

UNIVERSITÉ DU QUÉBEC À RIMOUSKI

IMPLANTATION DES RÉSEAUX DE NEURONES SUR FPGA POUR LA
MODÉLISATION ET LA LINÉARISATION PAR PRÉDISTORSION NUMÉRIQUE
DES AMPLIFICATEURS DE PUISSANCE

Mémoire présenté
dans le cadre du programme de maîtrise en ingénierie
en vue de l'obtention du grade de maître ès sciences appliquées (M. Sc. A.)

PAR
© ROGER SANDRIN NTOUNÉ NTOUNÉ

Janvier 2013

UNIVERSITÉ DU QUÉBEC À RIMOUSKI
Service de la bibliothèque

Avertissement

La diffusion de ce mémoire ou de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire « Autorisation de reproduire et de diffuser un rapport, un mémoire ou une thèse ». En signant ce formulaire, l'auteur concède à l'Université du Québec à Rimouski une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de son travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, l'auteur autorise l'Université du Québec à Rimouski à reproduire, diffuser, prêter, distribuer ou vendre des copies de son travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de la part de l'auteur à ses droits moraux ni à ses droits de propriété intellectuelle. Sauf entente contraire, l'auteur conserve la liberté de diffuser et de commercialiser ou non ce travail dont il possède un exemplaire.

Composition du jury :

Jean-François Méthot, président du jury, Université du Québec à Rimouski

Mohammed Bahoura, directeur de recherche, Université du Québec à Rimouski

Chan-Wang Park, codirecteur de recherche, Université du Québec à Rimouski

Hassan Ezzaidi, examinateur externe, Université du Québec à Chicoutimi

Dépôt initial le 14 Janvier 2013

Dépôt final le 09 Mai 2013

REMERCIEMENTS

Tout d'abord, j'aimerais remercier M. Mohammed Bahoura, mon directeur de recherche, pour son encadrement de qualité et ses conseils judicieux. Merci pour tout le temps qui m'a été consacré jusqu'à l'échéance de mon mémoire de maîtrise. Je souhaite aussi remercier M. Chan Wang Park, mon co-directeur de recherche, pour sa grande disponibilité et ses conseils avisés. Tous les deux, vous m'avez permis de publier des communications et d'assister à des conférences mondiales reconnues dans notre domaine, vous m'avez ouvert la porte en recherche en me proposant un sujet prometteur pour l'avenir dans le développement des solutions relatives aux problèmes en télécommunications. Grâce à vous, je suis mieux outillé pour faire face aux défis incessants du marché du travail. Je voudrais également remercier le président du jury M. Jean-François Méthot et le membre du jury externe M. Hassan Ezzaidi pour le précieux temps que vous avez accepté de consacrer à la correction de mon mémoire. Je vous suis très reconnaissant.

Par la suite, je souhaite remercier ma mère, Hélène Gounes Biouele, mon père, Roger Ntouné, mes sœurs, Dany Flore Mezogue Ntouné, Marthe Bella, Odile Yolande Ambono et Audrey Laéticia Nanga Ntouné, et frères, Raoul Noël Abada Ntouné et Arthur Rostand Gounes Ntouné, mon tonton, Georges Abada, et toute ma famille pour leur apport financier, leurs encouragements incessants et leur amour indubitable. Vous avez su me redonner confiance dans mes périodes de doute.

Je remercie aussi Manon Turmel pour son soutien inconditionnel, pour le tendre amour qu'elle m'a apportée et les moments de joie inoubliables partagés ensemble. Ta famille et toi m'avez offert un accueil chaleureux. Merci à Cyriac Oloum et sa famille pour m'avoir grand ouvert les portes de leur demeure et les conseils prodigieux qu'ils m'ont offerts à mon arrivée. Je remercie mes amis et mes confrères de travail. Nos discussions enrichissantes m'ont motivé à redoubler d'ardeur au travail. Je souhaite ensuite remercier l'UQAR et le département de mathématiques, d'informatique et de génie pour m'avoir fourni un environnement de travail adapté. À tous, merci de m'avoir permis de réaliser mon rêve d'obtenir une maîtrise dans une université canadienne.

RÉSUMÉ

Les non-linéarités et les effets mémoire des amplificateurs de puissance (PA) des stations de base en RF créent des distorsions des signaux dans la bande passante des systèmes de communication sans fil moderne à large bande. Parmi les méthodes de linéarisation développées ces deux dernières décennies, la prédistorsion numérique (DPD) prend de plus en plus de place parce qu'elle offre plus de stabilité et de flexibilité. La popularité de la technique DPD résulte de la puissance de calcul croissante des systèmes de traitement avec puces ASIC, DSP et FPGA. Les puces FPGA remplacent de plus en plus les puces DSP car ils sont hautement parallèles et flexibles en programmation. Les défis de l'implantation matérielle de n'importe quelle technique de DPD résident dans l'atteinte du traitement temps-réel en préservant les performances obtenues avec des modèles logiciels tels que Matlab/Simulink. Xilinx System Generator (XSG) est un outil de haut niveau qui a été utilisé pour programmer les puces FPGA fabriquées par Xilinx à partir de l'environnement Matlab/Simulink. Lors de la co-simulation matérielle le code est exécuté sur puce FPGA mais les données peuvent être échangés avec le logiciel Matlab. Les architectures proposées ont été utilisées dans la modélisation du comportement des PA avant d'être appliquées à leur linéarisation par prédistorsion numérique. Le réseau RVTDDN (*Real-Valued Time-Delay Neural Network*) basé sur le perceptron multicouche (MLP), auquel est ajouté une couche TDL (*time-delay layer*) est développé et comparé aux réseaux récurrents RVRNN (*Real-Valued Recurrent Neural Network*) et NARX (*nonlinear autoregressive with exogenous inputs*) auxquels sont ajoutés une couche TDL aux entrées et aux sorties. L'optimisation jusqu'à 156.128 MHz de la fréquence maximale d'opération (MOF) de ces architectures par pipeline a permis leur implantation, par l'interface *JTAG Hardware Co-simulation*, sur puce FPGA. Par ailleurs, un signal de test à bande de base modulé 16-QAM, de largeur de bande 1.35 MHz, est utilisé pour la validation des résultats. Le réseau NARX-8-pse corrige mieux les paramètres EVM que toutes les autres architectures de prédistorsion numérique, car il a 0.505 % de paramètre EVM et a une meilleure réduction des paramètres ACPR gauche (18.440 dB (decibels)) et droit (18.223 dB) par rapport au modèle de PA de Wiener.

Mots clés : non-linéarités, effets mémoire, PA, modélisation, DPD, FPGA.

ABSTRACT

Nonlinearities and memory effects of power amplifiers (PA) in RF base stations distort signals in the bandwidth of wireless broadband modern communication systems. Among the linearization methods developed over the past two decades, digital predistortion (DPD) is becoming more and more important because it provides more stability and flexibility. Digital predistortion's popularity results from the computing power increases of systems with ASIC, DSP and FPGA chips. DSP chips are progressively replaced by FPGA chip because they are parallel and highly flexible in programming. The challenges of hardware implantation of any DPD technique reside in achieving the real-time processing while preserving the performance obtained with software models such as Matlab/Simulink software. Xilinx System Generator (XSG) is a high-level tool that uses Matlab/Simulink environment to program Xilinx FPGA chip. During co-simulating in XSG tool, code is executed on FPGA chip but data can be exchanged with Matlab software. The proposed architectures are used in the PA behavioral modeling before being applied to linearize them by digital predistortion. The real-valued time-delay neural network (RVTDNN) based on multi-layer perceptron (MLP), with time delay layers (TDL) is developed and compared with real-valued recurrent neural network (RVRNN) and nonlinear autoregressive with exogenous inputs (NARX) on which are added TDL layer at inputs and outputs. Optimizing maximum operating frequency (MOF) up to 156.128 MHz in pipelining architecture allowed their FPGA implantation by JTAG Hardware Co-simulation. In addition, a modulated 16-QAM baseband test signal, with 1.35 MHz bandwidth, is used to validate the results. NARX-8-pse network corrects is other digital predistortion architectures in EVM parameter because it has 0.505 % in EVM parameter and it is better ACPR parameter on left (18.440 dB (decibels)) and on right (18.223 dB) reductions relative to Wiener PA model.

Keywords : nonlinearities, memory effects, PA, modeling, DPD, FPGA.

TABLE DES MATIÈRES

REMERCIEMENTS	vii
RÉSUMÉ	ix
ABSTRACT	xi
TABLE DES MATIÈRES	xiii
LISTE DES TABLEAUX	xvii
LISTE DES FIGURES	xix
LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES	xxvii
CHAPITRE 1 :INTRODUCTION GÉNÉRALE	1
1.1 État de l'art	1
1.2 Modélisation de la non-linéarité et des effets mémoire d'un PA	5
1.2.1 Modélisation de la non-linéarité	5
1.2.2 Modélisation de la non-linéarité et des effets mémoire	7
1.3 Caractérisation des distorsions	7
1.3.1 Point de compression à 1 dB	7
1.3.2 Mesure du paramètre ACPR avant et après amplification du signal	8
1.3.3 Mesure du paramètre EVM du signal à l'entrée et à la sortie d'un PA	9
1.3.4 Erreur quadratique moyenne (MSE)	11
1.4 Méthodes de linéarisation d'un PA	11
1.4.1 Linéarisation par contre-réaction	11
1.4.2 Linéarisation par post-compensation	12
1.4.3 Linéarisation par utilisation de composants non-linéaires	13
1.4.4 Linéarisation par prédistorsion numérique	14
1.5 Problématique	16
1.6 Objectifs	18
1.7 Hypothèses	18

1.8	Méthodologie	19
1.9	Contributions apportées	19
1.10	Organisation du mémoire	20

CHAPITRE 2 : MODÉLISATION DES AMPLIFICATEURS DE PUISSANCE 21

2.1	Principe de modélisation des amplificateurs de puissance	21
2.2	Architectures conventionnelles des réseaux RVDNN, RVRNN et NARX pour la modélisation des PAs	23
2.2.1	Architecture conventionnelle du réseau RVDNN	24
2.2.2	Architecture conventionnelle du réseau RVRNN	28
2.2.3	Architecture conventionnelle du réseau NARX	30
2.3	Architectures pseudo-conventionnelles des réseaux NARX, RVRNN et RVDNN pour la modélisation des PAs	32
2.3.1	Architecture pseudo-conventionnelle du réseau RVDNN	32
2.3.2	Architecture pseudo-conventionnelle du réseau RVRNN	36
2.3.3	Architecture pseudo-conventionnelle du réseau NARX	37
2.4	Implantation des architectures pseudo-conventionnelles des modèles RVDNN, RVRNN et NARX en utilisant les blocs de l'outil XSG	39
2.4.1	Implantation de l'architecture RVDNN-6-pse	39
2.4.2	Implantation de l'architecture RVRNN-6-pse	47
2.4.3	Implantation de l'architecture NARX-6-pse	49
2.5	Architectures pipelinées des réseaux RVDNN, RVRNN et NARX pour la modélisation des PAs	52
2.5.1	Architecture pipelinée du réseau RVDNN	53
2.5.2	Architecture pipelinée du réseau RVRNN	62
2.5.3	Architecture pipelinée du réseau NARX	65
2.6	Implantation des architectures pipelinées des modèles RVDNN, RVRNN et NARX en utilisant les blocs de l'outil XSG	66
2.6.1	Implantation de l'architecture RVDNN- N_1 -pip	66
2.6.2	Implantation de l'architecture RVRNN- N_1 -pip	72
2.6.3	Implantation de l'architecture NARX- N_1 -pip	74
2.7	Implantation des architectures de modélisation sur la puce FPGA	76

2.7.1	Implantation des architectures pseudo-conventionnelles	77
2.7.2	Implantation des architectures pipelinées	80
2.8	Expérimentation et analyse des résultats de modélisation par des architectures pseudo-conventionnelles et pipelinées	80
2.8.1	Expérimentation des architectures pseudo-conventionnelles et pipelinées	81
2.8.2	Analyse des résultats de modélisation par des architectures pseudo-conventionnelles et pipelinées	81
2.8.3	Performances des architectures de modélisation RVTDNN-6-pse, RV-RNN-6-pse et NARX-6-pse	82
2.8.4	Performances des architectures de modélisation RVTDNN-8-pse, RV-RNN-8-pse et NARX-8-pse	90
2.8.5	Performances des architectures de modélisation RVTDNN-6-pip, RV-RNN-6-pip et NARX-6-pip	96
2.8.6	Performances des architectures de modélisation RVTDNN-8-pip, RV-RNN-8-pip et NARX-8-pip	104

CHAPITRE 3 : LINÉARISATION DES AMPLIFICATEURS DE PUISSANCE PAR PRÉDISTORSION NUMÉRIQUE 115

3.1	Principe de la prédistorion numérique	115
3.2	Prédistorion par architectures conventionnelles, pseudo-conventionnelles et pipelinées des réseaux de neurones	117
3.3	Implantation des architectures de prédistorion en utilisant les blocs de l'outil XSG	118
3.3.1	Implantation des architectures pseudo-conventionnelles	118
3.3.2	Implantation des architectures pipelinées	118
3.4	Implantation des architectures de prédistorion sur la puce FPGA	120
3.4.1	Implantation des architectures pseudo-conventionnelles	120
3.4.2	Implantation des architectures pipelinées	121
3.5	Expérimentation et analyse des résultats de prédistorion par des architectures pseudo-conventionnelles et pipelinées	121
3.5.1	Expérimentation des architectures de prédistorion pseudo-conventionnelles et pipelinées	123

3.5.2	Analyse des résultats de prédistorsion par des architectures pseudo-conventionnelles et pipelinées	124
3.5.3	Performances des architectures de prédistorsion RVTDNN-6-pse, RV-RNN-6-pse et NARX-6-pse	124
3.5.4	Performances des architectures de prédistorsion RVTDNN-8-pse, RV-RNN-8-pse et NARX-8-pse	131
3.5.5	Performances des architectures de prédistorsion RVTDNN-6-pip, RV-RNN-6-pip et NARX-6-pip	138
3.5.6	Performances des architectures de prédistorsion RVTDNN-8-pip, RV-RNN-8-pip et NARX-8-pip	145
CHAPITRE 4 : CONCLUSION GÉNÉRALE ET PERSPECTIVES		157
4.1	Conclusion générale	157
4.2	Perspectives	163

LISTE DES TABLEAUX

Tableau 1 : Ressources utilisées et fréquence maximale d'opération des architectures de modélisation pseudo-conventionnelles.	79
Tableau 2 : Ressources utilisées et fréquence maximale d'opération des architectures de modélisation avec pipeline.	79
Tableau 3 : Résultats de simulation des architectures de modélisation pseudoconventionnelles et avec pipeline.	111
Tableau 4 : Ressources utilisées et fréquence maximale d'opération des architectures de prédistorsion pseudo-conventionnelles.	122
Tableau 5 : Ressources utilisées et fréquence maximale d'opération des architectures de prédistorsion avec pipeline.	122
Tableau 6 : Résultats de simulation des architectures de prédistorsion pseudoconventionnelles et avec pipeline.	153

LISTE DES FIGURES

Figure 1 :	Système représentant un amplificateur de puissance (PA) non-linéaire.	6
Figure 2 :	Modèle de PA de Wiener	7
Figure 3 :	Point de compression à 1 dB.	8
Figure 4 :	Illustration du paramètre ACPR avant et après amplification du signal.	9
Figure 5 :	Illustration du paramètre EVM du signal à l'entrée et à la sortie d'un PA.	10
Figure 6 :	Linéarisation par contre-réaction (<i>feedback</i>).	11
Figure 7 :	Linéarisation par post-compensation (<i>feedforward</i>).	13
Figure 8 :	Linéarisation par utilisation de composants non-linéaires (LINC).	14
Figure 9 :	Linéarisation par prédistorsion numérique (DPD).	15
Figure 10 :	Architecture du modèle polynomial (PM).	16
Figure 11 :	Principe de modélisation des PA pour un signal en bande de base.	22
Figure 12 :	Architecture d'un perceptron multicouche (MLP) à une couche cachée.	22
Figure 13 :	Architecture conventionnelle du réseau RVTDNN.	24
Figure 14 :	Structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture conventionnelle du réseau RVTDNN.	26
Figure 15 :	Structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture conventionnelle du réseau RVTDNN.	28
Figure 16 :	Architecture conventionnelle du réseau RVRNN.	29
Figure 17 :	Architecture conventionnelle du réseau NARX.	31
Figure 18 :	Architecture pseudo-conventionnelle du réseau RVTDNN.	32
Figure 19 :	Structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture pseudo-conventionnelle du réseau RVTDNN.	35
Figure 20 :	Structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture pseudo-conventionnelle du réseau RVTDNN.	36
Figure 21 :	Architecture pseudo-conventionnelle du réseau RVRNN.	37
Figure 22 :	Architecture pseudo-conventionnelle du réseau NARX.	38
Figure 23 :	Système de test avec modulation 16-QAM des architectures de modélisation pseudo-conventionnelles à bande de base.	40
Figure 24 :	Architecture de modélisation RVTDNN-6-pse.	42

Figure 25 : Blocs <i>Gateway In</i> et <i>Gateway Out</i> de l'outil XSG.	43
Figure 26 : Implantation d'un neurone de la couche cachée de l'architecture RVTDNN-6-pse en utilisant les blocs de l'outil XSG.	44
Figure 27 : Implantation de la variation des poids des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.	44
Figure 28 : Implantation de la mise à jour des poids et des biais des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.	45
Figure 29 : Implantation de la somme dans chaque neurone de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.	45
Figure 30 : Implantation de la tangente hyperbolique des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.	46
Figure 31 : Implantation d'un neurone de la couche de sortie de l'architecture RVTDNN-6-pse avec des blocs de l'outil XSG.	47
Figure 32 : Implantation de l'architecture RVRNN-6-pse en utilisant les blocs de l'outil XSG.	48
Figure 33 : Implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pse avec des blocs de l'outil XSG.	49
Figure 34 : Implantation de l'architecture NARX-6-pse avec des blocs de l'outil XSG.	50
Figure 35 : Implantation d'un neurone de la couche cachée de l'architecture NARX-6-pse avec des blocs de l'outil XSG.	51
Figure 36 : Architecture pipelinée du réseau RVTDNN-6-pip.	54
Figure 37 : Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDNN-6-pip.	56
Figure 38 : Structure des neurones de la couche cachée de l'architecture pipelinée RVTDNN-6-pip.	57
Figure 39 : Architecture pipelinée du réseau RVTDNN-8-pip.	58
Figure 40 : Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDNN-8-pip.	60
Figure 41 : Structure des neurones de la couche cachée de l'architecture pipelinée RVTDNN-8-pip.	62
Figure 42 : Architecture pipelinée du réseau RVRNN.	63

Figure 43 : Structure des neurones de la couche cachée de l'architecture pipelinée RVRNN-8-pip.	64
Figure 44 : Structure des neurones de la couche de sortie de l'architecture pipelinée RVRNN-6-pip.	64
Figure 45 : Architecture pipelinée du réseau NARX.	65
Figure 46 : Système de test avec modulation 16-QAM à bande de base des architectures de modélisation par le sous-système <i>Neural Networks</i> basé sur le réseau RVTDDN-8-pip, le réseau RVRNN- N_1 -pip et le réseau NARX- N_1 -pip et le bloc correspondant <i>JTAG Hardware Co-simulation</i>	67
Figure 47 : Système de test avec modulation 16-QAM à bande de base des architectures de modélisation par le sous-système <i>Neural Networks</i> basé sur le réseau RVTDDN-8-pip et le bloc correspondant <i>JTAG Hardware Co-simulation</i>	67
Figure 48 : Implantation de la somme dans chaque neurone de la couche cachée de l'architecture RVTDDN-6-pip.	68
Figure 49 : Implantation d'un neurone de la couche cachée de l'architecture RVTDDN-6-pip en utilisant les blocs de l'outil XSG.	69
Figure 50 : Implantation d'un neurone de la couche cachée de l'architecture RVTDDN-8-pip en utilisant les blocs de l'outil XSG.	69
Figure 51 : Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pip en utilisant les blocs de l'outil XSG.	70
Figure 52 : Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-8-pip en utilisant les blocs de l'outil XSG.	71
Figure 53 : Implantation de la somme dans chaque neurone de la couche de sortie de l'architecture RVTDDN-6-pip.	72
Figure 54 : Implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pip en utilisant les blocs de l'outil XSG.	73
Figure 55 : Implantation d'un neurone de la couche de sortie de l'architecture RVRNN-6-pip en utilisant les blocs de l'outil XSG.	74
Figure 56 : Implantation d'un neurone de la couche cachée de l'architecture NARX-6-pip en utilisant les blocs de l'outil XSG.	75
Figure 57 : Implantation de la somme dans chaque neurone de la couche cachée de l'architecture NARX-6-pip.	76

Figure 58 : Système de co-simulation matérielle contenant un ordinateur, la puce FPGA Virtex-6 de la carte ML605 et le câble USB.	78
Figure 59 : Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.	83
Figure 60 : Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.	84
Figure 61 : Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.	85
Figure 62 : Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.	86
Figure 63 : Spectres du signal d'entrée et des signaux de sorties du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.	87
Figure 64 : Convergence des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.	88
Figure 65 : Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.	90
Figure 66 : Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.	91
Figure 67 : Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.	92
Figure 68 : Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.	93
Figure 69 : Spectres du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.	94
Figure 70 : Convergence des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.	95

Figure 71 : Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.	97
Figure 72 : Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.	98
Figure 73 : Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.	99
Figure 74 : Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.	100
Figure 75 : Spectre du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.	101
Figure 76 : Convergence des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.	102
Figure 77 : Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.	104
Figure 78 : Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.	105
Figure 79 : Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.	106
Figure 80 : Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.	107
Figure 81 : Spectre du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.	108
Figure 82 : Convergence des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.	109

Figure 83 : Diagramme de l'apprentissage direct pour la prédistorsion numérique (DPD) en bande de base.	116
Figure 84 : Diagramme de l'apprentissage indirect pour la prédistorsion numérique (DPD) en bande de base.	116
Figure 85 : Système de test avec modulation 16-QAM des architectures pseudo-conventionnelles de prédistorsion à bande de base	118
Figure 86 : Système de test avec modulation 16-QAM de prédistorsion <i>Neural Networks</i> avec pipeline et du bloc <i>JTAG Hardware Co-simulation</i> exépté le réseau RVTDNN-6-pip	119
Figure 87 : Système de test avec modulation 16-QAM à bande de base des architectures de prédistorsion RVTDNN-6-pip et du bloc <i>JTAG Hardware Co-simulation</i>	119
Figure 88 : Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence de Wiener	125
Figure 89 : Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.	126
Figure 90 : Conversion AM/PM du modèle de référence de Wiener seul et des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.	127
Figure 91 : Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.	128
Figure 92 : Spectre de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse en série avec le modèle de référence.	129
Figure 93 : Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse en série avec le modèle de référence.	130
Figure 94 : Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.	132

Figure 95 : Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.	133
Figure 96 : Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.	134
Figure 97 : Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDDN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.	135
Figure 98 : Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion du réseau NARX-8-pse, du réseau RVRNN-8-pse et du réseau RVTDDN-8-pse en série avec le modèle de référence.	136
Figure 99 : Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse en série avec le modèle de référence.	137
Figure 100 : Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.	139
Figure 101 : Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.	140
Figure 102 : Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.	141
Figure 103 : Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.	142
Figure 104 : Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip en série avec le modèle de référence.	143

Figure 105 : Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVDNN-6-pip en série avec le modèle de référence.	144
Figure 106 : Diagramme de constellation des architectures de prédistorsion RVDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener et de l'entrée.	146
Figure 107 : Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener.	147
Figure 108 : Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener.	148
Figure 109 : Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener et composant cartésien Q du modèle de référence de Wiener.	149
Figure 110 : Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVDNN-8-pip en série avec le modèle de référence.	150
Figure 111 : Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVDNN-8-pip en série avec le modèle de référence.	151

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

ACLR	<i>Adjacent Channel Leakage Ratio</i> Taux de Fuite des Canaux Adjacents
ACPR	<i>Adjacent Channel Power Ratio</i> Taux de Puissance des Canaux Adjacents
ADC	<i>Analog to Digital Converter</i> Convertisseur Analogique Numérique
AM	<i>Amplitude Modulation</i> Modulation d'Amplitude
AM/AM	<i>Amplitude-Amplitude Conversion</i> Conversion Amplitude-Amplitude
AM/PM	<i>Amplitude-Phase Conversion</i> Conversion Amplitude-Phase
ASIC	<i>Application-Specific Integrated Circuit</i> Circuit Intégré Propre à une Application
CALLUM	<i>Combined Analog Locked-Loop Universal Modulator</i> Combiné Analogique à Boucle à Verrouillage Modulateur Universel
CDMA	<i>Code Division Multiple Access</i> Accès Multiple par Répartition en Code
CDMA2000	<i>Code Division Multiple Access 2000</i> Accès Multiple par Répartition en Code 2000
DAC	<i>Digital to Analog Converter</i> Convertisseur Numérique Analogique
DPD	<i>Digital PreDistortion</i> Prédistortion Numérique
DSP	<i>Digital Signal Processor</i> Processeur de Signaux Numériques
EVM	<i>Error Vector Magnitude</i> Amplitude du Vecteur d'Erreur
FIR	<i>Finite Impulse Response</i> Réponse Impulsionnelle Finie

