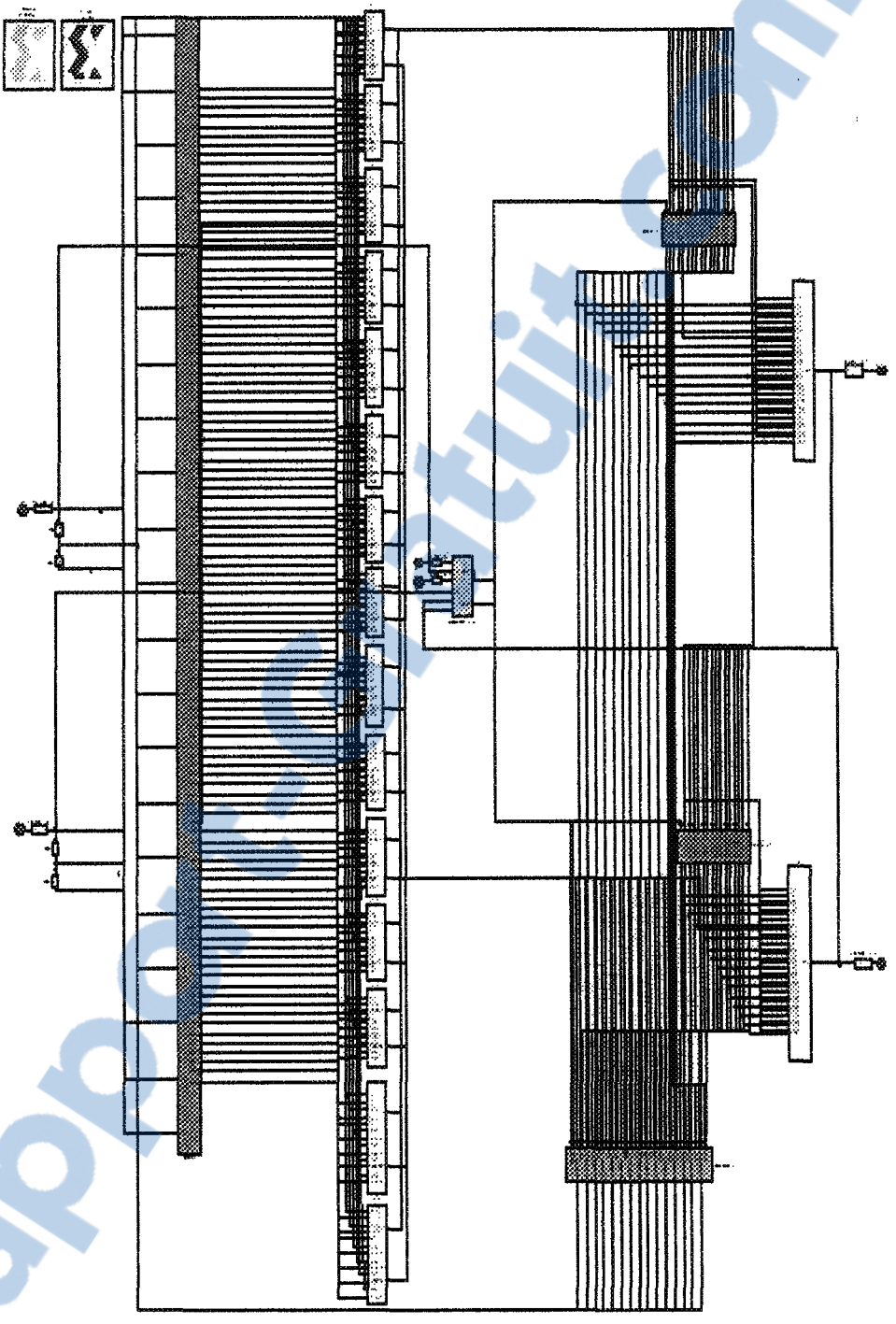
- Les effets des imperfections des circuits Mod./Dem. sur la prédistorsion ;
- Les effets de fuite de l'oscillateur local ;
- Les effets des bruits ;
- L'implantation du réseau NN sur circuit FPGA ;
- Le dévelopment et analyse de l'algorithme d'apprentissage (utilisation du Nat-Gr) ;
- L'expérimentation avec d'autre type de modulation et modèles du circuit AP ;
- L'expérimentation de l'influence des paramètres de la modulation et des filtres comme la largeur de bande et les effets du facteur de décroissance «roll-off factor» ;
- L'expérimentation pratique par l'implantation du réseau NN pour corriger des non-linéarités sans mémoire, avec un petit nombre de neurones dans la couche cachée au début. Il peut également être réalisé pour une non-linéarité avec mémoire ;
- Augmenter le nombre de neurones dans la couche cachée ;
- L'evaluation des effets de la précision des ressources du circuit FPGA comme les multiplicateurs, les additionneurs et la sensibilité des erreurs des poids [61] ;
- L'utilisation des réseaux NN avec des retards distribués (Matlab Neural Networks Toolbox), qui sont caractérisés par la distribution des retards à travers le réseaux de neurone ;