FPGA	<i>Field Programmable Gate Array</i> Réseau de Portes Programmables
HF	<i>High Frequency</i> Haute Fréquence
IOB	<i>Input/Output Block</i> Bloc d'Entrée/Sortie
JTAG	<i>Joint Test Action Group</i> Groupe d'Action Commun de Test
LINC	<i>Linear Amplification using Nonlinear Components</i> Amplification Linéaire Utilisant des Composants Non-linéaires
LTE	<i>Long Term Evolution</i> Évolution à Long Terme
LUT	<i>Look-Up Table</i> Table de Correspondance
MF	<i>Medium Frequency</i> Moyenne Fréquence
MLP	<i>Multi-Layer Perceptron</i> Perceptron Multicouches
MOF	<i>Maximum Operating Frequency</i> Fréquence Maximale d'Opération
MSDPD	<i>Mixed Signal Digital Pre-Distortion</i> Prédistorsion numérique du signal Modulé
MSE	<i>Mean Square Error</i> Erreur Quadratique Moyenne
M-QAM	<i>M-Array Quadrature Amplitude Modulation</i> Modulation d'Amplitude en Quadrature d'ordre M
NARX	<i>Nonlinear Autoregressive with Exogenous Inputs</i> Non-linéaire Autorégressif avec Entrées Exogènes
NRMSE	<i>Normalized Root Mean Square Error</i> Racine de l'Erreur Quadratique Moyenne Normalisée
OFDM	<i>Orthogonally Frequency Division Multiplexing</i> Multiplexage par Répartition Orthogonale de Fréquence

PA	<i>Power Amplifier</i> Amplificateur de Puissance
PM	<i>Polynomial model</i> Modèle Polynomial
QAM	<i>Quadrature Amplitude Modulation</i> Modulation d'Amplitude en Quadrature
QPSK	<i>Quadrature Phase-Shift Keying</i> Modulation par Déplacement de Phase en Quadrature
RAM	<i>Random Access Memory</i> Mémoire à Accès Direct
RF	<i>Radio frequency</i> Fréquence Radio
RNN	<i>Recurrent Neural Network</i> Réseaux de Neurones Récurrents
ROM	<i>Read Only Memory</i> Mémoire à Lecture Seule
RVFTDNN	<i>Real-Valued Focused Time-Delay Neural Network</i> Réseaux de Neurones concentrés à Valeurs Réelles avec des Retards
RVRNN	<i>Real-Valued Recurrent Neural Network</i> Réseaux de Neurones Récurrent à Valeurs Réelles
RVTDNN	<i>Real-Valued Time-Delay Neural Network</i> Réseaux de Neurones à Valeurs Réelles avec des Retards
TDL	<i>Time-Delay Layer</i> Couche de Retards Temporels
TDNN	<i>Time-Delay Neural Network</i> Réseaux de Neurones avec des Retards
TNTB	<i>Twin Nonlinear Two-Box</i> Boite Double non Linéaire Encadrée
TWT	<i>Travelling-Wave Tube</i> Tube à Ondes Progressives
VHDL	<i>Very High Speed Integrated Circuit Hardware Description Language</i> Langage de Description du Matériel pour Circuits Intégrés Très Rapides

xxx

VHF	<i>Very High Frequency</i> Très Haute Fréquence
W-CDMA	<i>Wide Band Code Division Multiple Access</i> Accès Multiple par Répartition en Code à Large Bande
XSG	<i>Xilinx System Generator</i> Système Générateur de Xilinx
3G	<i>Third Generation</i> Troisième Génération
4G	<i>Fourth Generation</i> Quatrième génération
16-QAM	<i>Quadrature Amplitude Modulation with 16 symbols</i> Modulation d'Amplitude en Quadrature avec 16 symboles

CHAPITRE 1

INTRODUCTION GÉNÉRALE

1.1 État de l'art

Les systèmes de communication sans fil sont omniprésents de nos jours. Le besoin des opérateurs de téléphonie mobile d'offrir plus de services et de les faire passer dans la même bande passante a permis le passage à des largeurs de bande plus grandes et l'utilisation de signaux à enveloppe non-constante. Le spectre de radio fréquence (RF) couvre, entre autres, les bandes de moyennes fréquences (MF) qui vont de 300 kHz à 3 MHz, de hautes fréquences (HF) qui vont de 3 MHz à 30 MHz et de très hautes fréquences (VHF) qui vont de 30 MHz à 300 MHz (Doufana, 2009; Golio et Golio, 2008). Afin d'augmenter l'efficacité spectrale (Barbieri et al., 2010), l'usage des techniques de modulation complexes, telles que la technique OFDM (*Orthogonally Frequency Division Multiplexing*) (Barbieri et al., 2010; Kolinko et Larson, 2009; Nader et al., 2011), se fait de plus en plus. Les émetteurs/récepteurs en radiocommunications numériques sont composés d'une partie analogique ou partie RF et d'une partie numérique ou à bande de base dont le spectre est centré sur la fréquence nulle (Baudoin et al., 2007). L'amplificateur de puissance (PA) des émetteurs de stations de base, qui réalise l'amplification du signal avant son envoi dans le canal de transmission d'une chaîne de communication sans fil, est un composant qui peut présenter de faibles (classe AB (Azam et al., 2008)) ou fortes (Doherty (Kwan et al., 2010; Kwon et al., 2009; Ye et al., 2008a)) non-linéarités dans sa zone de saturation. Les effets mémoire (Liu et al., 2008b; Tornblad et al., 2007; Ye et al., 2008b) peuvent être classifiés en deux catégories : les effets mémoire électrothermiques (Boumaiza et al., 2003) qui sont principalement causés par les capacités thermiques et les résistances thermiques (Liu et al., 2008b), et les effets mémoire électriques (Alghanim et al., 2007) qui sont attribués à la non-constance de la réponse fréquentielle du PA autour de la fréquence porteuse (Liu et al., 2008b). Les effets mémoire et les non-linéarités du PA créent des distorsions dans le signal transmis qui correspondent aux remontées spectrales dans les canaux adjacents ACPR (*Adjacent Channel Power Ratio*) (Bennadji, 2006; Bensmida, 2005; Dardenne, 2005; Marsalek, 2003) ainsi qu'aux déformations de la constellation EVM (*Error Vector Magnitude*) (Bennadji, 2006; Bensmida, 2005; Dardenne, 2005; Forestier et al., 2003; Halder et

Chatterjee, 2005; Marsalek, 2003) dans la bande passante des systèmes de communication sans fil modernes à large bande, tel que la technique W-CDMA (*Wide Band Code Division Multiple Access*) 3G (*Third Generation*) (Abusaid, 2010; Mellein, 2007; Subramaniam, 2005) et la technologie LTE-Advanced 4G (*Fourth Generation*). Pour ses besoins en efficacité spectrale et en haut débit de données, la technologie LTE (*Long Term Evolution*) utilise comme procédé de modulation multi-porteuse la technique OFDM (Rahman, 2011). Les modulations en bande de base supportées par la technologie LTE sont la modulation QPSK (*Quadrature Phase-Shift Keying*) et la modulation 16-QAM (*Quadrature Amplitude Modulation with 16 symbols*) (Rahman, 2011).

La méthode la plus simple pour solutionner ce problème de non-linéarité est de reculer le niveau de puissance (*Back-off*) (Jang et al., 2010; Marsalek, 2003). Dans cette méthode, la linéarité est meilleure pour les amplificateurs de classe AB, mais l'efficacité est réduite (Marsalek, 2003). Afin d'accroître cette dernière et de minimiser les distorsions, de nombreuses techniques de linéarisation, telles que *feedback* (Moon et Kim, 2010; Shan et Sundstrom, 2002), *feedforward* (Braithwaite et Hunton, 2010; Jeong et al., 2006; Tabatabai et Al-Raweshidy, 2007), LINC (*Linear Amplification using Nonlinear Components*) (Daho et al., 2011; Dardenne, 2005; Helaoui et Ghannouchi, 2010; Poitou et al., 2004; Woo et al., 2003), CALLUM (*Combined Analog Locked-Loop Universal Modulator*) (Baudoin et al., 2007; Jennings et McGeehan, 1998) et DPD (*Digital PreDistortion*) (Doufana, 2009; Guan et Zhu, 2010, 2011a; Liu et al., 2011; Mkaem, 2010; Rawat et al., 2010) sont utilisées. Il existe plusieurs modèles de prédistorsion, parmi lesquels on trouve les séries de Volterra (Guan et Zhu, 2010, 2011a; Liszewski et al., 2011), le modèle polynomial (PM) (Djamai et al., 2008; Fehri et Boumaiza, 2011; Liu et al., 2008a) et plusieurs modèles de réseaux de neurones. En raison de leur haute exactitude, leur flexibilité et, par conséquent, leur capacité à bien modéliser avec succès les fonctions non-linéaires des PA en RF, les réseaux de neurones tels que RVTDDNN (*Real-Valued Time-Delay Neural Network*) (Boumaiza et Mkaem, 2009; Doufana et al., 2010; Liu et al., 2004, 2008b) et RVRNN (*Real-Valued Recurrent Neural Network*) (Rawat et al., 2010) sont utilisés pour la modélisation et la linéarisation par la technique DPD. Les réseaux de neurones NARX (*Nonlinear Autoregressive with Exogenous Inputs*) ont également été proposés dans les travaux de ce mémoire.

Les architectures de réseaux de neurones (RVTDDNN, RVRNN et NARX) développées dans ce mémoire sont basées sur un perceptron multicouches (MLP), auquel est ajouté une

couche de retards temporels (TDL) en entrée, pour que les entrées des réseaux de neurones ne soient pas seulement constituées de valeurs courantes de leurs entrées, mais aussi des valeurs précédentes (Golio et Golio, 2008) et éventuellement des valeurs précédentes de leurs sorties. Ces architectures RVTDDN et RVRNN ont été choisies, car elles ont été implantées avec succès sous Matlab. Cependant, l'architecture NARX a été adaptée, car elle tient compte également de l'historique des sorties.

Liu et al. (2004) ont initialement proposé le réseau RVTDDN pour la modélisation dynamique en bande de base des non-linéarités des amplificateurs de puissance des stations de base 3G en RF. Les poids synaptiques du réseau sont mis à jour par l'algorithme de rétro-propagation du gradient. Une première topologie proposée consiste à utiliser deux réseaux de neurones différents pour traiter séparément les entrées en phase et en quadrature. Du fait que les deux réseaux de neurones apprennent séparément, la première topologie proposée souffre de problèmes de convergence lors de l'apprentissage. Dans la deuxième topologie utilisant un seul réseau de neurones, les poids synaptiques et les fonctions d'activation sont à valeurs complexes, parce que le réseau a une entrée et une sortie à valeurs complexes. De ce fait, le réseau de neurones nécessite un algorithme d'apprentissage complexe. Ainsi, les deux topologies de réseaux de neurones précitées ont un long temps d'apprentissage. Le réseau RVTDDN ne souffre pas des mêmes problèmes que les deux topologies précitées, car il utilise un seul réseau à entrée/sortie réelles (en phase et en quadrature). La validation des résultats du modèle RVTDDN avec un signal CDMA2000-SR3 à une porteuse est également satisfaisante (Liu et al., 2004).

Hwangbo et al. (2006) proposent une approche de modélisation et de linéarisation par prédistorsion numérique (DPD) des signaux à bande de base des amplificateurs de grande puissance avec processus d'apprentissage indirect. Cette approche est basée sur les réseaux de neurones à délais temporels TDNN (Time-Delay Neural Network), similaire au réseau RVTDDN. Les retards ont été introduits dans l'algorithme afin d'éliminer les effets mémoire causés par l'amplificateur de grande puissance. Par rapport à l'architecture de prédistorsion avec un réseau de neurones sans retards, la méthode de prédistorsion envisagée a montré une amélioration d'environ 15 dB (decibels) pour le paramètre ACLR (*Adjacent Channel Leakage Ratio*). Environ 65 dBc (decibels relative to the carrier) pour le paramètre ACLR ont été atteints à 5 MHz de décalage (offset). Pour un signal CDMA2000, les améliorations ont été respectivement d'environ 20 dB et moins de 10 dB, avec et sans retards.

Guan et Zhu (2010) présentent une implantation matérielle peu complexe et à faible coût d'une prédistorsion numérique à base de séries de Volterra dans la puce FPGA Xilinx Virtex-4 XC4VSX35. Le résultat de l'implantation est obtenu par l'utilisation d'une LUT (*Look-Up Table*) qui a permis l'indexation du gain et du multiplexage temporel qui sert au partage des multiplieurs. Des résultats expérimentaux montrent que l'implantation de cette prédistorsion numérique utilise beaucoup moins de ressources matérielles, mais conserve toujours d'excellentes performances comparativement aux approches conventionnelles. Les paramètres ACPR dans les premiers et les deuxièmes canaux adjacents ont été réduits à moins de -60 dBc et le paramètre NRMSE (*Normalized Root Mean Square Error*) a été réduit de 1.8 %. Les puces FPGA ont beaucoup d'avantages en traitement numérique de signaux dont une grande vitesse, un fonctionnement en parallèle et une implantation flexible. Dans le but de traiter le signal en temps-réel, les puces FPGA sont devenus l'un des principaux choix pour l'implantation en bande de base des prédistorsions numériques des PA en RF.

Kwan et al. (2010) proposent une prédistorsion numérique en utilisant un module TNTB (*Twin Nonlinear Two-Box*) qui est composé d'un compartiment de table LUT sans mémoire en parallèle avec un compartiment polynomial avec mémoire. En outre, un système matériel-logiciel est utilisé pour la modélisation et la linéarisation des PA larges bandes avec effets mémoire. Le système conçu comprend la génération de signaux et une rétro-propagation pour les applications comprises dans la bande de fréquence 1 GHz à 2.5 GHz et soutient une largeur de bande instantanée de 60 MHz. La plateforme matérielle est couplée avec une interface logicielle qui implémente l'algorithme de traitement des données du module TNTB pour la synthèse de la fonction de prédistorsion numérique. La validation expérimentale est réalisée avec un amplificateur de puissance 3G et un signal W-CDMA à huit porteuses. Les résultats montrent 55 dBc pour le paramètre ACLR pour un signal d'entrée ayant 40 MHz de bande passante instantanée.

Doufana (2009) présente une architecture de linéarisation par prédistorsion numérique en bande de base des PA basée sur le réseau RVTDDN. Il a développé cette architecture en utilisant pour la première fois l'outil XSG (Xilinx System Generator). Il a proposé une prédistorsion adaptative des données en ligne avec une architecture d'apprentissage direct, pour assurer une adaptation continue et qui n'interrompt pas le processus de transmission. Une amélioration de 30 dB des paramètres ACPR est réalisée avec une convergence rapide et une bonne stabilité. Le réseau de neurones utilisé est un réseau RVTDDN à 15 neurones

au niveau de la couche cachée et 2 neurones en sortie. Le signal de test est un signal modulé 16-QAM généré avec la boîte à outils de communication de Matlab. Les performances ont été obtenues sans se soucier du nombre de bits des données et des ressources requises. Doufana (2009) a obtenu de bons résultats de simulation et a démontré l'efficacité de la méthode proposée.

Rawat et al. (2010) ont fait une étude comparative d'un réseau RVFTDNN (*Real-Valued Focused Time-Delay Neural Network*) et d'un réseau RVRNN et ont conclu que le réseau RVFTDNN est mieux adapté pour la modélisation inverse en bande de base des PA des émetteurs sans fil en RF. Ils ont également testé sur les réseaux de neurones développés les algorithmes d'apprentissage les plus efficaces lors de la modélisation inverse des PAs. Le réseau RVFTDNN proposé a été validé pour linéariser un amplificateur de classe AB qui est légèrement non-linéaire et un PA Doherty fortement non-linéaire avec un signal large bande W-CDMA pour les applications utilisant des signaux mono et multi-porteuse. Une correction maximale pour le paramètre ACLR de 20 dB est obtenue, où environ 5 dB de correction est observé en raison de la suppression des effets mémoire des signaux larges bandes W-CDMA multi-porteuses. Aussi, ils présentent une topologie utilisant deux réseaux de neurones, avec comme principal inconvénient la convergence asynchrone des deux réseaux, où ceux-ci ne convergent pas vers un modèle optimal en même temps, ce qui conduit à un sous ou sur apprentissage des réseaux de neurones. Ils présentent également une autre topologie de réseaux de neurones qui s'est avérée efficace pour la modélisation des PA très peu linéaires, mais elle donne des résultats décevants lorsque le PA montre de forts effets mémoire. Le réseau RVFTDNN utilise 2 couches cachées, une de 7 et une autre de 15 neurones en plus de la couche de sortie à 2 neurones. Leur réseau RVRNN a aussi deux couches cachées, une de 10 et une autre de 17 neurones, avec également 2 neurones en couche de sortie. L'implantation de la prédistorsion s'est faite avec le logiciel Matlab.

1.2 Modélisation de la non-linéarité et des effets mémoire d'un PA

1.2.1 Modélisation de la non-linéarité

La figure 1 illustre un exemple de signal traversant un système non-linéaire, où le gain complexe $G(A)$ est défini par son module g et sa phase ϕ (Baudoin et al., 2007).



Figure 1: Système représentant un amplificateur de puissance (PA) non-linéaire.

$$G(A) = g(A)e^{j\phi(A)} \quad (1.1)$$

Plusieurs modèles d'amplificateur de puissance (PA) ont été proposés dans la littérature. Le modèle de Saleh (Saleh, 1981) proposé pour la modélisation d'un PA de type TWT (*Travelling-Wave Tube*) est probablement le plus populaire. Ce modèle peut être défini par les caractéristiques AM/AM (*Amplitude-Amplitude Conversion*) et AM/PM (*Amplitude-Phase Conversion*) suivantes (Saleh, 1981) :

$$A(r(t)) = \frac{\alpha_A r(t)}{1 + \beta_A r^2(t)} \quad (1.2)$$

$$\Phi(r(t)) = \frac{\alpha_\Phi r(t)}{1 + \beta_\Phi r^2(t)} \quad (1.3)$$

où $r(t)$ représente l'enveloppe du signal modulé appliqué à l'entrée et t est le temps. Les valeurs typiques des paramètres sont : $\alpha_A = 2,1587$, $\beta_A = 1,1517$, $\alpha_\Phi = 4,0033$ et $\beta_\Phi = 9,1040$ (Saleh, 1981). Les valeurs de ces paramètres proviennent des résultats optimaux obtenus avec les données expérimentales du PA TWT (Saleh, 1981). Le signal $x(t)$ à l'entrée du PA est :

$$x(t) = r(t) \cos(\omega_c t + \phi_0(t)) \quad (1.4)$$

Le signal $y(t)$ à la sortie du PA est :

$$y(t) = A(r(t)) \cos(\omega_c t + \phi_0(t) + \Phi(r(t))) \quad (1.5)$$

où $\omega_c = 2\pi f_c$, f_c étant la fréquence porteuse. $\phi_0(t)$ est la phase du signal d'entrée. L'enveloppe d'amplitude instantanée $A(r(t))$ est une fonction impaire de $r(t)$, avec un premier terme linéaire représentant la conversion AM/AM et $\Phi(r(t))$ est une fonction paire de $r(t)$, avec un premier terme quadratique représentant la conversion AM/PM (Saleh, 1981).

1.2.2 Modélisation de la non-linéarité et des effets mémoire

Notre modèle de référence est basé sur le modèle de Wiener qui consiste à modéliser le PA par un filtre linéaire suivi d'un modèle de PA non-linéaire sans effets mémoire (Bahoura et Park, 2011a; Doufana, 2009; Golio et Golio, 2008). Il inclut un filtre FIR (*Finite Impulse Response*) d'ordre 2 (Doufana et Park, 2008) pour tenir compte des effets mémoire suivi d'un modèle de PA non-linéaire sans mémoire comme celui de Saleh (1981).

L'implantation avec des blocs du logiciel Matlab/Simulink du filtre FIR d'ordre 2 suivi du modèle de PA de Saleh est représentée sur la figure 2 :

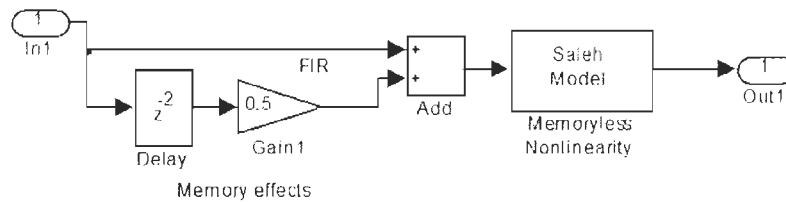


Figure 2: Modèle de PA de Wiener pour la modélisation de la non-linéarité et des effets mémoire.

La fonction de transfert du filtre FIR est :

$$H(z) = 1 + 0.5z^{-2} \quad (1.6)$$

1.3 Caractérisation des distorsions

1.3.1 Point de compression à 1 dB

Le point de compression à 1 dB est le point où le gain de l'amplificateur est réduit de 1 dB par rapport au gain dans sa zone linéaire (Doufana, 2009; Golio et Golio, 2008). Il permet de caractériser la zone de saturation du PA, dans laquelle a lieu les distorsions dans ce circuit (Baudoïn et al., 2007; Marsalek, 2003). Si $x(t)$ est un signal d'entrée, d'un seul ton, d'un système non-linéaire d'ordre 3 :

$$x(t) = A \cos(\omega t) \quad (1.7)$$

alors le signal de sortie $y(t)$ est défini par :

$$y(t) = a_1x(t) + a_2x^2(t) + a_3x^3(t) \quad (1.8)$$

$$y(t) = \frac{a_2A^2}{2} + \left(a_1A + \frac{3a_3A^3}{4}\right) \cos(\omega t) + \frac{a_2A^2}{2} \cos(2\omega t) + \frac{a_3A^3}{4} \cos(3\omega t) \quad (1.9)$$

Le gain en dB à la fréquence fondamentale est (Baudoin et al., 2007) :

$$G = 20 \log \left(a_1 + \frac{3a_3A^2}{4} \right) \quad (1.10)$$

Le gain linéaire G_0 en dB est :

$$G_0 = 20 \log(a_1) \quad (1.11)$$

Le gain à 1 dB de compression est :

$$G_{1dB} = G_0 - 1dB \quad (1.12)$$

Sur la figure 3 est représenté le point de compression à 1 dB.

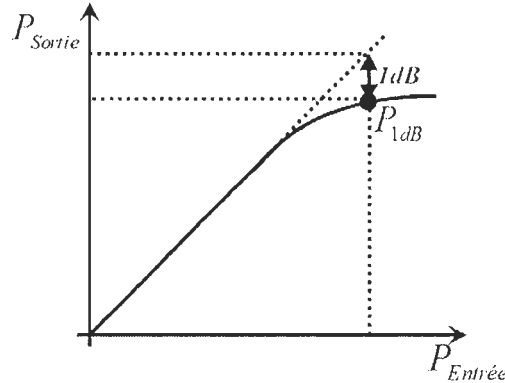


Figure 3: Point de compression à 1 dB.

1.3.2 Mesure du paramètre ACPR avant et après amplification du signal

En général, le paramètre ACPR est le rapport entre la puissance transmise d'un des canaux adjacents sur la puissance transmise du canal principal (Baudoin et al., 2007; Golio et Golio,

2008; Marsalek. 2003). Le paramètre ACPR peut aussi être défini comme étant la puissance transmise d'un des canaux alternatifs sur la puissance transmise du canal principal (Marsalek, 2003). Le canal alternatif est le canal adjacent du canal adjacent. Le paramètre ACPR, illustré sur la figure 4, est exprimé en dBc.

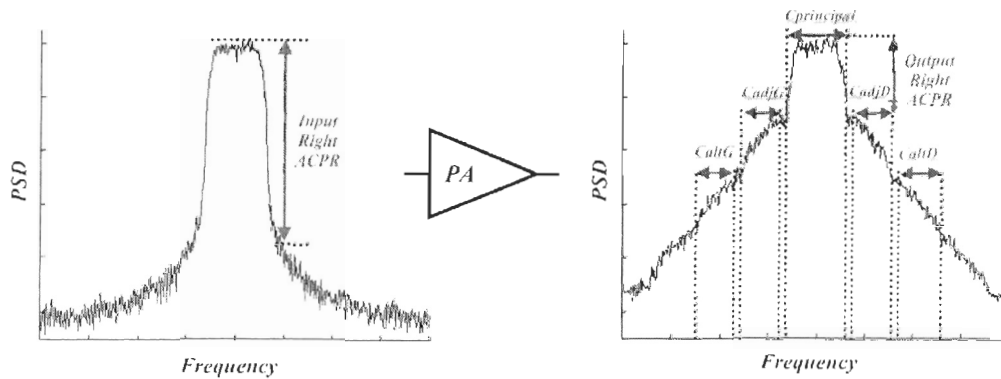


Figure 4: Illustration du paramètre ACPR avant et après amplification du signal.

Dans l'équation 1.13, le paramètre ACPR droit ($ACPR_D$) est calculé par le rapport de la puissance $P(f)$ du canal adjacent droit (C_{adjD}) sur celle du canal principal droit ($C_{principalD}$). Les paramètres f_{adjD_x} et $f_{principalD_x}$ sont respectivement la fréquence du canal adjacent droit et la fréquence du canal principal droit, où x correspond à *min* (minimum) ou *max* (maximum).

$$ACPR_D = 10 \log_{10} \frac{\int_{f_{adjD_{min}}}^{f_{adjD_{max}}} P(f) df}{\int_{f_{principalD_{min}}}^{f_{principalD_{max}}} P(f) df} \quad (1.13)$$

1.3.3 Mesure du paramètre EVM du signal à l'entrée et à la sortie d'un PA

Le paramètre EVM est calculé en analysant la dégradation à l'intérieur de la bande du signal modulé en bande de base après son amplification (Halder et Chatterjee, 2005). Le paramètre EVM, exprimé en pourcentage, est calculé suivant la variation de l'amplitude et de la phase de chaque point de constellation ou symbole tel que représenté par la figure 5. Au passage dans le PA, chaque point de constellation est dévié d'un angle $\Delta\theta$ et présente une variation d'amplitude ΔA . La variation de l'angle et de l'amplitude définissent le paramètre

EVM. Ce paramètre EVM crée par conséquent un décalage ΔI de la partie réelle du symbole et une variation ΔQ de sa partie imaginaire.