- La réalisation d'un linéarisateur statique en utilisant des réseaux NN sans multiplicateurs [45] afin de minimiser le nombre de ressources requises dans le circuit FPGA.

ANNEXE



Annexe 3.1: Polynomial memory model of PA realized by Simulink



Annexe 3.2: General view of the final Architecture of the NN.

RÉFÉRENCES

[1] H. Abdulkader, F. Langlet, D. Roviras and F. Castanie, "Natural Gradient Algorithm for Neural Networks Applied to Non-linear High Power Amplifiers," Inter. Journal of Adaptative control signal Process., vol.16, pp.557-576, 2002.

[2] "Agence Wallonne des Télécommunications,", www.awt.be.

[3] H. Alasady, R. Boutros and M. Ibnkahla, "Comparison between Digital and Analog Predistortion for Satellite Communications," Canadian Conf. on Electrical and Computer Engineering, IEEE CCECE, vol.1, pp.183-186, Iss.4-7, May 2003.

[4] N. A. Andrea, L. Vincenzo and R. Reggiannini, "RF Power Amplifier Linearization through Amplitude and Phase Predistortion," IEEE Trans. on Comm., vol.44, no. 11. Nov. 1996.

[5] E. Aschbacher and M. Rupp, "Modeling and Identification of a Nonlinear Power-amplifier with Memory for Nonlinear Digital Adaptive Predistortion," SPAWC workshop proc., pp.15-18, Rome, Italy, 2003.

[6] J. L. Bastos, H. P. Figueroa and A. Monti, "FPGA Implementation of Neural Network-based Controllers for Power Electronics Applications," Twenty-First Annual IEEE Applied Power Electronics Conf. and Exp., Iss.19-23, pp.1443-1448, Mar. 2006.

[7] G. Baudoin and P. Jardin, "Adaptive Polynomial Pre-distortion for Linearization of Power Amplifiers in Wireless Communications and WLAN," Inter. Conf. on Trends in Communications, EUROCON, vol.1, pp.157-160, 2001.

[8] G. Baudoin and P. Jardin, "Filter Lookup Table Method for Power Amplifier Linearization," IEEE Trans. on Vehicular Technology, Iss.3, vol.56, pp.1076-1087, May 2007.

[9] N. Benvenuto, F. Piazza and A. Uncini, "A Neural Network Approach to Data Predistortion with Memory in Digital Radio Systems," Proc. of ICC'93 Inter. Conf. on Comm., pp.232-236, Geneve, May 1993.

[10] S. Bensmida, "Conception d'un Système de Caractérisation Fonctionnelle d'Amplificateur de Puissance en présence de Signaux Modulés à l'aide de Réflectomètres Six- Portes," Ph.D. Thesis, École Nationale Supérieure des Télécommunications, 2005.

[11] S. Boumaaza, "Caractérisation, Modélisation, et linéarisation par Prédistorsion Numérique des Amplificateurs de Puissance pour les Systèmes de Communication de 3G," Ph.D. Thesis, École Polytechnique de Montréal, 2003.

[12] E. Cottais, Y. Wang and S. Toutain, "Experimental Results of Power Amplifiers Linearization Using Adaptive Baseband Digital Predistortion," The European Conf. on Wireless Technology, pp.329-332, 2005.

[13] S. Dardenne, "Amélioration de la Linéarité des Amplificateurs de Puissance par Injection de Composantes Basses Fréquences et d'Intermodulation, pour des Applications de Radiocommunications Mobiles," Thèse de doctorat, Université de Poitiers, France, 2005.