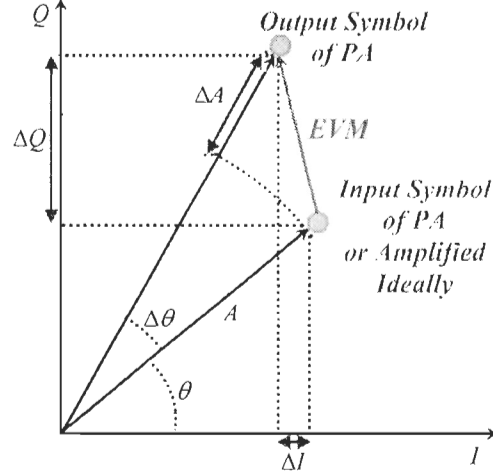


Figure 5: Illustration du paramètre EVM du signal à l'entrée et à la sortie d'un PA.

L'équation de EVM_{RMS} (Ba, 2009) est :

$$EVM_{RMS} = \sqrt{\frac{\sum_{k=1}^N |S_{i,k}(n) - S_{o,k}(n)|^2}{\sum_{k=1}^N S_{i,k}^2(n)}} \quad (1.14)$$

où $S_{i,k}(n)$ et $S_{o,k}(n)$ sont respectivement le signal d'entrée ou le signal idéalement amplifié et le signal de sortie du PA du k^{ieme} symbole. Le paramètre N est le nombre de symbole. Pour le calcul du paramètre EVM dans ce mémoire, une fonction du logiciel Matlab portant sur le calcul du paramètre EVM normalisé basé sur le signal de référence (équation 1.15) est utilisée (Matlab, 2012).

$$EVM_{RMS} = \sqrt{\frac{\frac{1}{N} \sum_{k=1}^N ((I_k(n) - \bar{I}_k(n))^2 + (Q_k(n) - \bar{Q}_k(n))^2)}{\frac{1}{N} \sum_{k=1}^N (I_k^2(n) + Q_k^2(n))}} \quad (1.15)$$

où $I_k(n)$ et $Q_k(n)$ représentent respectivement le composant en phase (I) et le composant en quadrature (Q) du k^{ieme} symbole de $S_{i,k}(n)$. $\bar{I}_k(n)$ et $\bar{Q}_k(n)$ représentent respectivement le composant en phase (I) et le composant en quadrature (Q) du k^{ieme} symbole de $S_{o,k}(n)$. Le paramètre EVM exprimé en pourcent est :

$$EVM_{\%} = 100EVM_{RMS} \quad (1.16)$$

1.3.4 Erreur quadratique moyenne (MSE)

L'algorithme de rétro-propagation utilisé pour l'apprentissage des réseaux de neurones est basé sur la minimisation de l'erreur quadratique moyenne (MSE) pour le $k^{\text{ième}}$ neurone de la couche de sortie. Le signal d'erreur $e_k(n)$ est la différence entre la sortie désirée $d_k(n)$ et la sortie réelle $y_k(n)$ de ce neurone selon l'algorithme développé (Haykin, 2009) :

$$e_k(n) = d_k(n) - y_k(n) \quad (1.17)$$

Le paramètre MSE est défini par :

$$\text{MSE} = \frac{1}{N} \sum_{n=0}^{N-1} E(n) \quad (1.18)$$

où $E(n) = \frac{1}{2}(e_1^2(n) + e_2^2(n))$ est l'énergie totale d'erreur et N est le nombre d'échantillons à chaque époque d'apprentissage.

1.4 Méthodes de linéarisation d'un PA

1.4.1 Linéarisation par contre-réaction (*feedback*)

La linéarisation par contre-réaction (*feedback*) est relativement simple à mettre en œuvre (figure 6). Le signal de sortie V_s du PA est ramené à l'entrée puis soustrait du signal modulé V_e :

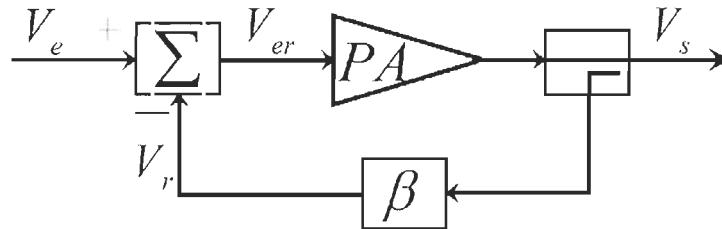


Figure 6: Linéarisation par contre-réaction (*feedback*).

$$V_{er} = V_e - V_r \quad (1.19)$$

où V_r est le signal de la contre-réaction (Marsalek, 2003). Le gain G du PA avec contre-réaction est (Baudoin et al., 2007) :

$$G = \frac{A}{1 + \beta A} \quad (1.20)$$

où A est le gain du PA en boucle ouverte et β est le gain de la réaction. Dans ce cas, le gain G varie en fonction du gain A selon la relation suivante :

$$\frac{\partial G}{\partial A} = \frac{1}{(1 + \beta A)^2} \quad (1.21)$$

Cependant, cette technique souffre de nombreux inconvénients lors de sa mise en œuvre dans un système RF. Elle présente une bande passante limitée lorsqu'elle est utilisée dans un système haute fréquence à large bande (Marsalek. 2003). La présence de la contre-réaction peut entraîner des problèmes de stabilité.

1.4.2 Linéarisation par post-compensation (*feedforward*)

La méthode de linéarisation par post-compensation (*feedforward*), illustrée dans la figure 7, consiste à soustraire du signal de sortie du PA le signal d'erreur. Contrairement à l'approche par contre-réaction, celle par post-compensation est applicable sur des larges bandes et est stable (Baudoin et al., 2007; Golio et Golio, 2008). Quant à son principal inconvénient, il résulte du fait que l'ajout d'une boucle pour une meilleure précision augmente sa taille et son coût. Bien que l'amplificateur à post-compensation soit un concept simple, il est relativement difficile à implanter (Golio et Golio, 2008). L'amplificateur de puissance (PA) introduit des distorsions dans le système. Une partie du signal d'entrée est retardée et ensuite soustraite du signal du PA, annulant le signal d'origine et ne laissant que les distorsions indésirables créées par le PA. Ce signal d'erreur est ensuite ajusté en amplitude et recombinaé avec le signal distordu à la sortie du coupleur, annulant ainsi la distorsion créée par le PA (Golio et Golio, 2008). Le processus de génération du signal d'erreur crée également des non-linéarités, ce qui limite les améliorations réalisées, et est donc un élément essentiel de la conception (Golio et Golio, 2008).

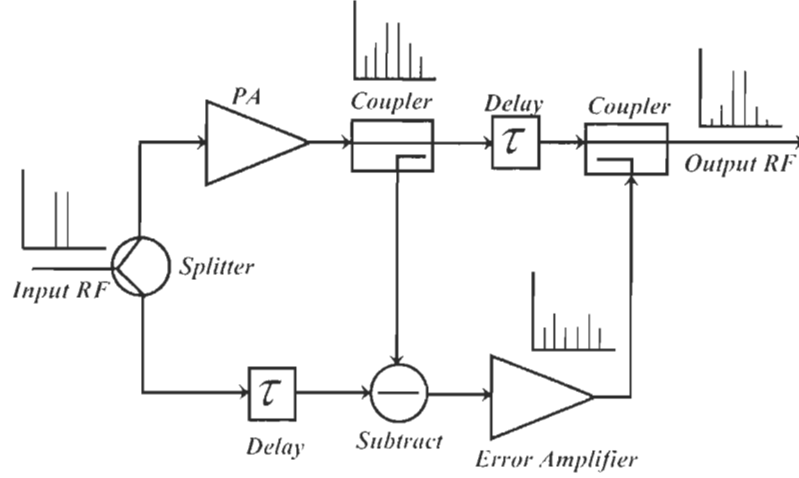


Figure 7: Linéarisation par post-compensation (*feedforward*).

1.4.3 Linéarisation par utilisation de composants non-linéaires (LINC)

La linéarisation par utilisation de composants non-linéaires (LINC) (figure 8) consiste à la décomposition du signal modulé $x(t)$ en deux signaux $x_1(t)$ et $x_2(t)$ à enveloppe constante (Baudoin et al., 2007; Daho et al., 2011). Le signal modulé :

$$x(t) = A(t) \cos(\omega t + \varphi(t)) \quad (1.22)$$

est décomposé en deux signaux à enveloppe constante :

$$x_1(t) = b \cos(\omega t + \varphi(t) + \alpha(t)) \quad (1.23)$$

et

$$x_2(t) = b \cos(\omega t + \varphi(t) - \alpha(t)) \quad (1.24)$$

tel que :

$$x(t) = x_1(t) + x_2(t) \quad (1.25)$$

Les composantes $x_1(t)$ et $x_2(t)$ ont un décalage de phase de $\alpha(t)$ et $-\alpha(t)$ relativement à la phase du signal $x(t)$ (Poitau et al., 2004). L'inconvénient est qu'il traite uniquement les

signaux à enveloppe constante. Le paramètre b est le facteur d'échelle choisi selon la conception du système. Généralement $b = \max(\frac{A(t)}{2})$. Le paramètre $\alpha(t)$ est la décomposition angulaire définie par (Daho et al., 2011) :

$$\alpha(t) = \cos^{-1}\left(\frac{A(t)}{b}\right) \quad (1.26)$$

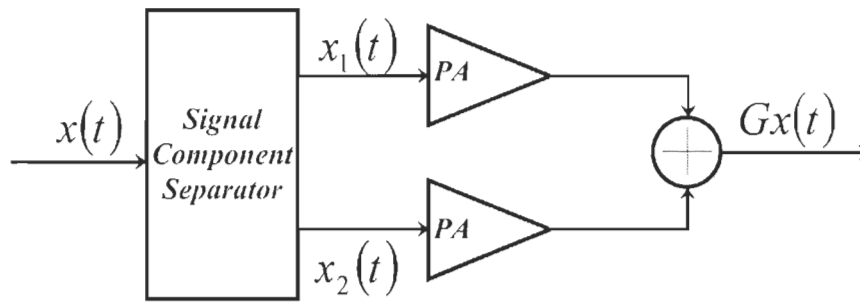


Figure 8: Linéarisation par utilisation de composants non-linéaires (LINC).

1.4.4 Linéarisation par prédistorsion numérique (DPD)

La figure 9 représente en détail le principe de prédistorsion numérique (DPD), qui consiste à compenser les distorsions des caractéristiques AM/AM et AM/PM par prédistorsion du signal avant l'entrée du PA. En d'autres termes, la technique DPD montée en série avec le PA, consiste à utiliser la fonction inverse du PA pour corriger ces distorsions. Le système ne pose pas de problème de stabilité, car la boucle ne sert qu'à faire la mise à jour de l'erreur du bloc de prédistorsion numérique (Baudoin et al., 2007). Cette technique n'a pas de contrainte de largeur de bande ou d'enveloppe constante, d'où sa large flexibilité (Baudoin et al., 2007). Dans la figure 9, $x(n)$ est le signal d'entrée, $z(n)$ est le signal à la sortie du prédistorteur, $y(n)$ est le signal corrigé à la sortie du PA, G est le gain du PA, $z'(n)$ est le signal à la sortie du modèle inverse de PA et $e(n)$ est l'erreur entre $z(n)$ et $z'(n)$. Dans l'architecture de prédistorsion indirecte de la figure 9, le prédistorteur reçoit une copie du modèle inverse de PA. Plusieurs techniques de prédistorsion ont été proposées dans la littérature tels que les réseaux de neurones, le modèle polynomial et les séries de Volterra.

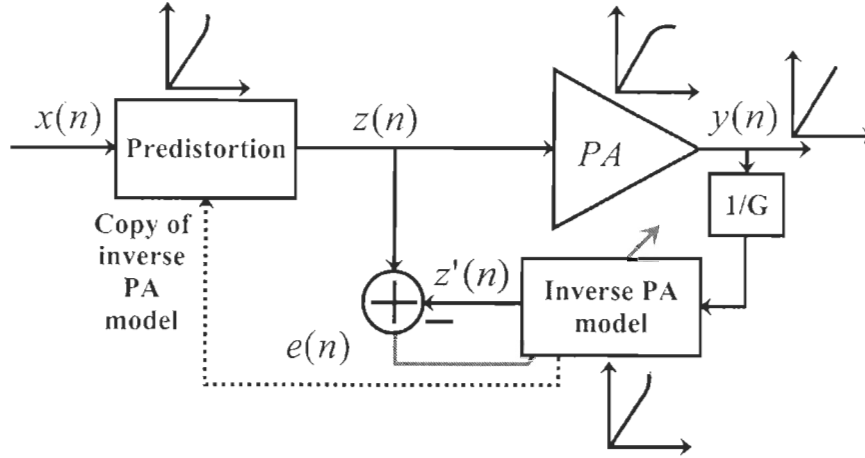


Figure 9: Linéarisation par prédistorsion numérique (DPD).

1.4.4.1 Séries de Volterra

Dans le domaine de temps discret, la sortie $y(n)$ des séries de Volterra est définie comme suit (Zhu et al., 2007) :

$$y(n) = \sum_{r=1}^R y_r(n) \quad (1.27)$$

avec

$$y_r(n) = \sum_{j_1=0}^I \dots \sum_{j_r=0}^I h_r(j_1, \dots, j_r) \prod_{i=1}^r x(n - j_i) \quad (1.28)$$

$y_r(n)$ est le r^{ieme} ordre de non-linéarité, $x(n)$ est l'entrée, $h_r(j_1, \dots, j_r)$ est le r^{ieme} ordre des séries Volterra Kernel et I est la profondeur mémoire (Zhu et al., 2006).

1.4.4.2 Modèle polynomial (PM) avec mémoire

Le modèle polynomial (PM) est un algorithme de prédistorsion utilisé dans la linéarisation des PAs. Le modèle PM qui est composé d'une couche TDL en entrée et d'une fonction non-linéaire statique qui peut être considérée comme une troncature des séries de Volterra générales. Le modèle PM ne prend en considération que les termes diagonaux dans les séries Volterra Kernel (Ku et Kenney, 2003). Le modèle PM de PA prenant en considération les non-linéarités et les effets mémoire peut être représenté comme suit (Ding et al., 2004; Fehri

et Boumaiza, 2011; Hammi et Ghannouchi, 2009; Mkadem, 2010) :

$$z(n) = \sum_{k=0}^K P_k(x(n-k)) \quad (1.29)$$

$x(n)$ est l'enveloppe complexe du signal à l'entrée et $z(n)$ est l'enveloppe complexe du signal à la sortie. La figure 10 représente l'architecture du modèle PM, dans laquelle $P_k(x(n-k))$ représente les polynômes.

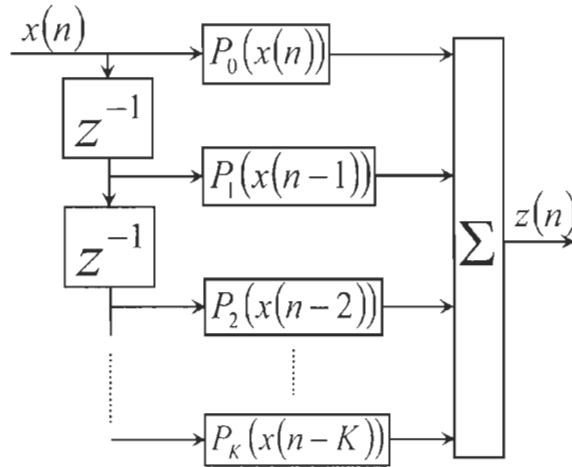


Figure 10: Architecture du modèle polynomial (PM).

Le prédistorteur à base de modèle PM a respectivement comme profondeur mémoire et ordre de non-linéarités K et J .

$$P_k(x(n-k)) = \sum_{j=1}^J a_{j,k} |x(n-k)|^{j-1} x(n-k) \quad (1.30)$$

où $a_{j,k}$ sont les coefficients complexes du modèle polynomial.

1.5 Problématique

L'amplificateur de puissance (PA) des émetteurs de stations de base, qui réalise l'amplification du signal avant son envoi dans le canal de transmission d'une chaîne de communication sans fil, est un composant qui peut présenter de faibles ou de fortes non-linéarités dans sa zone de saturation. Les effets mémoire et les non-linéarités du PA créent des distorsions dans le signal transmis qui correspondent aux remontées spectrales dans les canaux adjacents ACPR

ainsi qu'aux déformations de la constellation EVM dans la bande passante des systèmes de communication sans fil modernes à large bande, tels que la technique W-CDMA (3G) et la technologie LTE-Advanced (4G). La méthode la plus simple pour solutionner ce problème de non-linéarité est de reculer le niveau de puissance (*Back-Off*), mais cela aura comme conséquence la réduction de l'efficacité du PA. Afin d'accroître l'efficacité et de minimiser les distorsions, de nombreuses techniques de linéarisation, telles que *feedback*, *feedforward*, LINC, CALLUM et DPD sont utilisées. Il existe plusieurs modèles de prédistorsion, parmi lesquels les séries de Volterra, le modèle polynomial et plusieurs types de réseaux de neurones.

En raison de leur haute exactitude, leur flexibilité et, par conséquent, leur capacité à bien modéliser avec succès les fonctions non-linéaires des PA en RF, les réseaux de neurones tels que RVTDNN et RVRNN sont utilisés pour la modélisation et la linéarisation par la technique DPD. La prédistorsion numérique (DPD) est en voie de devenir l'une des techniques de linéarisation les plus importantes en raison de ses capacités de traitement rapide du signal numérique avec des puces tels que DSP (*Digital Signal Processor*), FPGA (*Field Programmable Gate Array*) et ASIC (*Application-Specific Integrated Circuit*), en remplacement de la technique *feedforward* qui est actuellement la technique la plus utilisée dans les stations de base. Les fabricants de circuits intégrés programmables, tels que Xilinx et Altera, des PA et des systèmes de support des stations de base, tel que Lucent-Alcatel et Ericsson, proposent différentes solutions de prédistorsion numérique (Gilabert et al., 2008). La puissance de calcul des puces FPGA dépasse celle des puces DSP en raison de leur capacité de traitement en parallèle et du fait qu'elles exécutent plus d'opérations par cycle d'horloge. Les puces FPGA sont généralement plus lentes que les puces ASIC. Cependant, elles ont beaucoup d'avantages dont leur faible coût de développement (Minhui, 2006). Les défis de l'implantation matérielle de n'importe quelle technique DPD résident dans l'atteinte du traitement temps-réel en préservant les performances obtenues avec des modèles logiciels tels que Matlab. L'optimisation des ressources et la réduction du chemin critique font également partie des défis de l'implantation matérielle.

L'optimisation des ressources nécessiterait l'optimisation du nombre de neurones à la couche cachée, le nombre de bits de données et de la taille de la mémoire utilisée dans l'estimation de la tangente hyperbolique. La réduction du chemin critique impliquerait le pipelining des architectures neuronales développées, ce qui signifierait l'optimisation de la fréquence maximale d'opération. Cette dernière consisterait à insérer, à redistribuer et à synchroniser les

retards dans tous les chemins (entrées-sorties) de chaque architecture dans le but d'augmenter sa fréquence maximale d'opération (Bahoura et Park, 2011b), sans toutefois trop retarder ses sorties.

1.6 Objectifs

L'objectif principal du projet est de proposer des systèmes dynamiques à base de réseaux de neurones et implantables sur des puces FPGA, pour la correction en temps-réel des distorsions causées par les non-linéarités et les effets mémoire des PA des stations de base en RF, à l'intérieur (EVM) et à l'extérieur (ACPR) de la bande de fréquence du signal. Pour atteindre cet objectif principal, il faut tout d'abord proposer un algorithme de modélisation. Pour atteindre cet objectif principal, il faut développer des architectures de modélisation capables de reproduire fidèlement le comportement du modèle de référence du PA avant de les appliquer à la linéarisation des PA par prédistorsion adaptative. Enfin, les architectures de modélisation et de prédistorsion devraient être optimisées en ressources et fréquence d'opération pour qu'elles soient implantées sur la puce FPGA Virtex-6 de la carte ML605, en utilisant l'environnement Matlab/Simulink et l'outil Xilinx System Generator.

1.7 Hypothèses

Les travaux de Doufana (2009) ont montré qu'il est possible d'implanter sur puce FPGA un système de linéarisation à base de réseaux de neurones. Cependant, son implantation ultime dépend de deux hypothèses. La première est l'optimisation du réseau de neurones en préservant les mêmes performances. La deuxième est la disponibilité des puces FPGA plus performantes du point de vue ressources.

L'optimisation du nombre de neurones à la couche cachée, le nombre de bits de données et de la taille de la mémoire utilisée dans l'estimation de la tangente hyperbolique pourraient permettre de réduire les ressources tout en conservant quasiment les mêmes performances et ainsi rendre ces architectures de réseaux de neurones implantables sur puce FPGA.

L'utilisation d'un réseau récurrent (RVRNN) devrait permettre de mieux modéliser les effets mémoire par rapport au réseau RVTDNN, car il permettrait non seulement de tenir compte de l'historique des entrées, mais aussi de l'historique des sorties du réseau, matérialisée par le retard unitaire ajouté au *feedback* des sorties vers l'entrée.

Il pourrait être possible d'avoir une meilleure historique des sorties en rajoutant autant de retards au *feedback* du réseau RVRNN qu'il y en a dans la profondeur mémoire en entrée et ainsi obtenir le réseau récurrent NARX. Par conséquent, ce réseau devrait être meilleur que le réseau RVRNN du point de vue modélisation et prédistorsion.

Par la suite, une architecture RVRNN ou NARX, du fait qu'elle possède un historique des entrées et des sorties, pourrait nécessiter moins de neurones à la couche cachée qu'une architecture RVDNN qui a uniquement un historique des entrées. Il est possible de penser que cela pourrait permettre de réduire les ressources estimées lors de l'implantation dans la puce FPGA.

Enfin, la fréquence maximale d'opération de chacune de ces architectures devrait être améliorée par la réduction du chemin critique (pipeline).

1.8 Méthodologie

La démarche consiste à la création des architectures de modélisation et de prédistorsion basées sur les réseaux RVDNN, RVRNN et NARX sous l'environnement Matlab/Simulink et à la réduction du nombre de neurones à la couche cachée en préservant les bonnes qualités de modélisation et de prédistorsion. Elle consiste ensuite à implanter ces réseaux avec les blocs de l'outil XSG en utilisant un nombre suffisant de bits (proche du format double), il s'agit de faire l'optimisation du nombre de bits de données, de la taille des tables LUT et servant à l'estimation de la tangente hyperbolique pour avoir des réseaux implantables sur puce FPGA. Par la suite, le chemin critique est réduit dans le but d'augmenter la fréquence maximale d'opération. Une fois les réseaux optimisés, ils sont implantés sur puce FPGA avec l'outil XSG. Enfin, le réseau RVDNN est comparé aux architectures de modélisation et de prédistorsion à base des réseaux récurrents RVRNN et NARX.

1.9 Contributions apportées

Comme contributions apportées dans ce mémoire, le nombre de bits de données et le nombre de neurones à la couche cachée du réseau RVDNN ont graduellement été réduits. La fréquence maximale d'opération a également été améliorée par pipeline. Cette approche a permis également l'implantation sur une puce FPGA de deux réseaux de neurones récurrents (RVRNN et NARX). La vérification du bon fonctionnement de ces réseaux de neurones dans

une puce FPGA a été faite par l'interface *JTAG Hardware Co-simulation*. Puis, les réseaux de neurones RVTDDN, RVRNN et NARX ont été comparés en termes de modélisation et linéarisation.

Une partie du travail consacré à la comparaison des architectures de modélisation RVTDDN pseudo-conventionnelle et RVRNN pseudo-conventionnelle, ainsi que la comparaison des architectures de modélisation RVTDDN pseudo-conventionnelle et RVTDDN avec pipeline ont fait l'objet de publications dans des conférences internationales (Ntouné Ntouné et al., 2012a,b). Enfin, dans le mémoire, les architectures RVTDDN, RVRNN et NARX ont été intégrées dans des systèmes de prédistorsion numérique à apprentissage direct qui donneront probablement lieu à plusieurs autres publications.

1.10 Organisation du mémoire

Après l'introduction générale, le chapitre 2 développe la modélisation des PA avec les réseaux de neurones. Dans la première partie de ce chapitre, les algorithmes de modélisation RVTDDN, RVRNN et NARX de 6 et 8 neurones à la couche cachée conventionnels, pseudo-conventionnels et avec pipeline sont développés, et, de ces algorithmes découlent les équations de propagation et de rétro-propagation. Dans la deuxième partie de ce chapitre, ces algorithmes sont implantés avec les blocs de l'outil XSG. Par la suite, ces architectures de modélisation sont implantées sur la puce FPGA de la carte ML605 et co-simulées après estimation des ressources. Enfin, dans ce chapitre, les courbes comparatives des différentes architectures sont analysées et commentées.

Dans le chapitre 3, la prédistorsion numérique (DPD) des PA avec les réseaux de neurones est développée. Les algorithmes de prédistorsion à apprentissage direct du réseau RVTDDN, du réseau RVRNN et du réseau NARX conventionnels, pseudo-conventionnels et avec pipeline à 6 et 8 neurones à la couche cachée sont développés, et, de ces algorithmes découlent les équations de propagation et de rétro-propagation de la prédistorsion. Ces architectures de pré-distorsion sont ensuite implantées sur la puce FPGA Virtex-6 de la carte ML605 et co-simulées après estimation des ressources. À la fin de ce chapitre, les courbes comparatives des différentes architectures sont analysées et commentées.

Le chapitre 4 est la conclusion générale et fait état des résultats obtenus dans ce projet et des différentes perspectives liées aux travaux futurs associés à la continuité du projet.