[14] R. Dennis, "Satellite Communications," 4^{th}edition, McGraw-Hill, 1989.

[15] L. Ding , G. Tong Zhou, D. R. Morgan, Z. Mat, J. Stevenson Kenny, J. Kim and R. G. Charles, "Memory Polynomial Predistorter Based on the

Indirect Learning Architecture," IEEE Global comm. conf., vol.1, pp.967-971, 2002.

[16] L. Ding , G. Tong Zhou, R. M. Dennis, Z. Ma, J. S. Kenney, J. Kim, and R. G. Charles, "A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials," IEEE Trans. on comm., vol.52, no.1, pp.159-167, Jan. 2004.

[17] L. Ding , "Digital Predistortion of Power Amplifiers for Wireless Applications," Ph.D. Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Mar. 2004.

[18] http ://www.dmccormick.org/DataCompression.htm

[19] C. Eun, "Design and Comparison of Nonlinear Compensators," Ph.D. Thesis, University of Texas at Austin, Dec. 1995.

[20] C. Eun and E. J. Powers, "A Predistorter Design for a Memory-less Nonlinearity Preceded by a Dynamic Linear System," IEEE Global Telecom. Conf., vol.1, pp.152-156, 1995.

[21] C. Eun and E. J. Powers, "A new Volterra Predisorter Based on Indirect Learning Archtecture," IEEE Trans. on Signal Processing, vol.45, no.1, Jan. 1997.

[22] Y. Fang, M. C. E. Yagoub, F. Wang and Q. Zhang, "A New Macromodeling Approach for Nonlinear Microwave Circuits Based on Recurrent Neural Networks," IEEE Trans. on MTTs, vol.48, no.12, pp.2335-2345. Dec. 2000.

[23] B. C. Haskins, "Diode Predistortion Linearization for Power Amplifier RFICs in Digital Radios," Master of Science thesis, faculty of the Virginia Polytechnic Institute and State University, Blacksburg, VA, Apr. 17, 2000.

[24] S. Haykin, "Neural Networks," 2^{nd} edition, Prentice Hall, 1999.

[25] S. Haykin, "Communication Systems," John Wiley& sons Inc., 2002.

[26] W. Henghui, Z. Hao, Y. Wu and F. Suili, "Neural Network Predistosion Based on Error-Filtering Algorithm in OFDM system," Inter. Conf. on Wireless Communications, Networking and Mobile Computing, vol.1, pp.23-26, Sep. 2005.

[27] T. Hoh, G. Jian-hua, G. Shu-jian and W. Gang, "A Nonlinearity Predistortion Technique for HPA with Memory Effects in OFDM Systems," Elsevier Nonlinear Analysis : Real World Applications, vol.8, Iss.1, pp.249-256, Feb. 2007.

[28] H. Hwangbo, S. Jung, Y. Yang and C. Park, "Power Amplifier Linearization Using an Indirect-learning-based Inverse TDNN Model," Microwave Journal, vol.49, no.11, pp.76-88. Nov. 2006.

[29] X. Huang, J. Lu and J. Zheng, "A Reduction in PAPR of OFDM System Using a Revised Companding," 8^{th} Inter. Conf. on Comm. Systems. pp.62-66. 2002.

[30] M. Isaksson, D. Wisell and D. Ronnow, "A Comparative Analysis of Behavioral Models for RF Power Amplifiers," IEEE Trans. on MTTs, vol.54, Iss.1, pp.348-359. Jan. 2006.

[31] J. Jeon, Y. Shin and S. Zmi, "A Data Predistortion Technique for the Compensation of Nonlinear Distortion in MC-CDMA systems," SPAWC'99. pp.174-177, 1999.

[32] W. Jenkins and A. Khanifar, "Linearization of Power Amplifiers through Injection of Low-frequency Second-order Products," Radio and Wireless Conf., pp.237-239, 2002.

[33] W. Jenkins and A. Khanifar, "Power Amplifier Linearisation Through Generation and Injection of Low-Frequency Second-Order Nonlinear Products," 33^{rd} European Microwave Conf., vol.1, pp.277-281, 2003.

[34] J. C. Pedro, and S. A. Maas, "A Comparative Overview of Microwave and Wireless Power-Amplifier Behavioral Modeling Approaches," IEEE Trans. on MTTs, vol.53, no.4, pp.1150-1163, Apr. 2005.