CHAPITRE 2

MODÉLISATION DES AMPLIFICATEURS DE PUISSANCE

2.1 Principe de modélisation des amplificateurs de puissance

La modélisation du comportement d'un amplificateur de puissance (PA) est une étape incontournable dans la conception d'un système de linéarisation basé sur la prédistorsion numérique (DPD) car cette dernière consiste tout simplement à utiliser un modèle inverse du PA. Plusieurs algorithmes de modélisation existent, parmi lesquels le modèle polynomial (Ye et al., 2008a), les séries de Volterra (Mkadem et al., 2011; Yu et al., 2011), ainsi que plusieurs architectures de réseaux de neurones tel que RNN (Cao et Zhang, 2009), RVF-TDNN (Rawat et al., 2010), TDNN (Hwangbo et al., 2006), RVRNN (Rawat et al., 2010), RVTDNN (Bahoura et Park, 2011a; Boumaiza et Mkadem, 2009; Doufana et Park, 2008; Liu et al., 2004) et NARX (Haykin, 2009). Dans ce mémoire, seules les architectures des réseaux RVTDNN, RVRNN et NARX seront développées. La figure 11 illustre le principe de la modélisation (Doufana, 2009; Doufana et Park, 2008) des PA pour un signal en bande de base. Les erreurs en phase $e_I(n)$ et en quadrature de phase $e_Q(n)$ sont respectivement la différence entre les sorties désirées $d_I(n)$ et $d_Q(n)$ avec les sorties $y_I(n)$ et $y_Q(n)$ du modèle de PA. Les erreurs sont utilisées dans l'adaptation itérative du modèle (mise à jour continue des poids synaptiques du réseau de neurones) par l'algorithme de rétro-propagation. Le perceptron multicouches (MLP) qui est à la base de ces réseaux, est constitué d'une couche d'entrée, d'une ou plusieurs couches cachées et d'une couche de sortie. Dans le réseau MLP, chaque couche du réseau est connectée avec tous les neurones de la couche précédente. Lors de la propagation, le signal se déplace couche par couche de l'entrée jusqu'à la sortie du réseau (Haykin, 2009). Pour les différents réseaux (RVTDNN, RVRNN et NARX), N_0 est la dimension du vecteur d'entrée \mathbf{x} , N_1 est le nombre de neurones de la couche cachée et N_2 est le nombre de neurones de la couche de sortie.

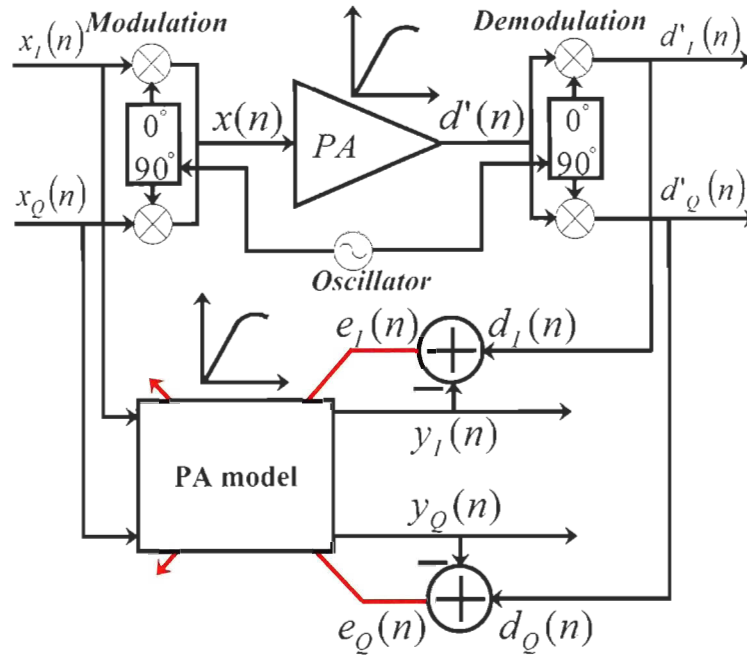


Figure 11: Principe de modélisation des PA pour un signal en bande de base.

La figure 12 représente un réseau MLP à une couche cachée, où HN et ON représentent respectivement les neurones de la couche cachée et de la couche de sortie. En fonction de la profondeur mémoire m choisie, le nombre d'entrées est $N_0 = 2m + 2$ pour le réseau RVTDDN, $N_0 = 2m + 4$ pour le réseau RVRNN et $N_0 = 4m + 4$ pour le réseau NARX. Pour tous les réseaux, le nombre de sorties N_2 est égal à 2.

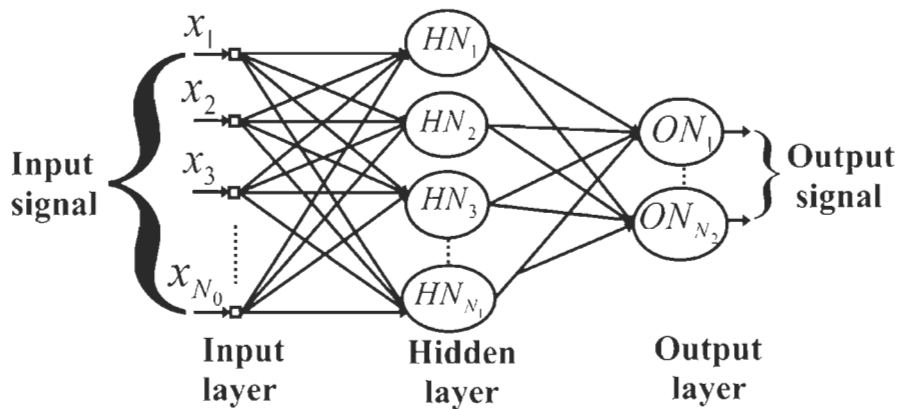


Figure 12: Architecture d'un perceptron multicouche (MLP) à une couche cachée.

2.2 Architectures conventionnelles des réseaux RVDNN, RVRNN et NARX pour la modélisation des PAs

Les architectures conventionnelles des réseaux de neurones (RVDNN, RVRNN et NARX) sont les implantations originales de ces réseaux. La profondeur mémoire de toutes ces architectures a été fixée à $m=2$, car le filtre FIR du modèle de référence de Wiener utilisé pour l'introduction des effets mémoire (voir section 1.2.2), est d'ordre 2. Donc, pour que le réseau NARX, le réseau RVRNN et le réseau RVDNN puissent modéliser correctement les distorsions dues aux non-linéarités et aux effets mémoire, il faut choisir en entrée une profondeur mémoire supérieure ou égale à l'ordre du filtre. Pour toutes les architectures de modélisation des PA développées, la fonction d'activation des neurones de la couche cachée choisie après une série de tests est la fonction tangente hyperbolique (Doufana et Park, 2008; Liu et al., 2004; Rawat et al., 2010) :

$$\varphi_h(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.1)$$

La fonction tangente hyperbolique est bipolaire, ce qui permet d'obtenir de meilleurs résultats qu'avec la fonction sigmoïde qui est unipolaire positive. Les poids synaptiques et les biais sont aléatoirement initialisés avec la fonction *randn* du logiciel Matlab. Ensuite, ils sont itérativement ajustés par l'algorithme d'apprentissage de rétro-propagation (Haykin, 2009) afin de converger vers leurs valeurs optimales. Les modes *Batch* et séquentiel sont les deux modes d'apprentissage possibles (Haykin, 2009). Dans le mode *batch* d'apprentissage supervisé, l'ajustement des poids synaptiques et des biais du réseau MLP est fait après la présentation de tous les échantillons d'une époque d'apprentissage. Dans le mode séquentiel, l'ajustement des poids synaptiques et des biais du réseau de neurones est fait après la présentation de chaque échantillon. Pour une implantation temps-réel, le mode d'apprentissage séquentiel est utilisé (Doufana, 2009). Les symboles \oplus , \otimes et \sum dans les différentes architectures représentent respectivement les opérations d'addition, de multiplication et de somme de leurs entrées. La fonction d'activation de la couche de sortie est linéaire (Bahoura et Park, 2011a) :

$$\varphi_o(x) = x \quad (2.2)$$

2.2.1 Architecture conventionnelle du réseau RVTDDN

Le réseau RVTDDN conventionnel est basé sur un réseau MLP non-récurrent. La figure 13 illustre l'architecture conventionnelle de ce réseau.

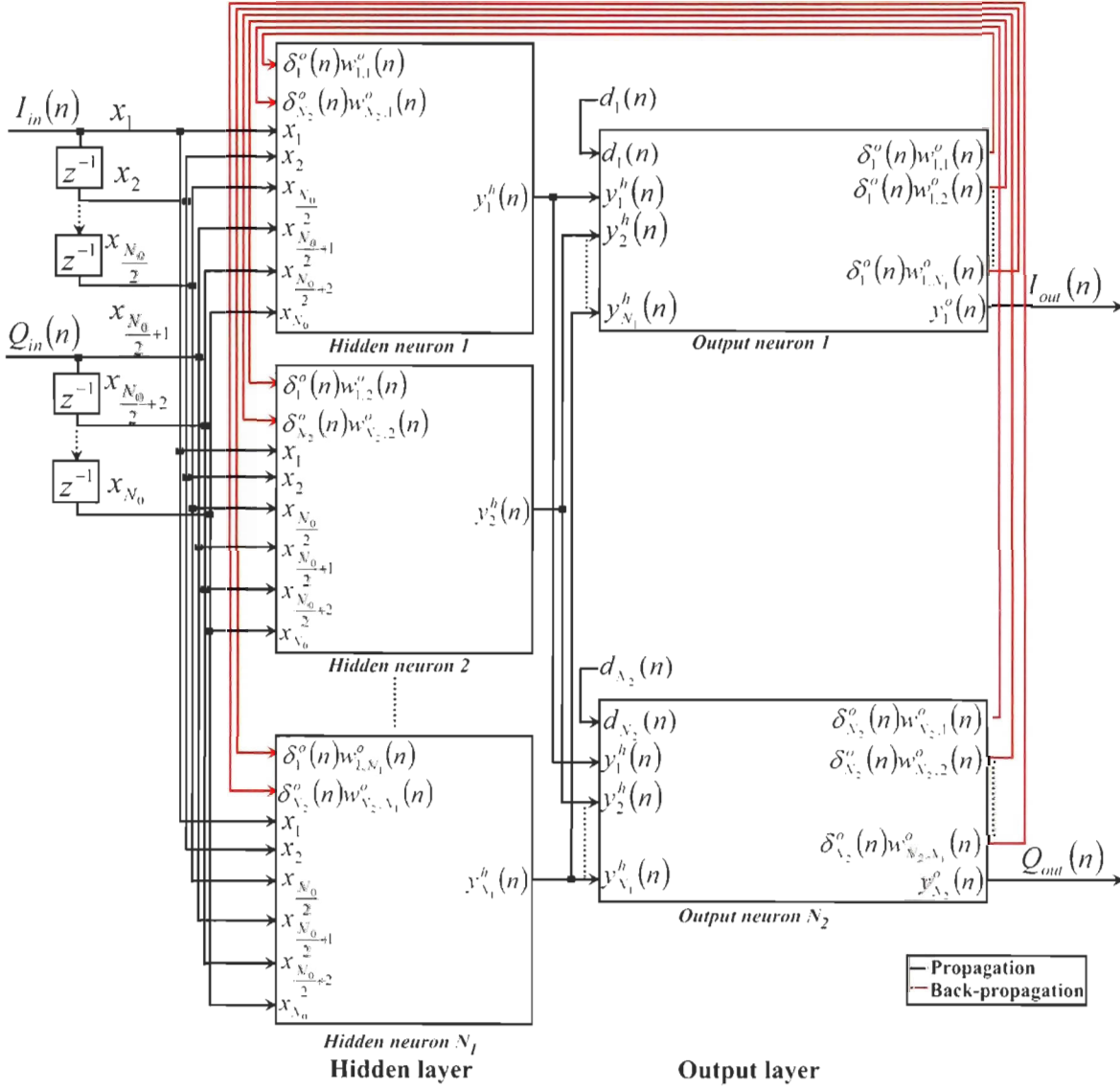


Figure 13: Architecture conventionnelle du réseau RVTDDN.

La taille du vecteur d'entrée \mathbf{x} du réseau RVTDDN est $N_0 = 2m + 2$.

$$\mathbf{x} := [I_{in}(n), \dots, I_{in}(n-m), Q_{in}(n), \dots, Q_{in}(n-m)] \quad (2.3)$$

où m est la profondeur mémoire, $I_{in}(n)$ et $Q_{in}(n)$ sont respectivement les entrées en phase et en quadrature de phase. $I_{out}(n)$ et $Q_{out}(n)$ sont respectivement les sorties en phase et en quadrature de phase. Chaque neurone effectue une somme pondérée de ses entrées et est suivi d'une fonction d'activation.

2.2.1.1 Propagation des signaux de l'architecture conventionnelle de modélisation du réseau RVTDNN

La sortie du neurone j dans la couche cachée est donnée par :

$$y_j^h(n) = \varphi_h \left(v_j^h(n) \right) \quad j = 1, \dots, N_1 \quad (2.4)$$

avec

$$v_j^h(n) = \sum_{i=1}^{N_0} w_{j,i}^h(n) x_i(n) + w_{j,0}^h(n) \quad i = 1, \dots, N_0 \quad (2.5)$$

où $w_{j,0}^h(n)$ est le biais de ce neurone et $w_{j,i}^h(n)$ représente le poids de sa connexion à l'entrée i de la couche d'entrée.

La sortie du neurone k à la couche de sortie est définie par :

$$y_k^o(n) = \varphi_o \left(v_k^o(n) \right) \quad k = 1, \dots, N_2 \quad (2.6)$$

avec

$$v_k^o(n) = \sum_{j=1}^{N_1} w_{k,j}^o(n) y_j^h(n) + w_{k,0}^o(n) \quad (2.7)$$

où $w_{k,0}^o(n)$ est le biais de ce neurone et $w_{k,j}^o(n)$ représente le poids de sa connexion au neurone j de la couche cachée.

2.2.1.2 Rétro-propagation de l'erreur de l'architecture conventionnelle de modélisation du réseau RVTDNN

Le calcul du signal d'erreur pour la mise à jour des poids se fait avec l'algorithme rétro-propagation (Haykin, 2009). Le signal d'erreur $e_k(n)$ est la différence entre la sortie désirée

$d_k(n)$ fournie par le modèle de référence du PA et la sortie réelle $y_k(n)$ du réseau de neurones de modélisation du PA en cours d'apprentissage (Haykin, 2009) :

$$e_k(n) = d_k(n) - y_k(n) \quad (2.8)$$

2.2.1.3 Structure des neurones de la couche de sortie

La figure 14 représente la structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture conventionnelle du réseau RVTDDN.

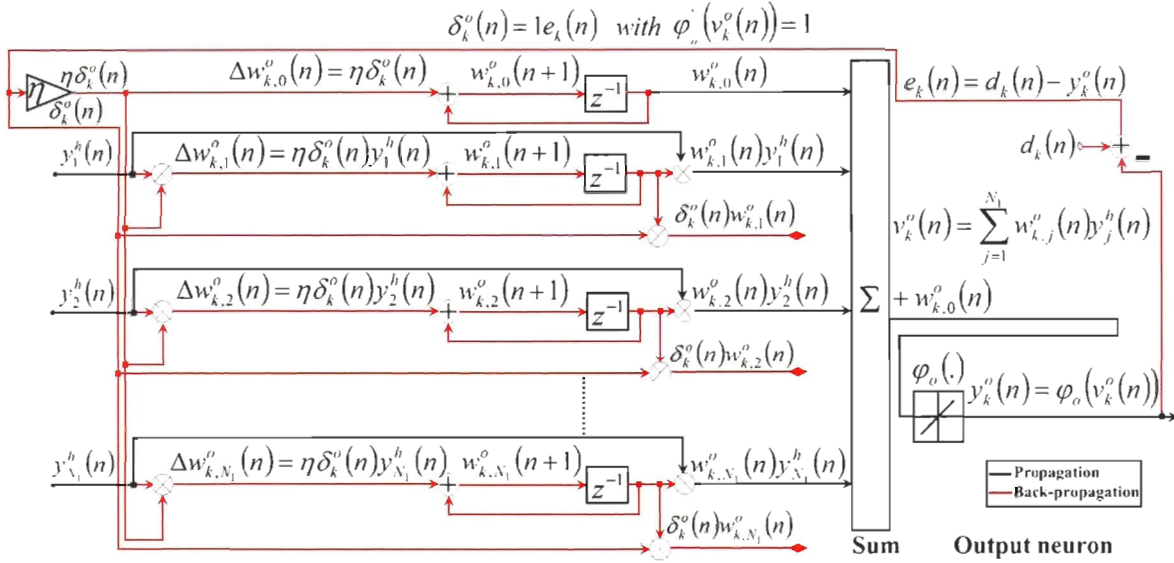


Figure 14: Structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture conventionnelle du réseau RVTDDN.

La variation du poids synaptique $w_{k,j}^o(n)$ est donnée par :

$$\Delta w_{k,j}^o(n) = \eta \delta_k^o(n) y_j^h(n) \quad (2.9)$$

Le taux d'apprentissage a été fixé à 0.015625, parce que la valeur du taux d'apprentissage ne doit être ni trop petite, ni trop grande, pour assurer une convergence rapide sans risque de divergence et cette valeur choisie fonctionne avec toutes les architectures développées. Cette valeur précise permet de réaliser la multiplication par η en utilisant un simple décalage de 6 bits à droite.

Le gradient local est défini par :

$$\delta_k^o(n) = e_k(n)\varphi'_o(v_k^o(n)) \quad (2.10)$$

où $\varphi'_o(v_k^o(n)) = 1$, car la fonction d'activation à la couche de sortie est une fonction linéaire.

L'ajustement du poids synaptique $w_{k,j}^o(n)$ est défini par :

$$w_{k,j}^o(n+1) = w_{k,j}^o(n) + \Delta w_{k,j}^o(n) \quad (2.11)$$

La correction $\Delta w_{k,0}^o(n)$ appliquée au biais est un cas particulier de la correction des poids $\Delta w_{k,j}^o(n)$ et est définie par :

$$\Delta w_{k,0}^o(n) = \eta \delta_k^o(n) \quad (2.12)$$

2.2.1.4 Structure des neurones de la couche cachée

La structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture conventionnelle du réseau RVTDNN est représentée sur la figure 15.

La variation du poids synaptique $w_{j,i}^h(n)$ est donnée par :

$$\Delta w_{j,i}^h(n) = \eta \delta_j^h(n) x_i(n) \quad (2.13)$$

où le gradient local est défini par :

$$\delta_j^h(n) = \varphi'_h(v_j^h(n)) \sum_{k=1}^{N_2} \delta_k^o(n) w_{k,j}^o(n) \quad (2.14)$$

pour la tangente hyperbolique utilisée à la couche cachée :

$$\varphi'_h(v_j^h(n)) = 1 - \varphi_h^2(v_j^h(n)) \quad (2.15)$$

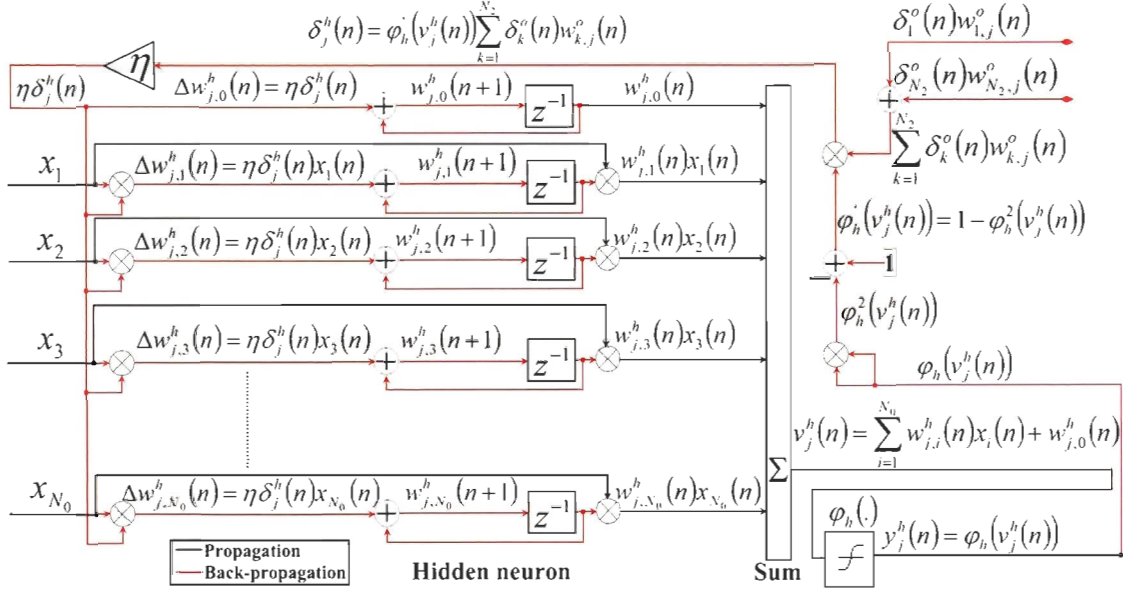


Figure 15: Structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture conventionnelle du réseau RVTDDN.

La mise à jour des poids est définie par :

$$w_{j,i}^h(n+1) = w_{j,i}^h(n) + \Delta w_{j,i}^h(n) \quad (2.16)$$

Dans ce cas, la correction $\Delta w_{j,i}^h(n)$ appliquée aux biais est :

$$\Delta w_{j,0}^h(n) = \eta \delta_j^h(n) \quad (2.17)$$

2.2.2 Architecture conventionnelle du réseau RVRNN

L'habileté des réseaux de neurones à modéliser les non-linéarités dynamiques des PA pourrait être améliorée en ajoutant une réaction des sorties vers l'entrée (Golio et Golio, 2008). Le réseau RVRNN conventionnel est un réseau de neurones récurrent, qui a en plus de ces entrées, la réaction de ses sorties retardés de 1 délai. Cette architecture, implantée pour la première fois avec l'outil XSG, est proposée dans le but de mieux modéliser les effets mémoire par rapport au réseau RVTDDN, car le réseau RVRNN permet non seulement de tenir compte de l'historique des entrées, mais aussi de l'historique des sorties, matérialisée par le retard unitaire ajouté à la réaction des sorties vers l'entrée.

La différence majeure entre le réseau RVTDDN et le réseau RVRNN représentés respectivement par la figure 13 et la figure 16 réside dans le nombre de leurs entrées.

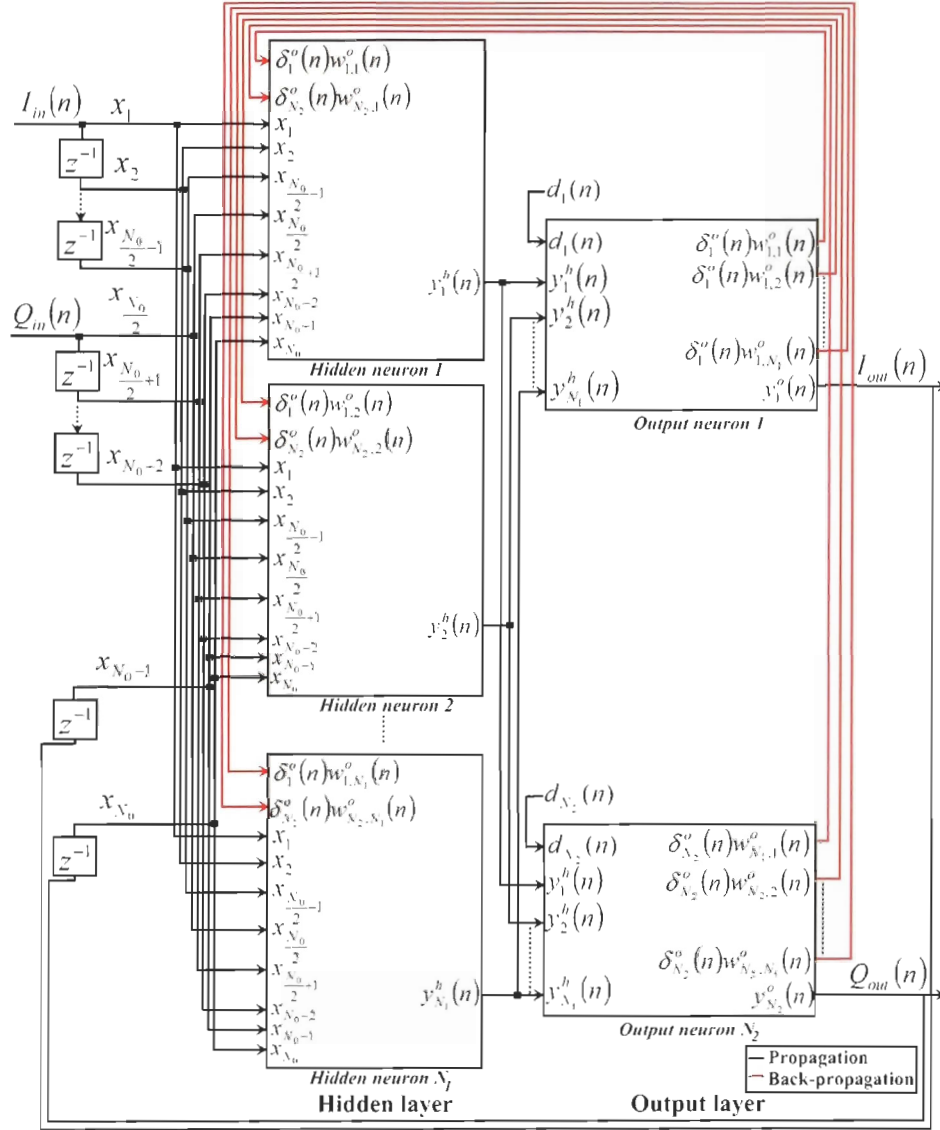


Figure 16: Architecture conventionnelle du réseau RVRNN.

La taille du vecteur d'entrée \mathbf{x} du réseau RVRNN conventionnel est $N_0 = 2m + 4$.

$$\mathbf{x} := [I_{in}(n), \dots, I_{in}(n - m), Q_{in}(n), \dots, Q_{in}(n - m), I_{out}(n - 1), Q_{out}(n - 1)] \quad (2.18)$$

où m est la profondeur mémoire. Les équations de propagation et de rétro-propagation, la

structure des neurones de la couche cachée et la structure des neurones de la couche de sortie du réseau RVRNN conventionnel sont les mêmes que celles du réseau RVTDDN conventionnel.