[35] J. Xu, M. C. E. Yagoub, R. Ding and Q. J. Zhang, "Neural-Based Dynamic Modeling of Nonlinear Microwave Circuits,", IEEE MTT-S Inter. Microwave Symposium Digest, vol.2, pp.1101-1104, 2002.

[36] P. B. Kenington, "High-Linearity RF Amplifier Design," Artech House, 2000.

[37] J. Kim and K. Konstantinou, "Digital predistortion of Wideband Signals Based on Power Amplifier Model With Memory" Electronics Letters, vol.37, Iss.23, pp.1417-1418, 8 Nov. 2001.

[38] W. J. Jung, W. R. Kim, K. M. Kim and K. B. Lee, "Digital Predistortion using Multiple Look-up Tables," Electronics Letters, vol.39, Iss.19, pp.1386-1388, 18 Sep. 2003.

[39] F. Langlet, H. Abdulkader, D. Rovirasl, A. Malletand and F. Castanik, "Comparison of Neural Network Adaptive Predistorsion Techniques for Satellite Down Links," Inter. Joint Conf. on Neural Networks proc, pp.709-715, 2001.

[40] K. C. Lee and P. Gardner, "Adaptive Neuro-fuzzy Inference System (ANFIS) Digital Predistorter for RF Power Amplifier linearization," IEEE Trans. on Vehicular Technology, vol.55, Iss.1, pp.43 - 51, Jan. 2006.

[41] T. Liu, S. Boumaiza and F. M. Ghannouchi, "Dynamic Behavioral Modeling of 3G Power Amplifiers using Real-valued Time-delay Neural networks," IEEE Trans. on MTTs, vol.52, Iss.3, pp.1025-1033, Mar. 2004.

[42] T. Liu, "Dynamique Behavior Modeling and Pre-Compensation for Broadband Transmitters," Ph.D. Thesis, École polytechnique de Montréal, Dec. 2005.

[43] G. Maral and M. Bousquet, "Satellite Communications Systems : Systems, Techniques and Technology," Fourth Edition, Wiley 2005.

[44] J. F. Marcello, "Wideband digital predistorsion linearization of radio frequency power amplifier with memory," Ph.D. Thesis, university of Drexon, 2005.

[45] M. Marchesi, G. Orlandi, F. Piazza and A. Uncini, "Fast Neural Networks without Multipliers," IEEE Trans. on Neural Networks, vol.4, Iss.1, pp.53-62, Jan. 1993.

[46] R. *Maršálek*, "Contributions to the Power Amplifier Linearization using Digital Baseband Adaptive Predistortion," Ph.D. Thesis. Université de Marne la Vallée, 2003.

[47] "Raised Cosine Transmit Filter,", Matlab communications blockset help.

[48] M. Schiff, "Baseband Pulse Shaping for Improved Spectral Efficiency," Application Note AN.131, Elanix, Inc., Mar.3, 2000.

[49] Z. Minhui, "FPGA Implementation of a Neural Network Control System for Helicopter," Master of sciense thesis, Faculty of Royal Miltary College of Canada, Apr. 2006.

[50] M. Ibnkahla, N. J. Bershad, J. Sombrin and F. Castanie, "Neural network modeling and identification of nonlinear channels with memory : algorithms,

applications, and analytic models," IEEE Trans. on Signal Proc., vol.46, Iss.5, pp.1208-1220, May 1998.

[51] N. Naskas and Y. Papananos, "Neural Network-based Adaptive Baseband Predistortion Method for RF Power Amplifiers," IEEE Trans. on Circuits and Systems, vol.51, Iss.11, pp.619-623, Nov. 2004.

[52] I. Park, "Characterization and Compensation of Nonlinear Distortion," Ph.D. Thesis, University of Texas at Austin, Dec. 1998.

[53] I. Petra, D. J. Holding, K. J. Blow, B. Tam, X. Ma and P. N. Brett, "The Design of a Flexible Digit towards Wireless Tactile Sense Feedback," Control, Automation, Robotics and Vision conf., vol.1, pp.1468- 473, 2004.

[54] H. Qian and G. T. Zhou, "A Neural Network Predistorter for Nonlinear Power Amplifiers with Memory," Digital Signal Processing Workshop, pp.312-316, 2002.