2.2.3 Architecture conventionnelle du réseau NARX

L'architecture conventionnelle du réseau NARX est basée sur celle du réseau RVTDDN, mais a en plus de ses entrées, la réaction de ses sorties retardé de 1 délai plus le même nombre de retards utilisé pour la profondeur mémoire en entrée. Cette architecture, implantée pour la première fois avec l'outil XSG, est proposée dans le but d'avoir une meilleure historique des sorties. La figure 17 illustre l'architecture conventionnelle du réseau NARX. Pour un temps donné n , le signal en bande de base à l'entrée de chaque neurone est défini comme étant un vecteur \mathbf{x} de $N_0 = 4m + 4$ composants pour le réseau NARX, où m est la profondeur mémoire.

$$\mathbf{x} = [I_{in}(n), \dots, I_{in}(n - m), Q_{in}(n), \dots, Q_{in}(n - m), \\ I_{out}(n - 1), \dots, I_{out}(n - 1 - m), Q_{out}(n - 1), \dots, Q_{out}(n - 1 - m)] \quad (2.19)$$

Les équations de propagation et de rétro-propagation du réseau NARX conventionnel sont les mêmes que celles du réseau RVTDDN conventionnel. La structure du $k^{\text{ième}}$ neurone de la couche de sortie reste la même que celle de la figure 14. La structure du $j^{\text{ième}}$ neurone de la couche cachée reste également la même que celle de l'architecture conventionnelle du réseau RVTDDN de la figure 15. En résumé, les architectures des réseaux RVTDDN, RVRNN et NARX représentées respectivement aux figures 13, 16 et 17 ne diffèrent que par leur nombre d'entrées. L'implantation des fonctions d'activation (tangente hyperbolique) dans ces réseaux nécessite des mémoires RAM distribuées qui n'introduisent pas de retards. Le nombre de ce type de mémoires disponible sur la puce FPGA Virtex-6 de la carte ML605 ne permet pas d'implanter des réseaux fonctionnels.

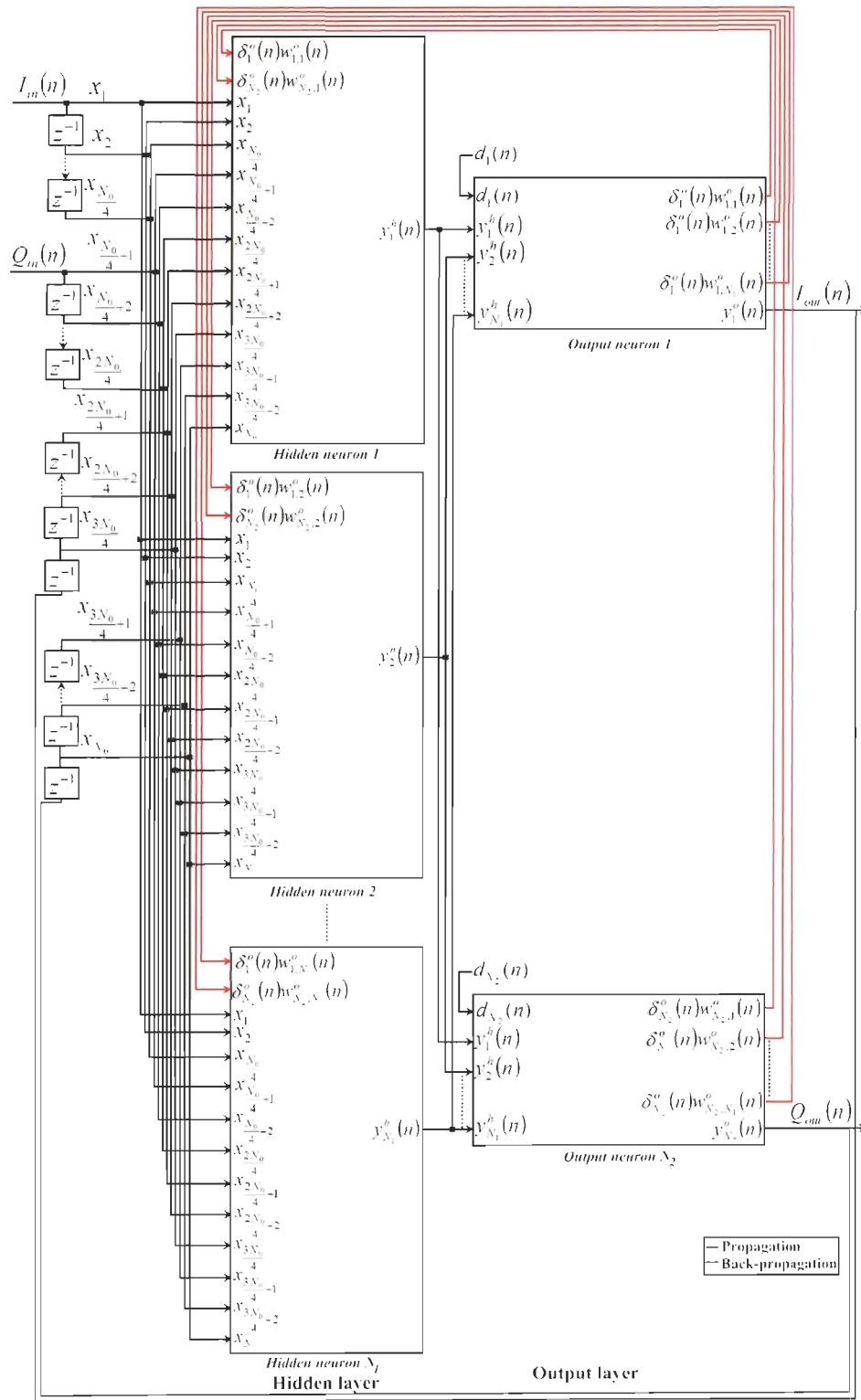


Figure 17: Architecture conventionnelle du réseau NARX.

2.3 Architectures pseudo-conventionnelles des réseaux NARX, RVRNN et RVTDDN pour la modélisation des PAs

L'implantation des réseaux de modélisation fonctionnels ne peut être fait qu'avec les blocs RAM pour l'implantation des fonctions d'activation (tangente hyperbolique). Ces blocs, disponibles en grand nombre sur la puce FPGA, introduisent au moins 1 délai. L'algorithme de rétro-propagation de l'erreur a été ajusté en conséquence. Les retards ajoutés sont hachurés en gris.

2.3.1 Architecture pseudo-conventionnelle du réseau RVTDDN

La figure 18 illustre l'architecture pseudo-conventionnelle du réseau RVTDDN. Les équations de l'architecture pseudo-conventionnelle ont été ajustées pour tenir compte du retard introduit par la fonction d'activation de la couche cachée.

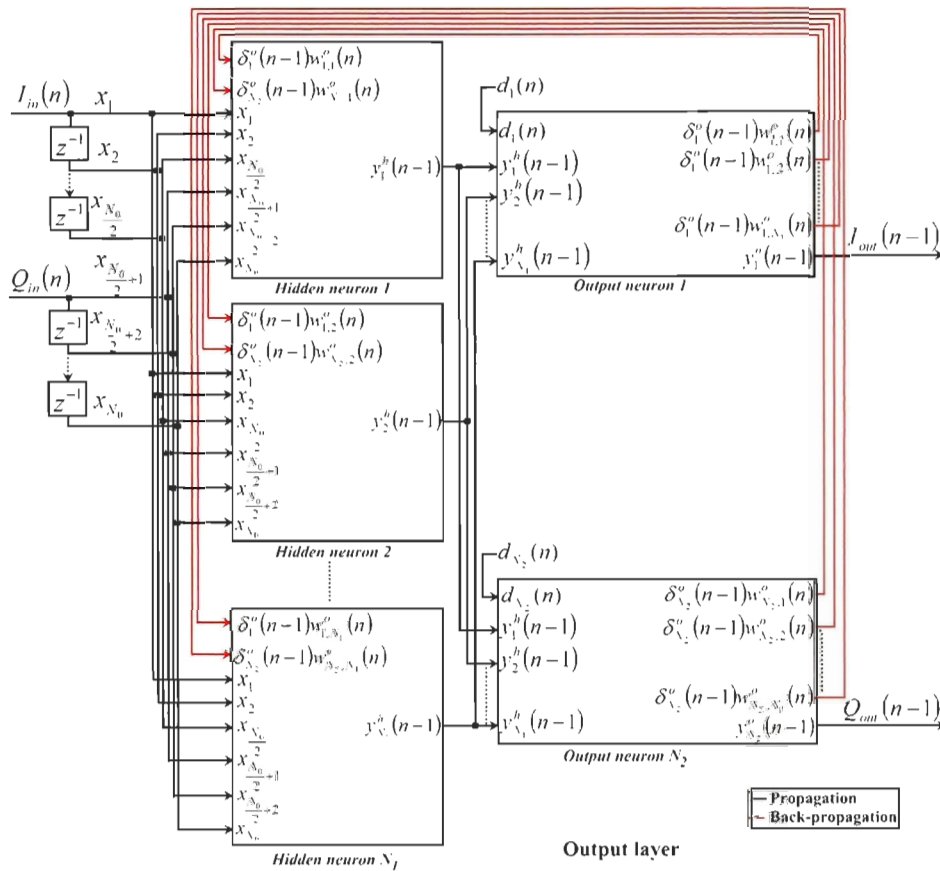


Figure 18: Architecture pseudo-conventionnelle du réseau RVTDDN.

2.3.1.1 Propagation des signaux de l'architecture RVTDNN pseudo-conventionnelle de modélisation

La sortie du $j^{\text{ième}}$ neurone de la couche cachée est retardée de 1 délai à cause de la fonction d'activation :

$$y_j^h(n-1) = \varphi_h(v_j^h(n)) \quad j = 1, \dots, N_1 \quad (2.20)$$

avec

$$v_j^h(n) = \sum_{i=1}^{N_0} w_{j,i}^h(n)x_i(n) + w_{j,0}^h(n) \quad i = 1, \dots, N_0 \quad (2.21)$$

où $w_{j,0}^h(n)$ est le biais de ce neurone et $w_{j,i}^h(n)$ représente le poids de connexion au neurone i à l'entrée de la couche d'entrée.

Ce retard se propage à la sortie du neurone k de la couche de sortie :

$$y_k^o(n-1) = \varphi_o(v_k^o(n-1)) \quad k = 1, \dots, N_2 \quad (2.22)$$

avec

$$v_k^o(n-1) = \sum_{j=1}^{N_1} w_{k,j}^o(n)y_j^h(n-1) + w_{k,0}^o(n) \quad (2.23)$$

où $w_{k,0}^o(n)$ est le biais de ce neurone et $w_{k,j}^o(n)$ représente le poids de connexion au $j^{\text{ième}}$ neurone de la couche cachée. Le délai introduit à la couche cachée par la fonction d'activation se propage jusqu'à la couche de sortie.

2.3.1.2 Rétro-propagation de l'erreur de l'architecture RVTDNN pseudo-conventionnelle de modélisation

La sortie dérivée du modèle de référence du PA est retardée de 1 délai pour qu'elle soit synchronisée avec la sortie réelle du modèle en apprentissage. L'erreur retardée est définie par :

$$e_k(n-1) = d_k(n-1) - y_k(n-1) \quad (2.24)$$

2.3.1.3 Structure des neurones de la couche de sortie

La figure 19 représente la structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture pseudo-conventionnelle du réseau RVTDDN.

La variation des poids synaptiques $w_{k,j}^o(n)$ des neurones de la couche de sortie est définie par :

$$\Delta w_{k,j}^o(n) = \eta \delta_k^o(n-1) y_j^h(n-1) \quad (2.25)$$

où $\eta = 0,015625$ est le taux d'apprentissage. Le gradient local est défini par :

$$\delta_k^o(n-1) = e_k(n-1) \varphi'_o(v_k^o(n-1)) \quad (2.26)$$

où $\varphi'_o(v_k^o(n-1)) = 1$, car la fonction d'activation à la couche de sortie est une fonction linéaire.

L'ajustement des poids est défini par :

$$w_{k,j}^o(n+1) = w_{k,j}^o(n) + \Delta w_{k,j}^o(n) \quad (2.27)$$

La correction $\Delta w_{k,0}^o(n)$ appliquée au biais est un cas particulier de la correction des poids $\Delta w_{k,j}^o(n)$ et est définie par :

$$\Delta w_{k,0}^o(n) = \eta \delta_k^o(n-1) \quad (2.28)$$

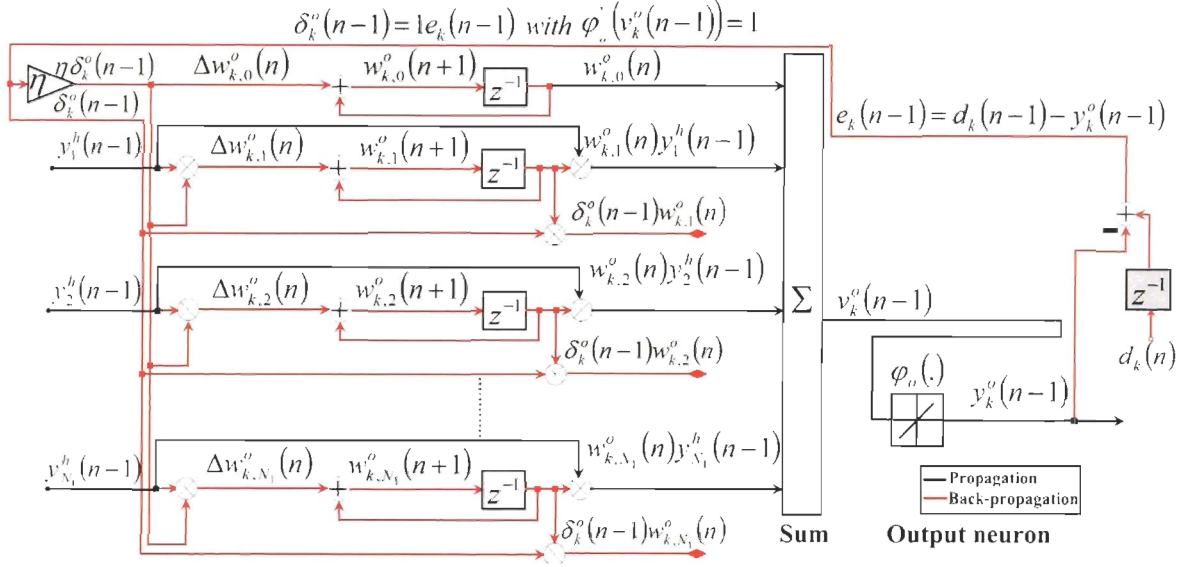


Figure 19: Structure du $k^{\text{ième}}$ neurone de la couche de sortie de l'architecture pseudo-conventionnelle du réseau RVTDDN.

2.3.1.4 Structure des neurones de la couche cachée

Sur la figure 20, la structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture pseudo-conventionnelle du réseau RVTDDN est représentée. La variation des poids synaptiques $w_{j,i}^h(n)$ des neurones de la couche cachée est définie par :

$$\Delta w_{j,i}^h(n) = \eta \delta_j^h(n-1) x_i(n-1) \quad (2.29)$$

où le gradient local est :

$$\delta_j^h(n-1) = \varphi_h'(v_j^h(n-1)) \sum_{k=1}^{N_2} \delta_k^o(n-1) w_{k,j}^o(n) \quad (2.30)$$

pour une fonction tangente hyperbolique utilisée à la couche cachée :

$$\varphi_h'(v_j^h(n-1)) = 1 - \varphi_h^2(v_j^h(n-1)) \quad (2.31)$$

La mise à jour des poids synaptiques à la couche cachée est :

$$w_{j,i}^h(n+1) = w_{j,i}^h(n) + \Delta w_{j,i}^h(n) \quad (2.32)$$

La correction $\Delta w_{j,i}^h(n)$ appliquée aux biais est :

$$\Delta w_{j,0}^h(n) = \eta \delta_j^h(n-1) \quad (2.33)$$

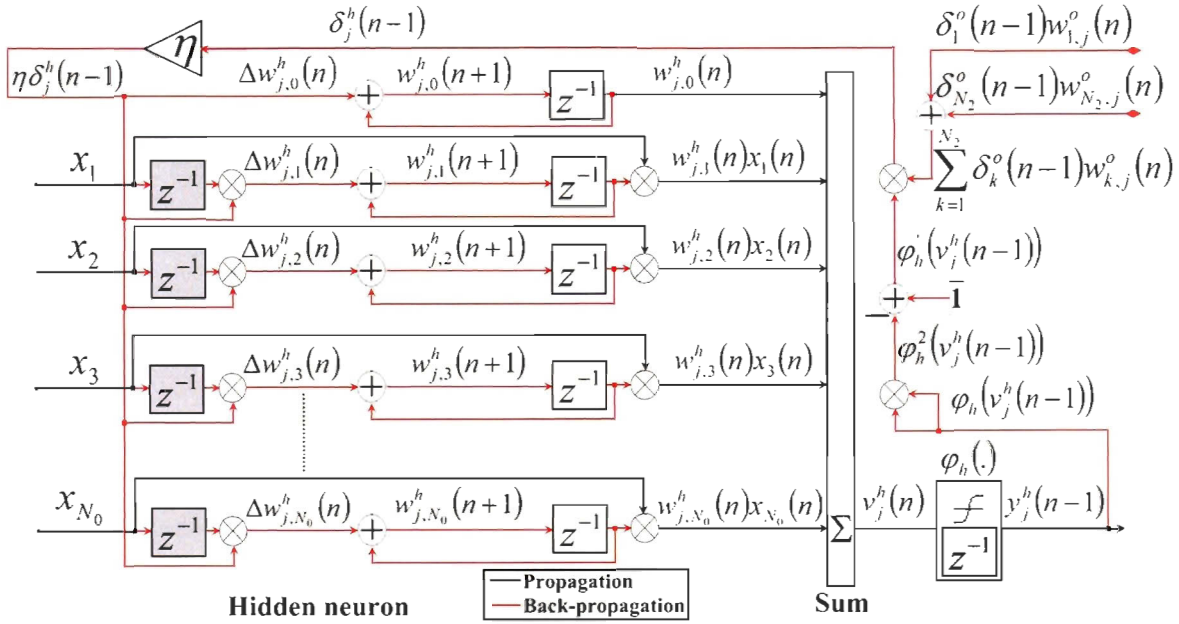


Figure 20: Structure du $j^{\text{ième}}$ neurone de la couche cachée de l'architecture pseudo-conventionnelle du réseau RVTDDN.

2.3.2 Architecture pseudo-conventionnelle du réseau RVRNN

La figure 21 représente l'architecture pseudo-conventionnelle du réseau RVRNN. Les équations de propagation et de rétro-propagation de l'architecture pseudo-conventionnelle du réseau RVRNN sont les mêmes que celles du réseau RVTDDN et la structure de ses neurones de la couche de sortie est la même que ceux du réseau RVTDDN pseudo-conventionnel de la figure 19. La structure des neurones de la couche cachée de l'architecture pseudo-conventionnelle du réseau RVRNN est la même que ceux du réseau RVTDDN de la figure 20.

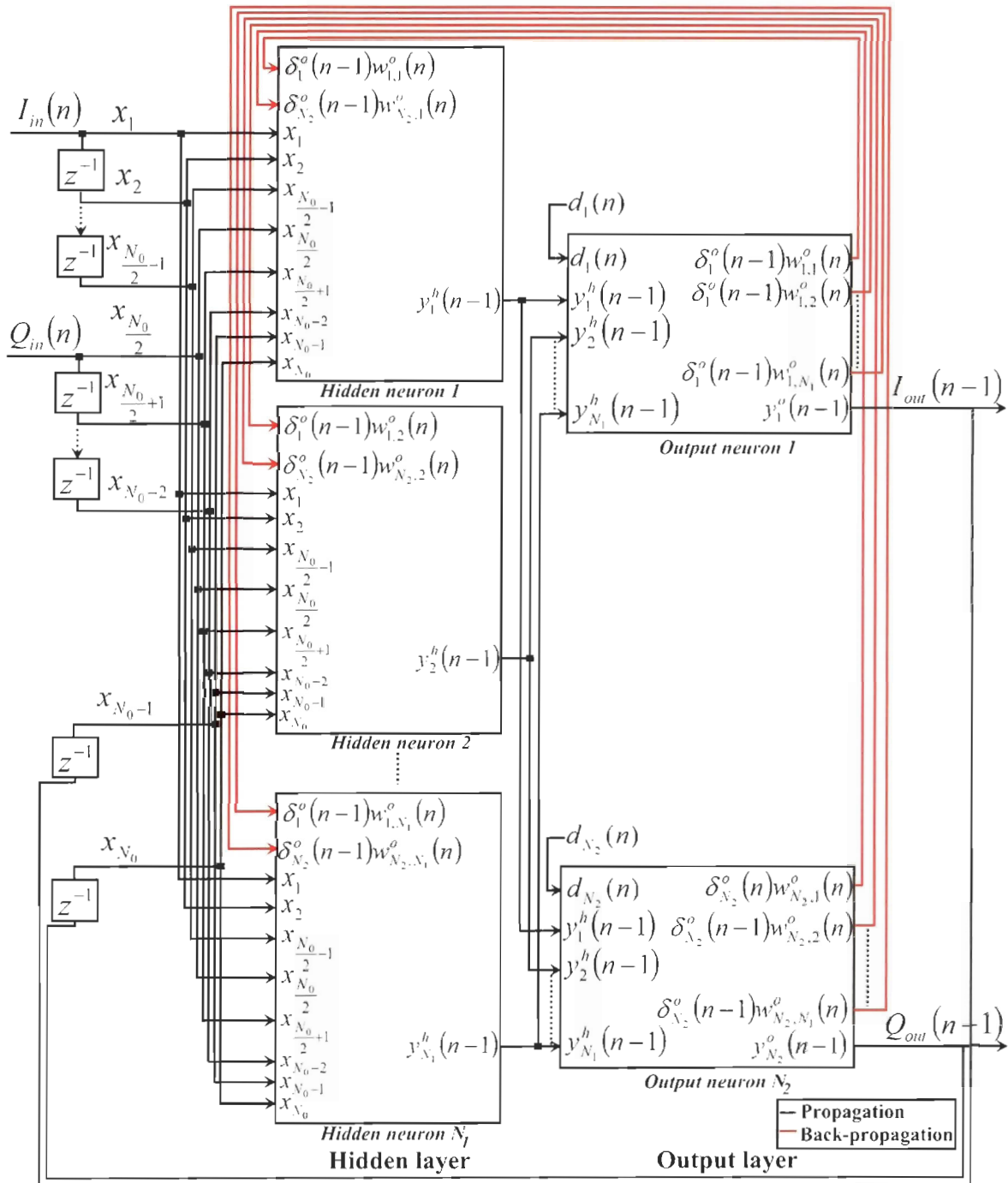


Figure 21: Architecture pseudo-conventionnelle du réseau RVRNN.

2.3.3 Architecture pseudo-conventionnelle du réseau NARX

La figure 22 représente l'architecture pseudo-conventionnelle du réseau NARX.

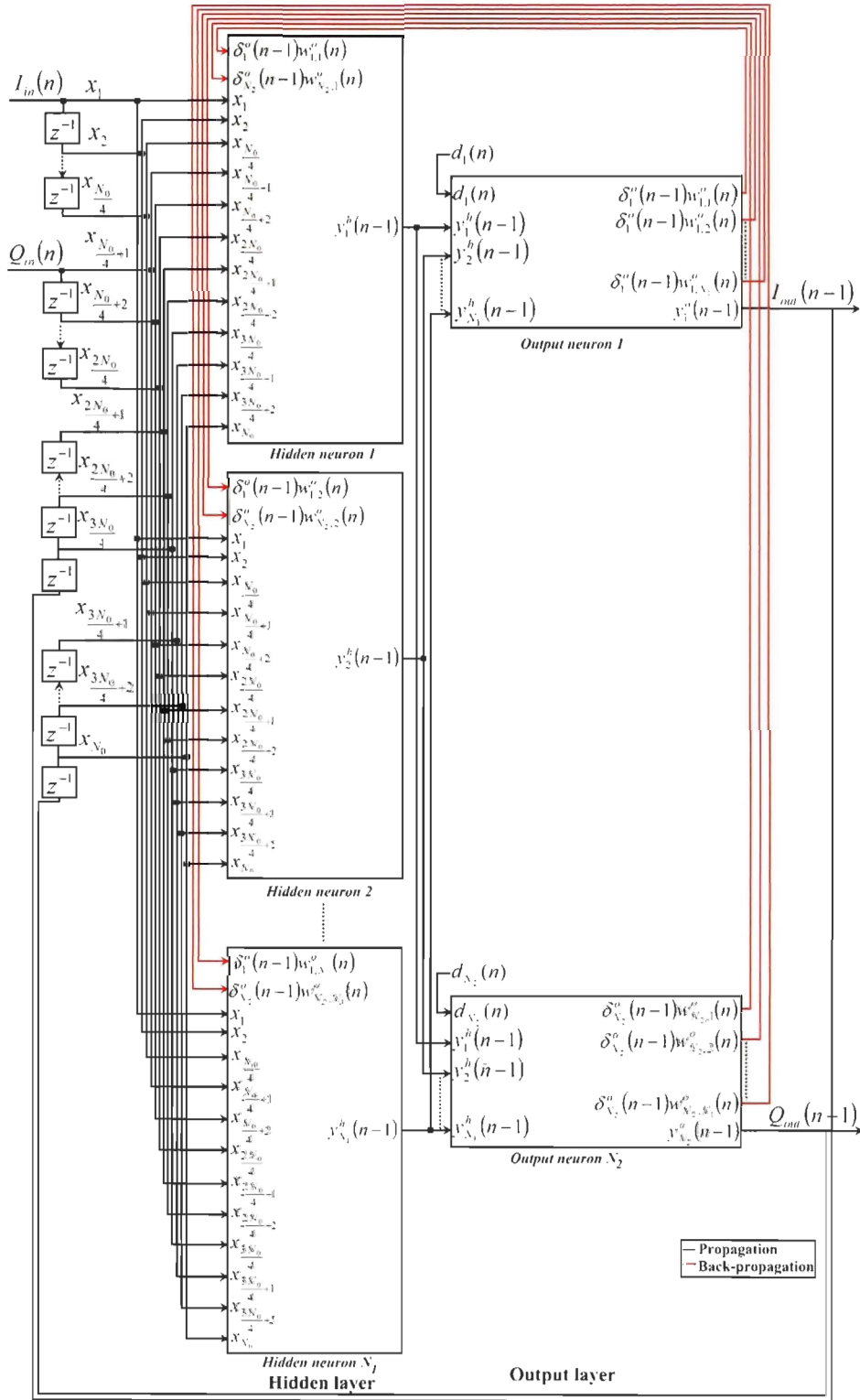


Figure 22: Architecture pseudo-conventionnelle du réseau NARX.

Le réseau NARX pseudo-conventionnel a le même principe d'allocation et de synchronisation des retards que le réseau RVTDDN et le réseau RVRNN pseudo-conventionnels. La structure de ses neurones de la couche de sortie et de la couche cachée sont les mêmes que celles du réseau RVTDDN respectivement représentées aux figures 19 et 20.

2.4 Implantation des architectures pseudo-conventionnelles des modèles RVTDDN, RVRNN et NARX en utilisant les blocs de l'outil XSG

L'architecture de test avec modulation 16-QAM est utilisée pour valider les architectures de modélisation pseudo-conventionnelles sous l'environnement Matlab/Simulink. Les blocs de l'outil XSG sont utilisés pour la programmation des architectures de modélisation. La complexité de cette étape réside dans le paramétrage des divers blocs de l'outil XSG d'une architecture, afin d'optimiser au maximum les ressources pour faciliter son implantation sur la puce FPGA Virtex-6 de la carte ML605.

Les architectures pseudo-conventionnelles RVTDDN, RVRNN et NARX choisies pour l'implantation matérielle sont les plus optimales en terme de nombre de neurones à la couche cachée, du choix du nombre de bits de données et de la taille mémoire des ROMs (*Read Only Memory*) utilisées dans le calcul des tangentes hyperboliques. Au final, les architectures des réseaux de neurones choisies sont le réseau NARX- N_1 -pse, le réseau RVRNN- N_1 -pse et le réseau RVTDDN- N_1 -pse, avec N_1 correspondant à 6 ou 8 neurones à la couche cachée. Le nombre 6 étant la valeur optimale du nombre de neurones à la couche cachée. Le nombre d'entrées N_0 dépend de l'architecture en question. Pour le réseau NARX par exemple, $N_0 = 4m + 4$, où $m = 2$ est la profondeur mémoire choisie (voir section 2.3). Par conséquent, le réseau NARX a au total 12 entrées. En suivant le même raisonnement, la taille du vecteur d'entrée du réseau RVRNN est $N_0 = 2m + 4$ où $m = 2$, d'où vient le fait que le réseau RVRNN a un total de 8 entrées. De même, le réseau RVTDDN a 6 entrées, car $N_0 = 2m + 2$. Toutes les architectures ont 2 neurones à la couche de sortie et pse correspond à pseudo-conventionnel.

2.4.1 Implantation de l'architecture RVTDDN-6-pse

Le système de test avec modulation 16-QAM des architectures pseudo-conventionnelles à bande de base est représenté sur la figure 23.

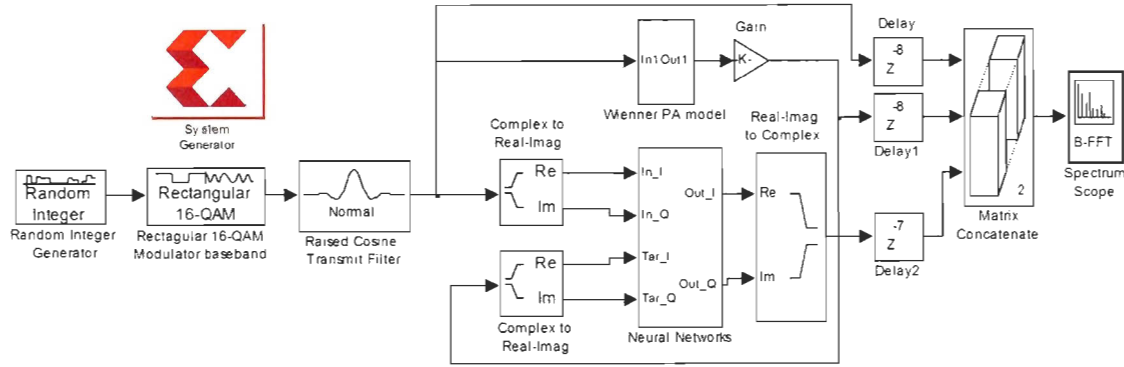


Figure 23: Système de test avec modulation 16-QAM des architectures de modélisation pseudo-conventionnelles à bande de base.

Ce diagramme est constitué d'un générateur de nombres entiers aléatoires de 0 à $M-1$, où $M=16=2^k$ est le nombre de symboles de modulation et $k = 4$ est le nombre de bits (Baudoin et al., 2007). Les entiers de 0 à 15, générés à 1 MHz, sont modulés en 16-QAM. Le symbole complexe C est (Doufana, 2009) :

$$C = I + jQ \quad (2.34)$$

avec $I, Q \in \{-3A, -A, A, 3A\}$. La forme d'onde du signal continu est :

$$x_c(n) = \sum_{u=0}^{\infty} Ch(n - uT_s) \quad (2.35)$$

où u correspond à l'indice du symbole. La période symbole est $T_s = 1/F_s$, où F_s est la fréquence symbole. Le signal $h(n)$ est une forme d'onde rectangulaire, non nulle, dans l'intervalle allant de 0 à T_s . Dans le bloc *Rectangular QAM Modulator Baseband*, le nombre M -aire est 16, les données en entrée sont de type entier, l'ordre de la constellation est binaire et la méthode de normalisation est basée sur la puissance moyenne. Cette dernière est 0.3 W et le type de données de sorties est double. Ayant un spectre infini, le signal passe à travers un filtre à cosinus sur-élevé pour limiter son spectre à une bande de transmission limitée (Doufana, 2009). La réponse impulsionnelle $h_g(n)$ du filtre à cosinus sur-élevé normal utilisée est :

$$h_g(n) = \frac{\sin(\frac{\pi n}{T_s}) \cos(\frac{\pi \beta n}{T_s})}{\frac{\pi n}{T_s} \left(1 - \frac{4\beta^2 n^2}{T_s^2}\right)} \quad (2.36)$$

Le paramètre β est le facteur d'adoucissement (Rolloff) et est compris entre 0 et 1. Le filtre à cosinus sur-élevé a un retard de groupe de 4, un facteur d'adoucissement de 0.35, un facteur de sur-échantillonnage de 8 et le gain du filtre est normalisé. Un retard dans les blocs de l'outil XSG est équivalent à $0.125 \mu s$. La largeur de bande B est (Doufana, 2009) :

$$B = \frac{1}{T_s}(1 + \beta) \quad (2.37)$$

L'efficacité spectrale Γ est définie par :

$$\Gamma = \frac{k}{1 + \beta} \quad (2.38)$$

où $k = \log_2(M)$ est le nombre de bits par symbole. $\beta = 0.35$, donc $\Gamma = 2.96$ bits/s/Hz. La largeur de bande est donc $B = 1.35$ MHz (voir l'équation 2.37). Le bloc *Wiener PA Model* est détaillé dans la section 1.2.2. Le gain du modèle de référence de Wiener est d'environ 2.77. Pour le compenser, un gain de 0.36 est inséré via le bloc *Gain* en cascade avec le modèle de référence. Le signal à la sortie du bloc *neural networks* est retardé de 7, car dans ce bloc, le traitement d'un échantillon de l'entrée à la sortie introduit un retard, le sur-échantillonnage étant de 8. Donc, la sortie du bloc *neural networks* de modulation est retardée de 7, plus 1 retard dans le bloc, cela donne un total de 8. Ainsi, les sorties du PA et l'entrée sont retardées de 8. La période d'un retard est $0.125 \mu s$, car après le sur-échantillonnage de 8 au niveau du filtre à cosinus sur-élevé, la fréquence est passée de 1 MHz à 8 MHz. Le spectroscopie est utilisé pour visualiser le spectre. Le réseau de neurones de modélisation est programmé avec des blocs de l'outil XSG. Le bloc *System Generator* sert de tour de contrôle des paramètres du système et de la simulation des blocs de l'outil XSG. Le bloc *System Generator* doit avoir une fréquence d'échantillonnage multiple de la fréquence d'échantillonnage des différents blocs, soit 8 MHz. La figure 24 représente l'architecture du réseau RVTDDN-6-pse. Les retards présents en entrée de la structure sont utilisés pour la profondeur mémoire.

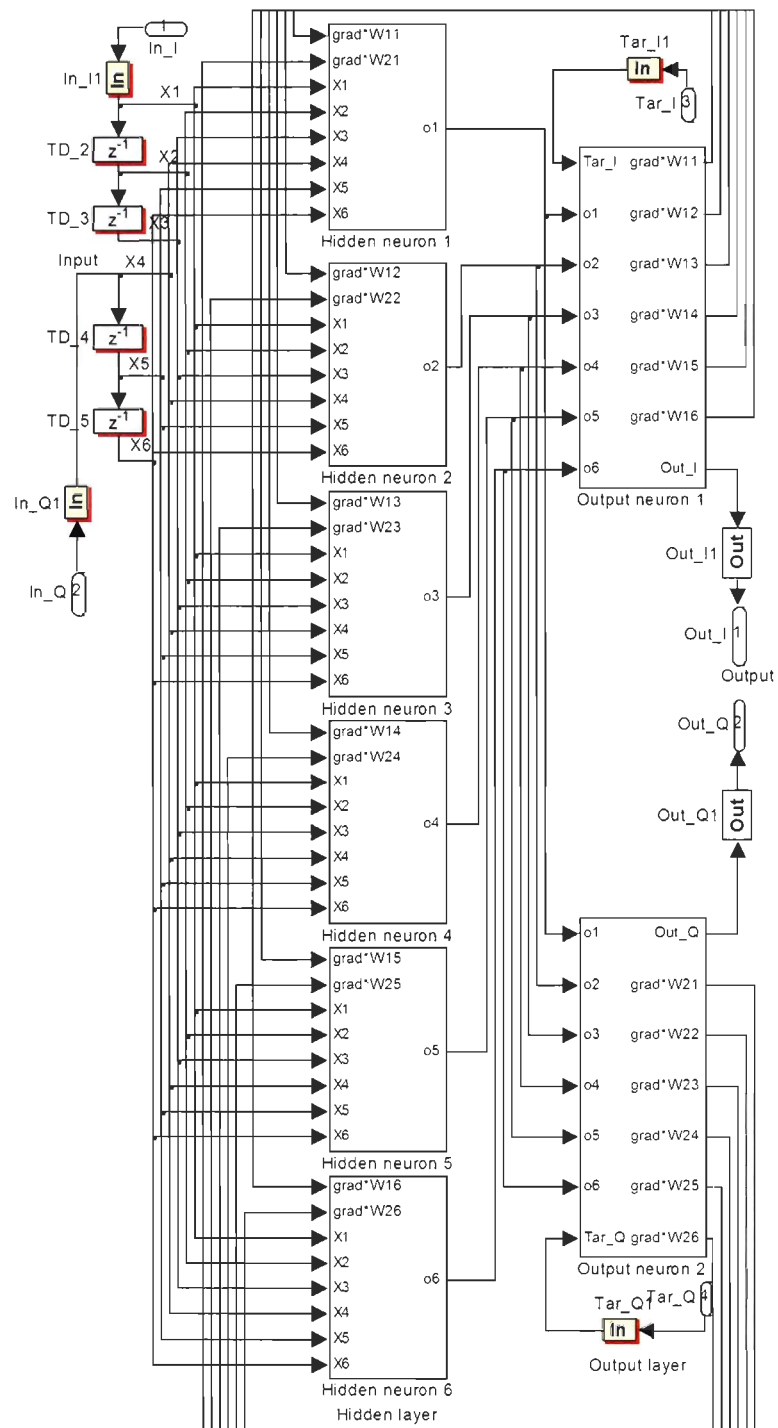


Figure 24: Architecture de modélisation RVTDDN-6-pse.

2.4.1.1 Blocs d'interface entre l'environnement Simulink et l'outil XSG

Les composants (blocs) ordinaires de l'environnement Simulink utilisent des données en format double précision (virgule flottante) alors que ceux de l'outil XSG manipulent des données en format fixe (virgule fixe). Le passage des données de l'environnement Simulink vers l'outil XSG se fait par le biais du composant *Gateway In* et l'inverse se fait par *Gateway Out* (voir figure 25). Le composant *Gateway In* est équivalent à un convertisseur analogique-numérique. Il nécessite la détermination de la fréquence d'échantillonnage et le format de quantification. Le format utilisé est FIX24RS (Bahoura et Park, 2012), qui est un format complément à 2 signé, à virgule fixe avec 24 bits dont une partie fractionnaire de 20 bits et comme option de débordement *saturate* et de quantification *round*.



Figure 25: Blocs *Gateway In* et *Gateway Out* de l'outil XSG.

2.4.1.2 Implantation d'un neurone de la couche cachée de l'architecture RVTDNN-6-pse en utilisant les blocs de l'outil XSG

La figure 26 représente l'implantation d'un neurone de la couche cachée du réseau RVTDNN-6-pse, où $N_0 = 6$, $N_1 = 6$ et $N_2 = 2$. Dans chaque neurone de la couche cachée, le registre à décalage (bloc (a) de la figure 26), à droite de 6 bits, correspond à une multiplication avec un taux d'apprentissage $\eta = 2^{-6} = 0.015625$. La valeur du taux d'apprentissage doit être bien choisie pour assurer une convergence rapide sans risque de divergence. Le multiplieur (bloc (b) de la figure 26), comme tous les autres multiplieurs de cette architecture, a comme format de données FIX24RS à sa sortie. L'implantation de la variation des poids (bloc (c) de la figure 26) est illustrée sur la figure 27.

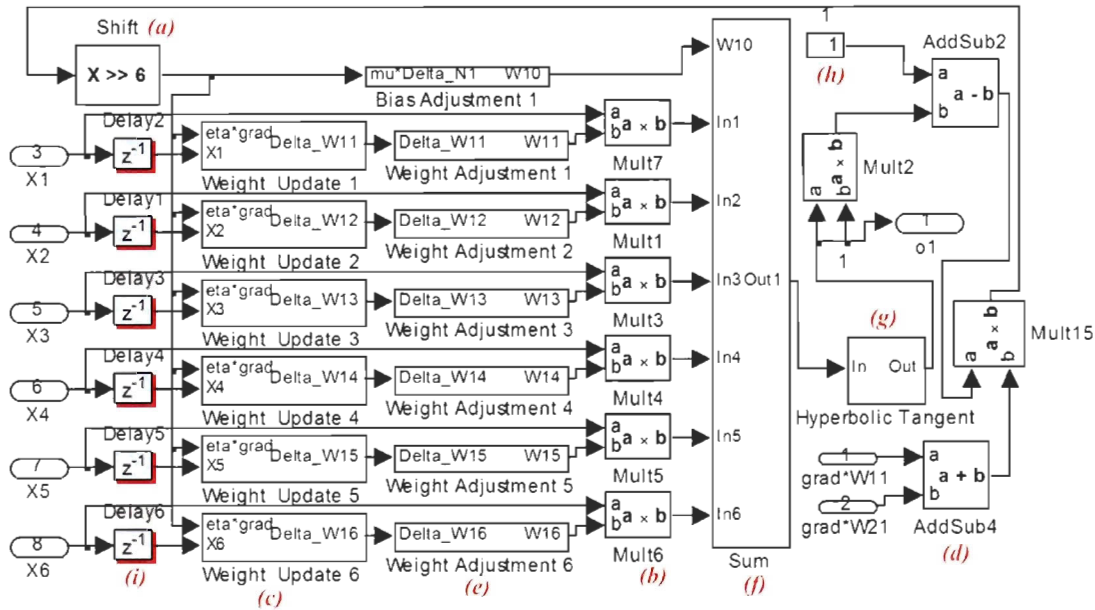


Figure 26: Implantation d'un neurone de la couche cachée de l'architecture RVTDNN-6-pse en utilisant les blocs de l'outil XSG.

L'unique multiplieur de ce bloc a les mêmes caractéristiques que le bloc (b) de la figure 26.

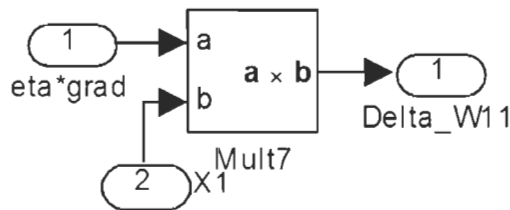


Figure 27: Implantation de la variation des poids (équation 2.29) des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.

Les données en sortie de l'additionneur (bloc (d) de la figure 26) ont comme format de données FIX24RS. L'ajustement des biais et des poids (bloc (e) de la figure 26) est représenté dans la figure 28. L'unique additionneur de ce bloc a les mêmes caractéristiques que le bloc (d) de la figure 26. Le registre à décalage de la figure 28 est utilisé pour initialiser aléatoirement les biais et les poids avec la fonction *Randn* du logiciel Matlab et ainsi obtenir une suite

unique suivant la valeur d'initialisation fixée dans l'option *InitFcn* de *Callback* des propriétés du logiciel Matlab. Dans cette option, *randn('state',1)* a été utilisé. Le choix d'un chiffre dans cette fonction permet d'avoir une suite de poids unique. S'il y a 1 comme chiffre, les 2 premières valeurs d'initialisation obtenues sont 0.864397318249258 et 0.094202610858281.

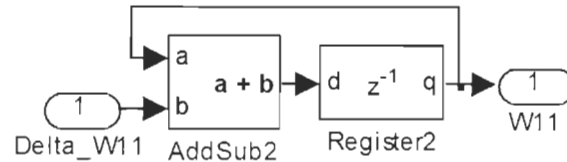


Figure 28: Implantation de l'ajustement des poids et des biais (équation 2.32) des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.

La somme (bloc (f) de la figure 26) est constituée d'un ensemble de blocs additionneurs comme illustré dans la figure 29. Le nombre d'entrées de la somme de chaque neurone à la couche cachée correspond au nombre de poids N_0 de chaque neurone plus 1, 1 correspondant au biais.

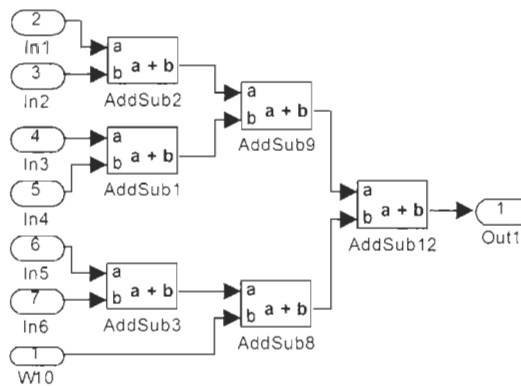


Figure 29: Implantation de la somme (équation 2.21) dans chaque neurone de la couche cachée de l'architecture pseudo-conventionnelle RVTDNN-6-pse.

Dans la fonction d'activation tangente hyperbolique (bloc (g) de la figure 26), la partie (bloc (ga) de la figure 30), a pour rôle de rendre positives toutes les valeurs réelles à son entrée avant qu'elles ne soient converties en adresses. Le registre à décalage (bloc (gb1) de la figure 30) à gauche de 12 bits est utilisé pour convertir l'entrée en adresse. La valeur de

décalage du registre correspond à une multiplication par $(2^M/A) = 2^{12}$, où $M = 15$ est le nombre de bits du bus d'adresse et $A=8$ est l'amplitude maximale de l'entrée.

Ce registre à décalage fournit une sortie non signée de M bits et sans partie fractionnaire, car l'adressage de la ROM (bloc $(gb2)$ de la figure 30) se fait avec des entiers positifs. Donc, l'adressage de cette ROM $(gb2)$ va de 0 à 4095, soit 4096 adresses. Les valeurs initiales de la ROM sont $tansig((0 : 2^M - 1)(A/2^M))$ où $tansig(\cdot)$ est la fonction tangente hyperbolique du logiciel Matlab. Les adresses vont par conséquent de 0 à $(2^M) - 1$ et sont divisées par $(2^M)/A$, car la valeur a été multipliée par $(2^M)/A$ dans le registre à décalage qui la précède. À cause de l'asymétrie de la tangente hyperbolique, les arguments des entrées négatives sont tout simplement les opposés de ceux des entrées positives. Dans cette ROM, l'option *Block RAM* est choisie pour utiliser les blocs RAM disponibles en grand nombre sur la puce FPGA. Cela implique un retard supplémentaire dans la ROM (bloc $(gb2)$ de la figure 30), qui nécessite l'ajout du retard de synchronisation à l'entrée *sel* du multiplexeur (bloc $(gc2)$ de la figure 30). Dans la partie (gc) de la figure 30, les nombres qui étaient négatifs à l'entrée redeviennent négatifs et ceux positifs le restent par l'utilisation du multiplexeur et de *Negate*, avec la même logique que la partie (ga) .

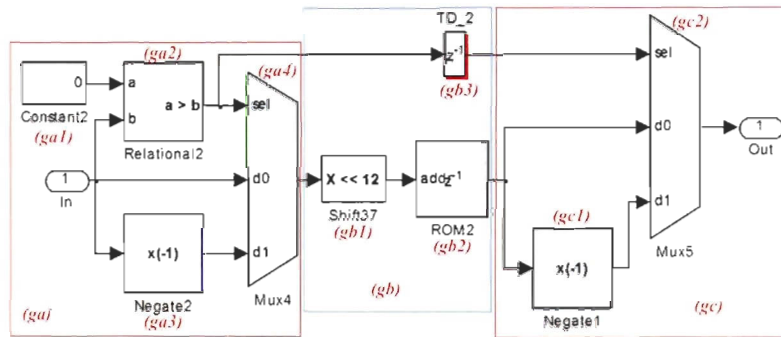


Figure 30: Implantation de la tangente hyperbolique des neurones de la couche cachée de l'architecture pseudo-conventionnelle RVTDDN-6-pse.

2.4.1.3 Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pse en utilisant les blocs de l'outil XSG

La figure 31 représente l'implantation d'un neurone de la couche de sortie de l'architecture pseudo-conventionnelle RVTDDN-6-pse. La somme, la variation des poids, la mise à jour

des poids des biais et poids sont des sous-systèmes de même constitution que dans la section 2.4.1.2. Dans cette architecture, le nombre de neurones à la couche cachée est égal au nombre d'entrées. Cela explique pourquoi la structure de la somme de chaque neurone à la couche cachée est identique à celle de la somme de chaque neurone de la couche de sortie.

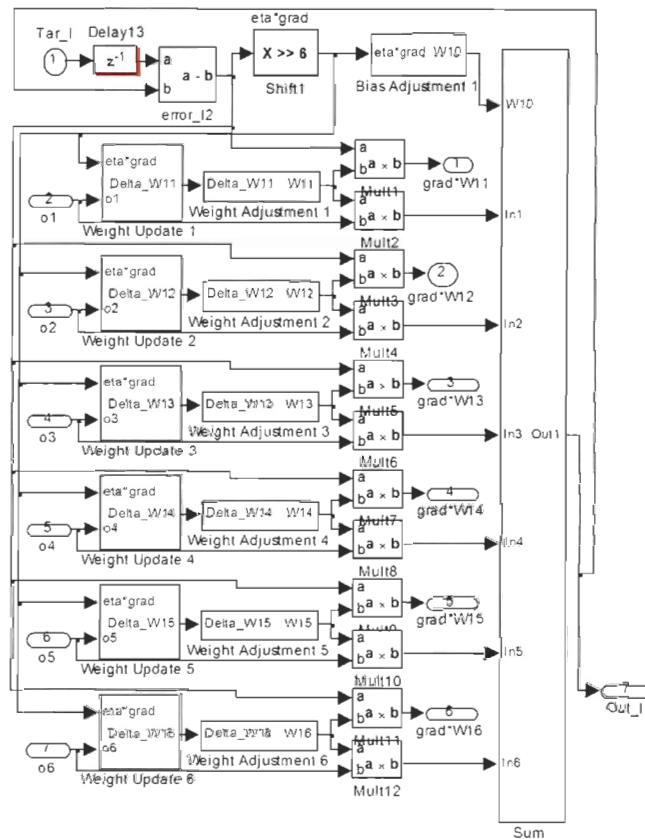


Figure 31: Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pse avec des blocs de l'outil XSG.

2.4.2 Implantation de l'architecture RVRNN-6-pse

La figure 32 représente l'implantation de l'architecture RVRNN-6-pse en utilisant les blocs de l'outil XSG.