[55] H. Qian, "Power Efficiency Improvements for Wireless Transmissions," Ph.D. thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Aug. 2005.

[56] Y. Qian and F. Liu, "Neural Network Predistortion Technique for Nonlinear Power Amplifiers with Memory," First Inter. Conf. on Communications and Networking, ChinaCom. 06, pp.1-5, 25-27 Oct. 2006.

[57] A. Saleh, "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers," IEEE Trans. on Comm., vol.29, Iss.11 pp.1715-1720, Nov. 1981.

[58] N. S. *Sengör*, "Lecture Notes on Articficiel Neural Networks for RF and Microwave Design," http ://www.ct.tkk.fi/courses/annrf/annrf.pdf, Spring 2001.

[59] N. Srirattana, "High-Efficiency Linear RF Power Amplifiers Development," Ph.D. Thesis, Georgia Institute of Technology, Apr. 2005.

[60] S. P. Stapleton and J. K. Cavers, "A New Technique for Adaptation of Linearizing Predistorters," 41^{st} IEEE Vehicular Technology Conf., pp.753-758, 1991.

[61] M. Stevenson and B. Widrow, "Sensititivity of feedforward Neural Networks to Weight errors," IEEE Trans. on Neural Networks, vol.1, no.1, pp.71-80, Mar. 1990.

[62] S. A. Maas, "Nonlinear Microwave and RF Circuits," second Edition, Artech House Publisher, 2003.

[63] Y. Takahashi, R. Ishikawa, and K. Honjo, "Precise Modeling of Thermal Memory Effect for Power Amplifier Using Multi-Stage Thermal RC-Ladder Network," Proc. 2006 Asia Pacific Microwave Conf, vol.1, pp.287-290, Dec. 2006.

[64] S. Tertois, "Réduction des Effets des Non-Linéarités dans une Modulation Multiporteuse à l'aide de Réseaux de Neurones," Ph.D. Thesis, Université de Rennes.1, Déc. 2003.

[65] T. Wang, "Compensation of Nonlinear Effects With Memory in Digital Transmitters using a Frequency Domain Identifiacation Method," Master Thesis, Dalhousie University, 2003.

[66] K. D. Vijaya, Xi. Changgeng, and Q. J. Zhang, "A Neural Network Approach to the Modeling of Heterojunction Bipolar Transistors from s-parameter Data," 28^{th} European Microwave Conf. and Exhibition, vol.1, pp.306-310, 1998.

[67] A. Walker, M. Steer and K. G. Gard, "A Vector Intermodulation Analyzer Applied to Behavioral Modeling of Nonlinear Amplifiers with Memory", IEEE Trans. on MTTs, vol.54, Iss.5, pp.1991-1999, May 2006.

[68] B. E. Watkins and R. North, "Predistortion of Nonlinear Amplifiers using Neural Networks," Military Commun. Conf., MILCOM'96, vol.1, pp.316-320, 1996.

[69] A. S. Wright and W. G. Durtler, "Experimental Performance of an Adaptive Digital Linearized Power Amplifier," IEEE MTT-S Digest, vol.2, pp.1105-1108, 1-5 Jun. 1992.

[70] www.xilinx.com

[71] R. Yates, "Fixed-Point Arithmetic : An Introduction," www.digitalsignallabs.com, Apr 17, 2007.

[72] Q. J. Zhang and K. C Gupta, "Neural networks for RF and Microwave Design," Norwood, MA, Artech House, 2000.

[73] D. Zhou and V. DeBrunner, "A Novel Adaptive Nonlinear Predistorter Based on the Direct Learning Algorithm," IEEE Inter. Conf. on Comm., vol.4, pp.2362-2367, 2004.

[74] A. Zhu and T. J. Brazil, "Optimal Digital Volterra Predistorter for Broadband RF Power Amplifier Linearization," 31^{st} Euro. Microwave Conf. EuMC 2001, vol.1, pp.1-4, 2001.

[75] A. Zhu, J. C. Pedro and T. J. Brazil, "Dynamic Deviation Reduction-Based Volterra Behavioral Modeling of RF Power Amplifiers," IEEE Trans. on MTTs, vol.54, Iss.12, part.2, pp.4323-4332, Dec. 2006.