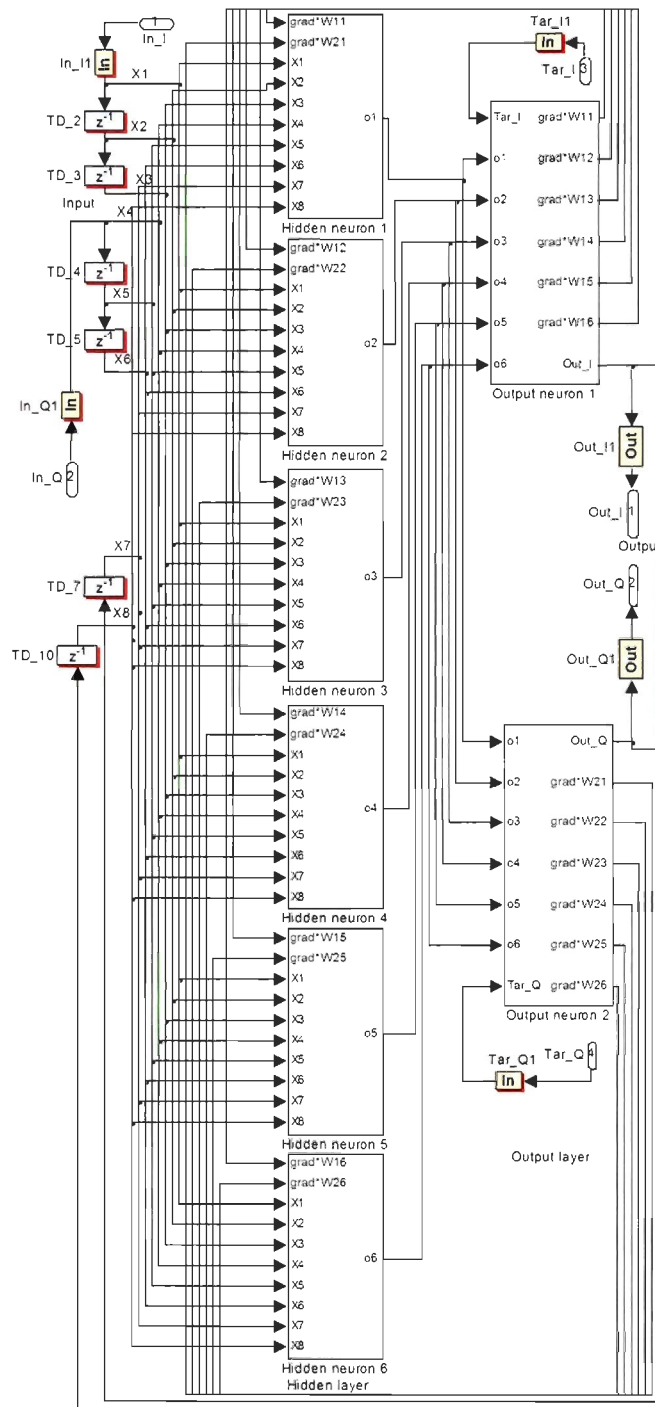


Figure 32: Implantation de l'architecture RVRNN-6-pse en utilisant les blocs de l'outil XSG.

Elle possède la même structure que celle du réseau RVTDDN-6-pse, mis à part la structure de la somme à la couche cachée qui a deux entrées supplémentaires. La dimension du vecteur d'entrée de l'architecture RVRNN-6-pse est $N_2 = 8$, car $m = 2$. La figure 33 illustre l'implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pse en utilisant les blocs de l'outil XSG.

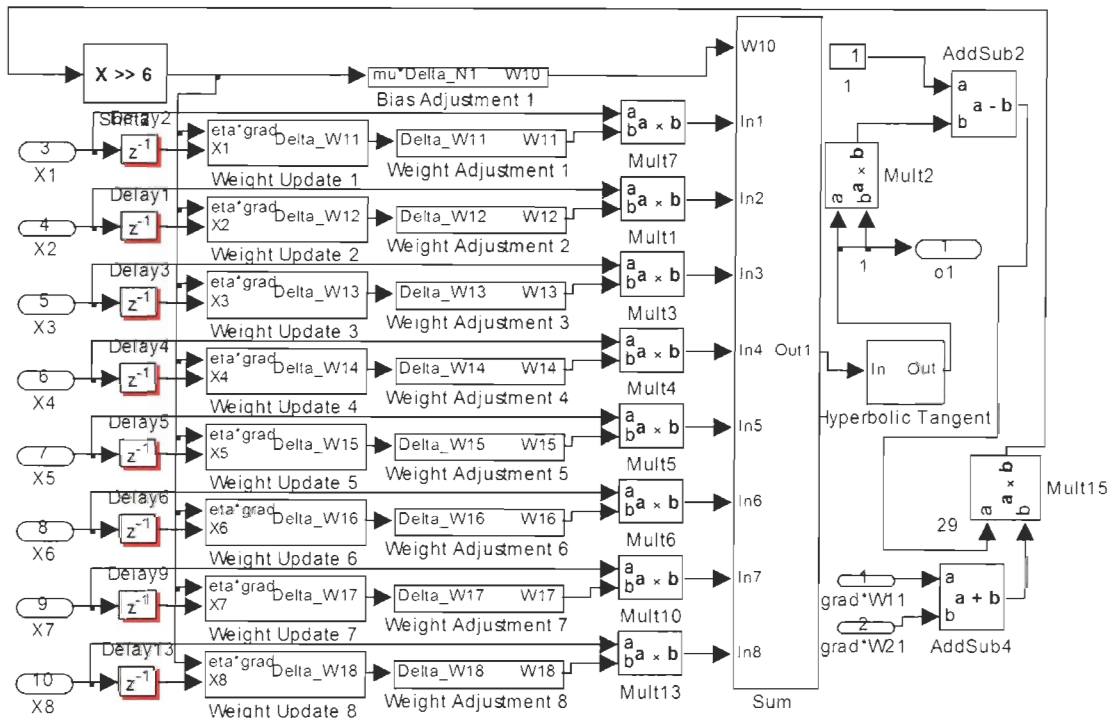


Figure 33: Implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pse avec des blocs de l'outil XSG.

L'implantation d'un neurone de la couche de sortie de l'architecture RVRNN-6-pse est la même que celle de l'architecture RVTDDN-6-pse décrite dans la figure 31.

2.4.3 Implantation de l'architecture NARX-6-pse

Tous les composants de l'architecture de modélisation NARX-6-pse ont la même structure à base de l'outil XSG que ceux de l'architecture de modélisation RVTDDN-6-pse section 2.4.1.

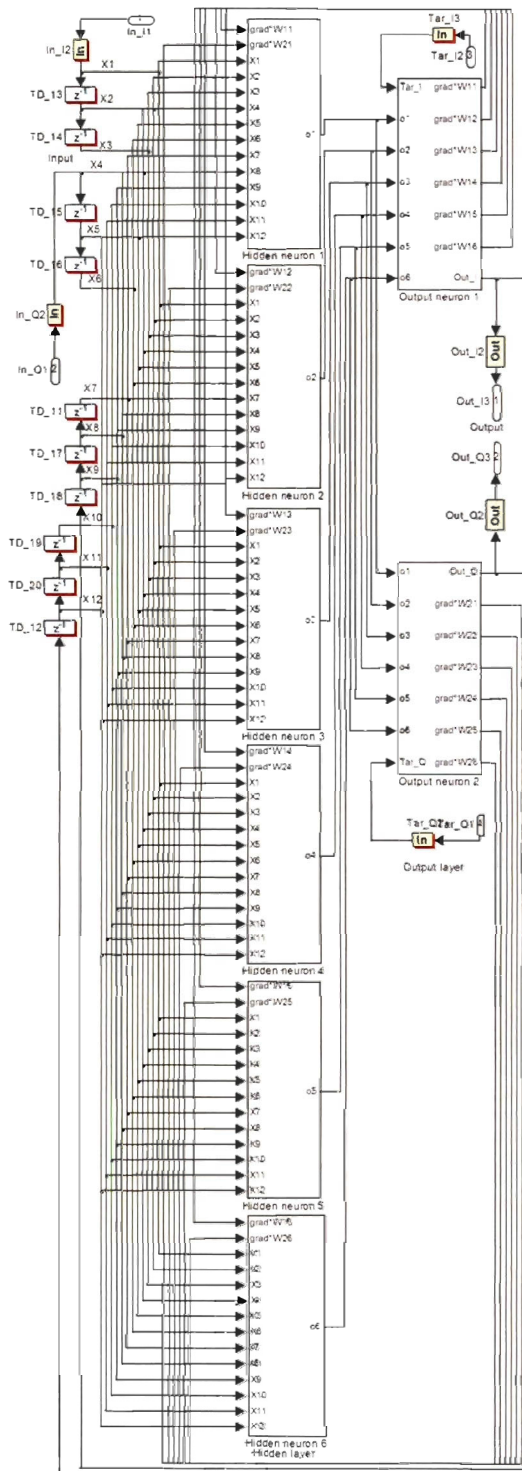


Figure 34: Implantation de l'architecture NARX-6-pse avec des blocs de l'outil XSG.

Ce qui diffère, c'est aussi la somme à la couche cachée, car le réseau NARX-6-pse a un nombre d'entrées de réaction égal au nombre d'entrées. Cela donne un total de 12 entrées, parce qu'il en a 6 de plus que l'architecture de modélisation RVTDDN-6-pse. Sur la figure 34, l'implantation de l'architecture NARX-6-pse en utilisant les blocs de l'outil XSG est représentée. La taille de son vecteur d'entrée étant égal à $4m + 4$, le réseau NARX-6-pse a un total de 12 entrées, avec $m = 2$. L'architecture de modélisation NARX-6-pse a $N_0 = 12$, $N_1 = 6$ et $N_2 = 2$. La figure 35 représente l'implantation d'un neurone de la couche cachée de l'architecture NARX-6-pse.

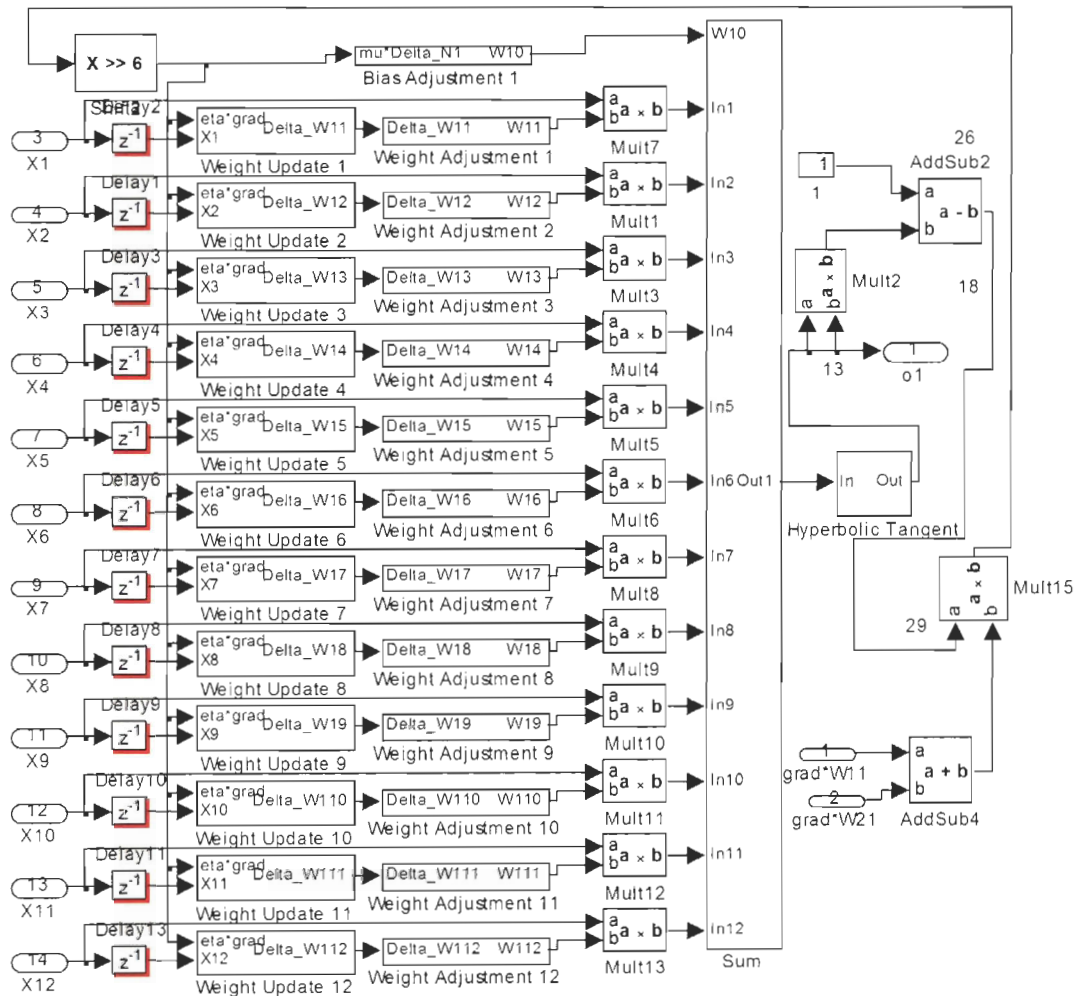


Figure 35: Implantation d'un neurone de la couche cachée de l'architecture NARX-6-pse avec des blocs de l'outil XSG.

L'implantation d'un neurone de la couche de sortie de l'architecture NARX-6-pse est la même que celle de l'architecture RVTDNN-6-pse décrite dans la Figure 31. L'implantation des architectures RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse en utilisant les blocs de l'outil XSG, consiste à ajouter deux neurones respectivement à la couche cachée aux architectures RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse. Pour augmenter la fréquence maximale d'opération, le pipeline de ces différentes architectures est proposé.

2.5 Architectures pipelinées des réseaux RVTDNN, RVRNN et NARX pour la modélisation des PAs

Le pipeline des architectures de modélisation RVTDNN, RVRNN et NARX permet d'augmenter considérablement la fréquence de traitement d'un échantillon, soit la fréquence maximale d'opération (MOF). Le principal inconvénient de l'architecture pseudo-conventionnelle est que sa fréquence maximale d'opération est insuffisante par rapport à la fréquence minimale d'horloge disponible dans la puce FPGA de la carte ML605 pour l'interface *JTAG Hardware co-simulation* (voir section 2.7). Le chemin critique est le temps de calcul le plus long, parmi tous les chemins, ne contenant pas de retards (Bahoura et Ezzaidi, 2011). Le chemin critique affecte la fréquence maximale d'opération ou vitesse de traitement d'une architecture donnée, parce qu'il représente le temps minimal exigé avant le traitement d'un nouvel échantillon.

La fréquence maximale d'opération devrait être au moins égale à la fréquence minimale de co-simulation de la puce FPGA Virtex-6 de la carte ML605 pour que le système soit capable de traiter le signal en temps-réel. Ainsi la réduction du chemin critique peut se faire par pipeline. L'opération de pipeline consiste à insérer des retards et de les redistribuer en tenant compte de la synchronisation des différents trajets des architectures (Bahoura et Park, 2011b). Cependant, cette approche a un défaut. Elle entraîne des retards à la sortie du système. La fréquence MOF étant inversement proportionnelle au chemin critique, l'améliorer consiste à diminuer la durée du chemin critique. L'implantation des architectures pipelinées est basée sur celle des architectures pseudo-conventionnelles. Il est à noter que le chemin critique est fortement pénalisé par le temps des opérations de multiplication et que le nombre de multiplieurs utilisés dans chacun des blocs de multiplication dépend du nombre de bits des données. Ainsi, la réduction du chemin critique consiste essentiellement à insérer un nombre suffisant (optimal) de délais dans chacun des blocs de multiplication. Pour la quantification à

24 bits. il suffit d'insérer 4 retards par bloc de multiplieur au lieu de 7 pour une quantification à 36 bits (Bahoura et Park, 2012). Le pipeline d'une architecture se fait en 4 grandes étapes.

1. L'optimisation des blocs de multiplication, en terme de retards insérés, en réduisant le nombre de bits de données.
2. L'ajout des retards avant et après les autres blocs comme ceux d'addition.
3. La synchronisation de tous les chemins de propagation.
4. La synchronisation des chemins de rétro-propagation du même niveau.

2.5.1 Architecture pipelinée du réseau RVDNN

Dans l'architecture pipelinée du réseau RVDNN, la distribution des retards dans le système est optimisée. La synchronisation des opérations se fait pour équilibrer les retards devant chaque opérateur (voir section 2.5). Les équations de l'architecture de modélisation RVDNN avec pipeline sont influencées par les retards ajoutés. Les délais insérés par le pipeline ont été distribués dans le système de telle sorte que les retards aux sorties des réseaux soient toujours multiples de 8, car le signal modulé a été sur-échantillonné de 8. Cela explique pourquoi, dans toutes les architectures de modélisation avec pipeline, il y a 24 retards aux sorties. excepté le réseau RVDNN-6-pip, qui en a 16. Le vecteur d'entrée du signal à bande de base de l'architecture pipelinée du réseau RVDNN est identique à celui de son architecture pseudo-conventionnelle décrite à la section 2.2.1.

2.5.1.1 Propagation des signaux dans l'architecture pipelinée du réseau RVDNN-6-pip

La figure 36 représente l'architecture pipelinée du réseau RVDNN-6-pip.

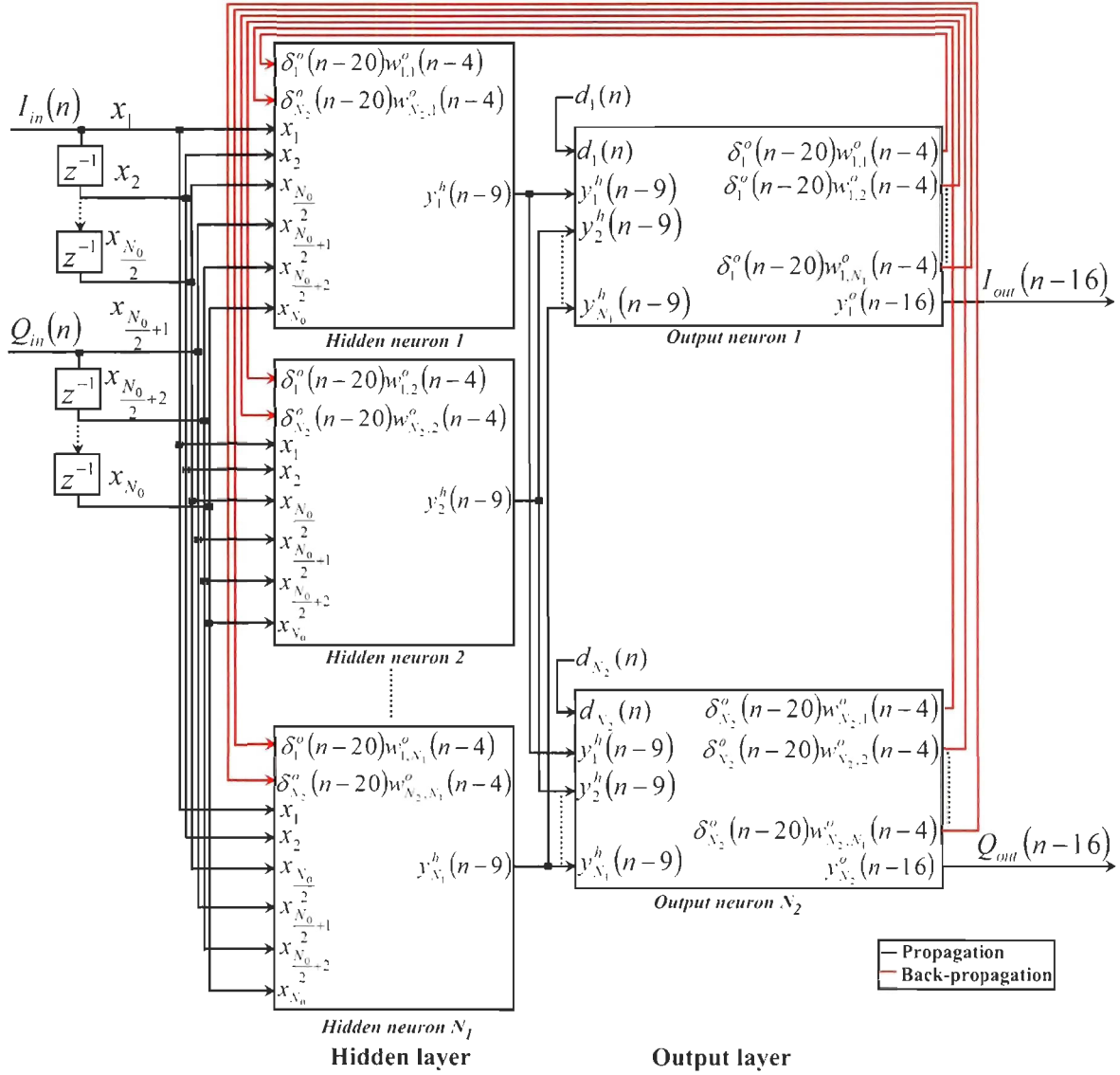


Figure 36: Architecture pipelinée du réseau RVTDDN-6-pip.

Comme indiqué sur la figure 36, les sorties des neurones de la couche cachée sont retardées de 9 délais.

$$y_j^h(n-9) = \varphi_h(v_j^h(n-9)) \quad j = 1, \dots, N_1 \quad (2.39)$$

avec

$$v_j^h(n-9) = \sum_{i=1}^{N_0} w_{j,i}^h(n-9)x_i(n-9) + w_{j,0}^h(n-9) \quad i = 1, \dots, N_0 \quad (2.40)$$

où $w_{j,0}^h(n-9)$ et $w_{j,i}^h(n-9)$ représentent respectivement le biais et le poids de sa connexion à la $j^{\text{ième}}$ entrée. La sortie du $k^{\text{ième}}$ neurone de la couche de sortie est retardée de 16 délais.

$$y_k^o(n-16) = \varphi_o(v_k^o(n-16)) \quad k = 1, \dots, N_2 \quad (2.41)$$

avec

$$v_k^o(n-16) = \sum_{j=1}^{N_1} w_{k,j}^o(n-7)y_j^h(n-16) + w_{k,0}^o(n-7) \quad (2.42)$$

où $w_{k,0}^o(n-7)$ et $w_{k,j}^o(n-7)$ représentent respectivement le biais et le poids de sa connexion au $j^{\text{ième}}$ neurone de la couche cachée retardée de 7 délais.

2.5.1.2 Rétro-propagation de l'erreur dans l'architecture RVTDDN-6-pip

Le signal d'erreur retardé de 16 est la différence entre la sortie du modèle de PA retardée de 16 délais et la sortie réelle retardée de 16 du réseau de neurones :

$$e_k(n-16) = d_k(n-16) - y_k(n-16) \quad (2.43)$$

2.5.1.3 Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDDN-6-pip

La structure des neurones de la couche de sortie de l'architecture pipelinée à 6 neurones à la couche cachée RVTDDN-6-pip est représentée sur la figure 37. Les variations des poids $w_{k,j}^o(n)$ des neurones de la couche de sortie sont définies par :

$$\Delta w_{k,j}^o(n) = \eta \delta_k^o(n-20)y_j^h(n-20) \quad (2.44)$$

où $\eta = 0.015625$ est le taux d'apprentissage. Le gradient local est défini par :

$$\delta_k^o(n-20) = e_k(n-20)\varphi_o'(v_k^o(n-20)) \quad (2.45)$$

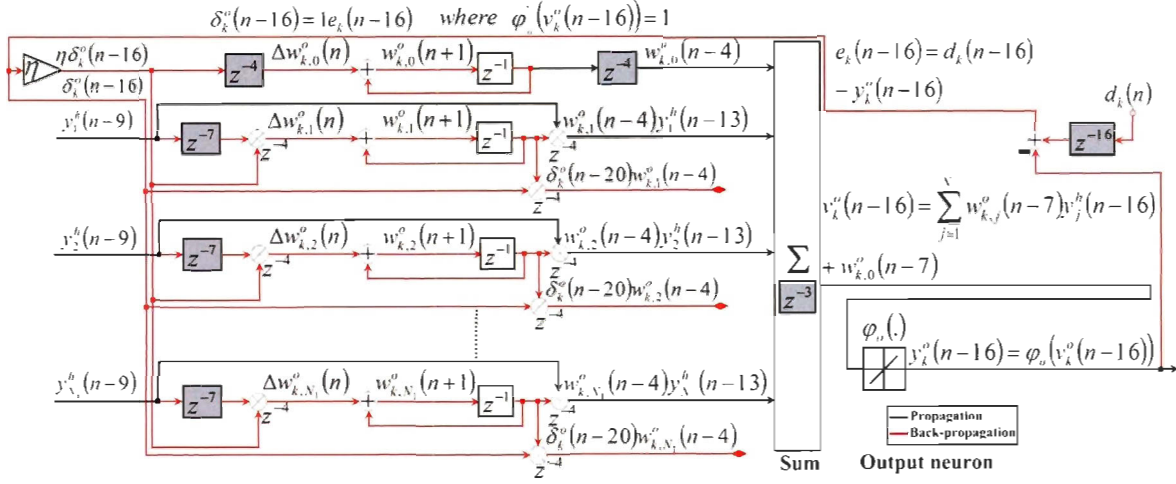


Figure 37: Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDDN-6-pip.

où $\varphi'_o(v_k^o(n-20)) = 1$, car la fonction d'activation à la couche de sortie est linéaire.

La mise à jour des poids est définie par :

$$w_{k,j}^o(n+1) = w_{k,j}^o(n) + \Delta w_{k,j}^o(n) \quad (2.46)$$

La correction $\Delta w_{k,0}^o(n)$ appliquée au biais est un cas particulier de la correction des poids $\Delta w_{k,j}^o(n)$ et est définie par :

$$\Delta w_{k,0}^o(n) = \eta \delta_k^o(n-20) \quad (2.47)$$

2.5.1.4 Structure des neurones de la couche cachée de l'architecture pipelinée RVTDDN-6-pip

La structure des neurones de la couche cachée de l'architecture pipelinée RVTDDN-6-pip est représentée sur la figure 38. La variation des poids $w_{j,i}^h(n)$ de la couche cachée est définie par :

$$\Delta w_{j,i}^h(n) = \eta \delta_j^h(n-29) x_i(n-29) \quad (2.48)$$

où le gradient local est :

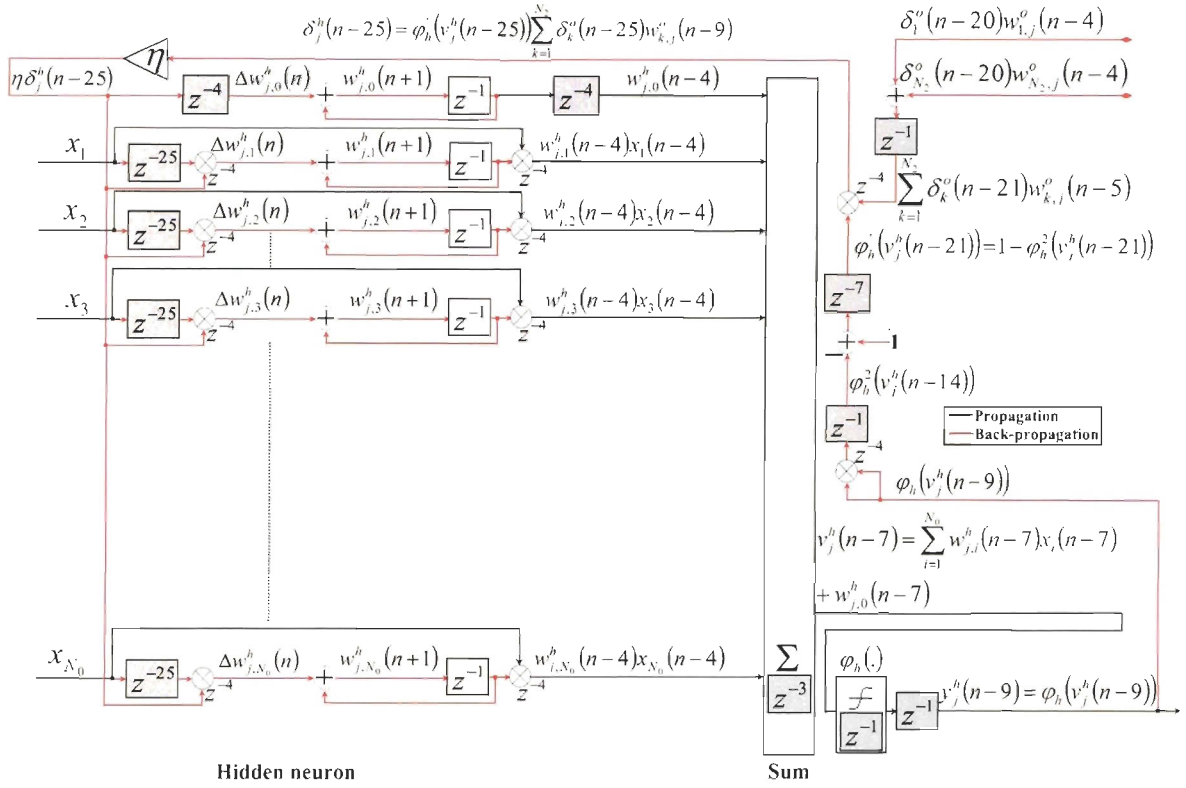


Figure 38: Structure des neurones de la couche cachée de l'architecture pipelinée RVTDDN-6-pip.

$$\delta_j^h(n-29) = \varphi'_h(v_j^h(n-29)) \sum_{k=1}^{N_2} \delta_k^o(n-29) w_{k,j}^o(n-13) \quad (2.49)$$

où

$$\varphi'_h(v_j^h(n-29)) = 1 - \varphi_h^2(v_j^h(n-29)) \quad (2.50)$$

car une fonction tangente hyperbolique est utilisée à la couche cachée. La mise à jour des poids à la couche cachée est :

$$w_{j,i}^h(n+1) = w_{j,i}^h(n) + \Delta w_{j,i}^h(n) \quad (2.51)$$

La correction $\Delta w_{j,i}^h(n)$ appliquée aux biais est :

$$\Delta w_{j,0}^h(n) = \eta \delta_j^h(n-29) \quad (2.52)$$

2.5.1.5 Propagation des signaux dans l'architecture pipelinée du réseau RVTDNN-8-pip

La figure 39 illustre l'architecture pipelinée RVTDNN-8-pip.

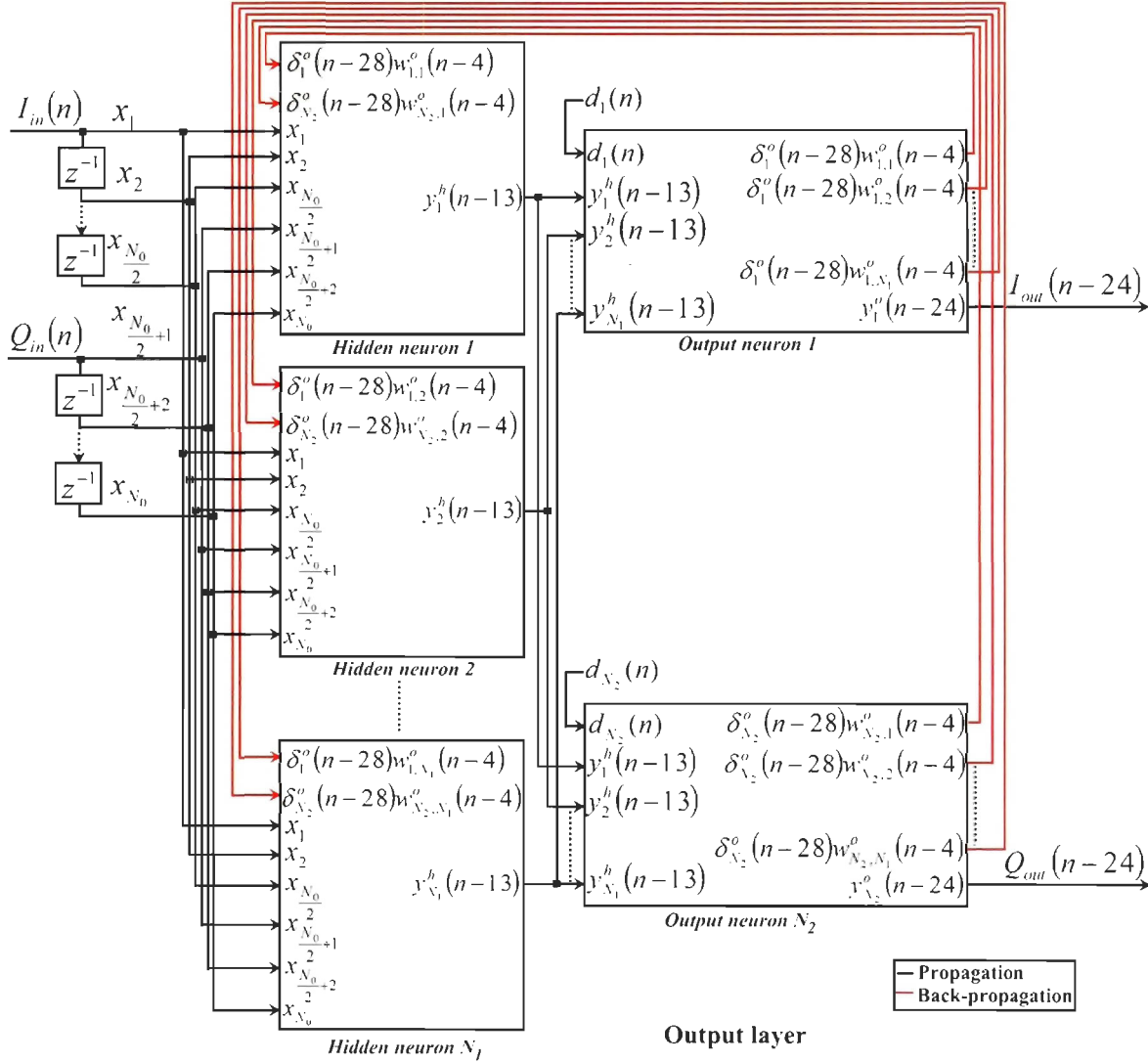


Figure 39: Architecture pipelinée du réseau RVTDNN-8-pip.

Comme indiqué sur la figure 39, les sorties des neurones de la couche cachée sont retardées de 13 délais.

$$y_j^h(n-13) = \varphi_h(v_j^h(n-13)) \quad j = 1, \dots, N_1 \quad (2.53)$$

avec

$$v_j^h(n-13) = \sum_{i=1}^{N_0} w_{j,i}^h(n-13)x_i(n-13) + w_{j,0}^h(n-13) \quad i = 1, \dots, N_0 \quad (2.54)$$

où $w_{j,0}^h(n-13)$ et $w_{j,i}^h(n-13)$ représentent respectivement le biais et le poids de sa connexion à la $i^{\text{ième}}$ entrée. La sortie du $k^{\text{ième}}$ neurone de la couche de sortie est retardée de 24 délais.

$$y_k^o(n-24) = \varphi_o(v_k^o(n-24)) \quad k = 1, \dots, N_2 \quad (2.55)$$

avec

$$v_k^o(n-24) = \sum_{j=1}^{N_1} w_{k,j}^o(n-11)y_j^h(n-24) + w_{k,0}^o(n-11) \quad (2.56)$$

où $w_{k,0}^o(n-11)$ et $w_{k,j}^o(n-11)$ représentent respectivement le biais et poids de la connexion au $j^{\text{ième}}$ neurone de la couche cachée retardé de 11 délais.

2.5.1.6 Rétro-propagation de l'erreur de l'architecture RVTDDN-8-pip

Le calcul du signal d'erreur pour la mise à jour des poids de l'architecture pipelinée du réseau RVTDDN-8-pip se fait également avec l'algorithme de rétro-propagation (Haykin, 2009). Le signal d'erreur retardé de 24 délais $e_k(n-24)$ est la différence entre la sortie du modèle de PA avec 24 retards et la sortie actuelle du réseau NARX retardée de 24 délais (Bahoura et Ezzaidi, 2011) comme indiqué dans la figure 40.

$$e_k(n-24) = d_k(n-24) - y_k(n-24) \quad (2.57)$$

2.5.1.7 Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDDN-8-pip

Cependant, la structure des neurones de la couche de sortie de l'architecture de modélisation RVTDDN-8-pip est différente de celle du réseau RVTDDN-6-pip avec pipeline, car le nombre de retards à la sortie est 24 au lieu de 16. La structure des neurones de la couche de sortie de

l'architecture pipelinée du réseau RVTDDN-8-pip est représentée sur la figure 40.

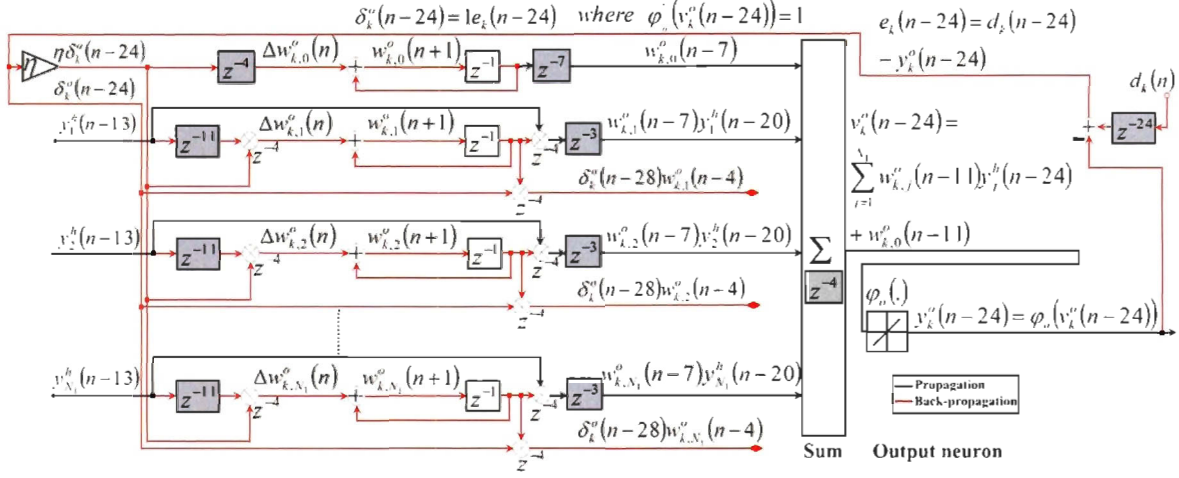


Figure 40: Structure des neurones de la couche de sortie de l'architecture pipelinée RVTDDN-8-pip.

Les variations des poids synaptiques $w_{k,j}^o(n)$ des neurones de la couche de sortie sont définies par :

$$\Delta w_{k,j}^o(n) = \eta \delta_k^o(n-28) y_j^h(n-28) \quad (2.58)$$

où $\eta = 0.015625$ est le taux d'apprentissage. Le gradient local est défini par :

$$\delta_k^o(n-28) = e_k(n-28) \varphi_o'(v_k^o(n-28)) \quad (2.59)$$

où $\varphi_o'(v_k^o(n-28)) = 1$, car la fonction d'activation $\varphi_o(\cdot)$ est linéaire.

La mise à jour des poids est définie par :

$$w_{k,j}^o(n+1) = w_{k,j}^o(n) + \Delta w_{k,j}^o(n) \quad (2.60)$$

La variation $\Delta w_{k,0}^o(n)$ des biais est un cas particulier des variations des poids $\Delta w_{k,j}^o(n)$ et est définie par :

$$\Delta w_{k,0}^o(n) = \eta \delta_k^o(n-28) \quad (2.61)$$

2.5.1.8 Structure des neurones de la couche cachée de l'architecture pipelinée RVTDNN-8-pip

La structure d'un neurone de la couche cachée de l'architecture pipelinée du réseau RVTDNN-8-pip est représentée à la figure 41.

La variation des poids des neurones de la couche cachée est donnée par :

$$\Delta w_{j,i}^h(n) = \eta \delta_j^h(n-37) x_i(n-37) \quad (2.62)$$

où le gradient local est défini par :

$$\delta_j^h(n-37) = \varphi'_h(v_j^h(n-37)) \sum_{k=1}^{N_2} \delta_k^o(n-37) w_{k,j}^o(n-13) \quad (2.63)$$

pour la tangente hyperbolique

$$\varphi'_h(v_j^h(n-37)) = 1 - \varphi_h^2(v_j^h(n-37)) \quad (2.64)$$

La mise à jour des poids des neurones à la couche cachée est :

$$w_{j,i}^h(n+1) = w_{j,i}^h(n) + \Delta w_{j,i}^h(n) \quad (2.65)$$

La variation des biais des neurones de cette couche est donnée par :

$$\Delta w_{j,0}^h(n) = \eta \delta_j^h(n-37) \quad (2.66)$$

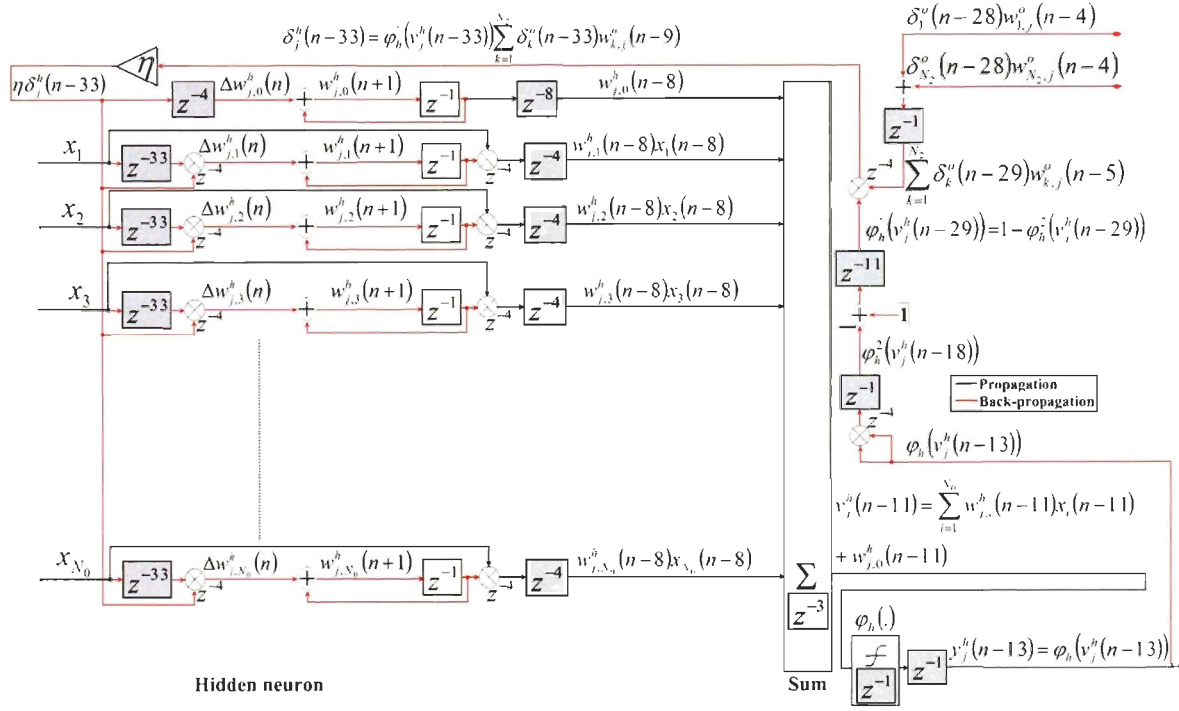


Figure 41: Structure des neurones de la couche cachée de l'architecture pipelinée RVTDDN-8-pip.

2.5.2 Architecture pipelinée du réseau RVRNN

La figure 42 illustre l'architecture pipelinée du réseau RVRNN. Cette architecture diffère de celle du réseau RVTDDN pipelinée uniquement par la taille du vecteur d'entrée qui affectera la taille du bloc somme des neurones de la couche cachée. Les équations de propagation et de rétro-propagation de l'architecture pipelinée RVRNN sont les mêmes que celles du réseau RVTDDN-8-Pip. La structure des neurones de la couche cachée de l'architecture pipelinée du réseau RVRNN dépend du nombre de neurones à cette couche, pour le réseau RVRNN-8-pip, elle est représentée par la figure 43, par contre la structure des neurones de la couche de sortie est la même que celle du réseau RVTDDN-8-pip qui est représentée par la figure 40.

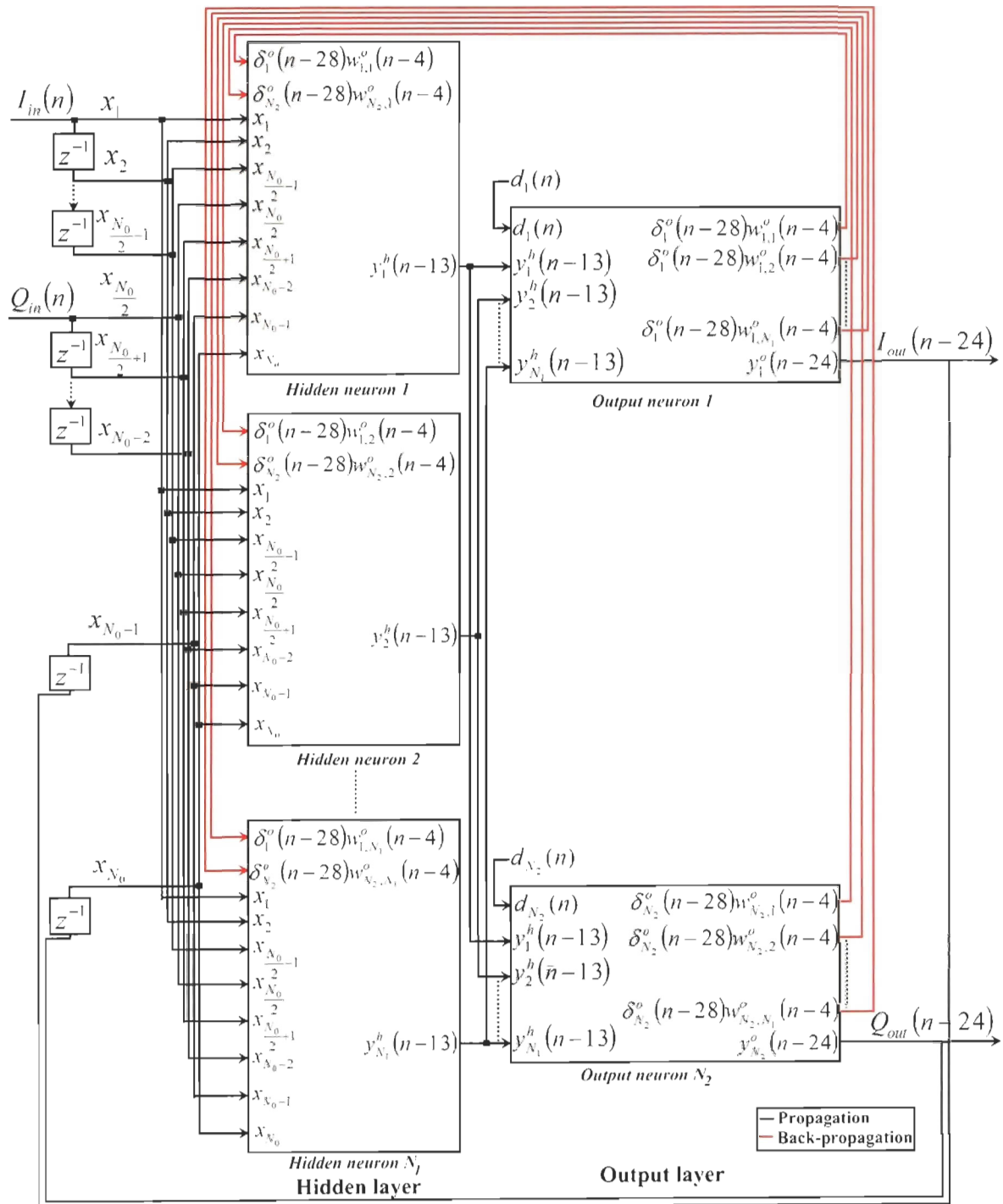


Figure 42: Architecture pipelinée du réseau RVRNN.

La structure des neurones de la couche cachée de l'architecture pipelinée du réseau RVRNN est représentée par la figure 43 :

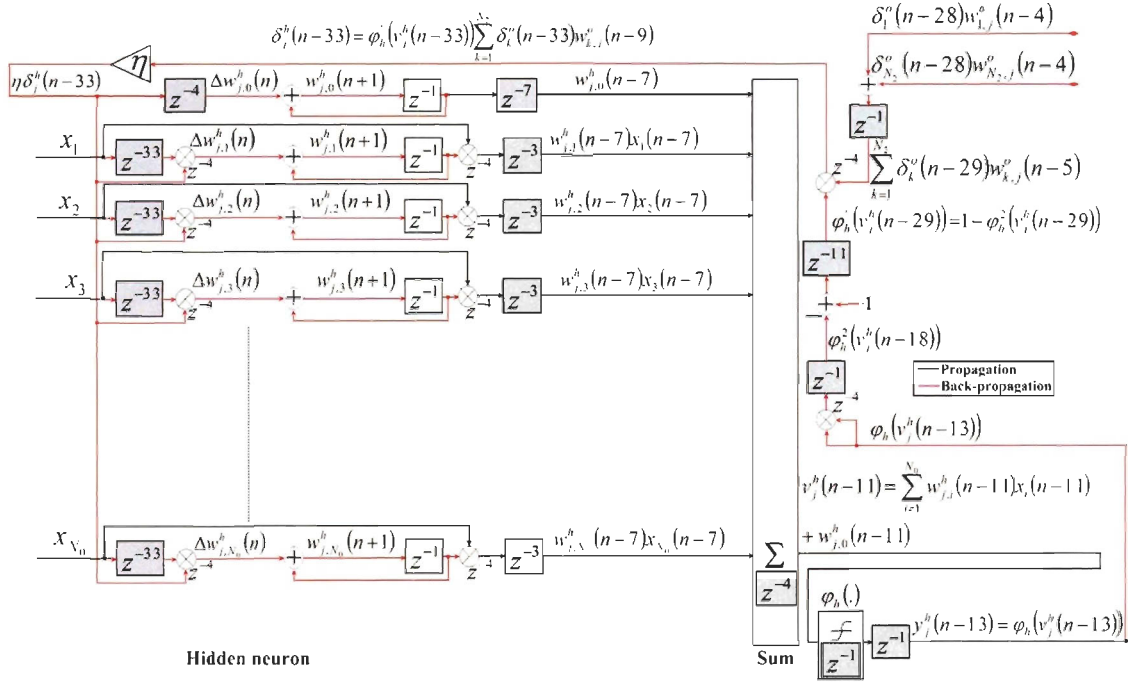


Figure 43: Structure des neurones de la couche cachée de l'architecture pipelinée RVRNN-8-pip.

La structure des neurones de la couche de sortie de l'architecture pipelinée RVRNN-6-pip est représentée par la figure 44. Par contre la structure des neurones de la couche cachée du réseau RVRNN-6-pip est la même que celle du réseau RVRNN-8-pip représentée par la figure 43.

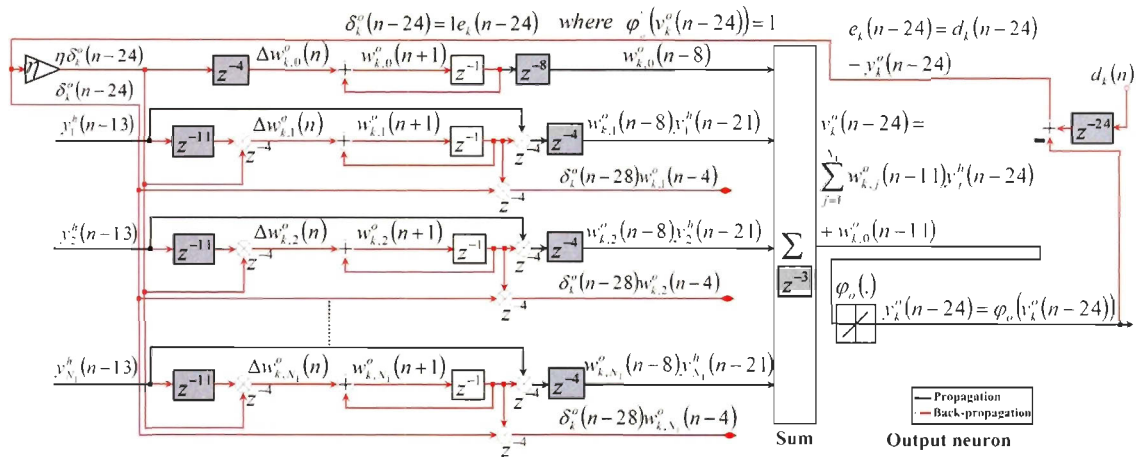


Figure 44: Structure des neurones de la couche de sortie de l'architecture pipelinée RVRNN-6-pip.

2.5.3 Architecture pipelinée du réseau NARX

La figure 45 illustre l'architecture pipelinée du réseau NARX.

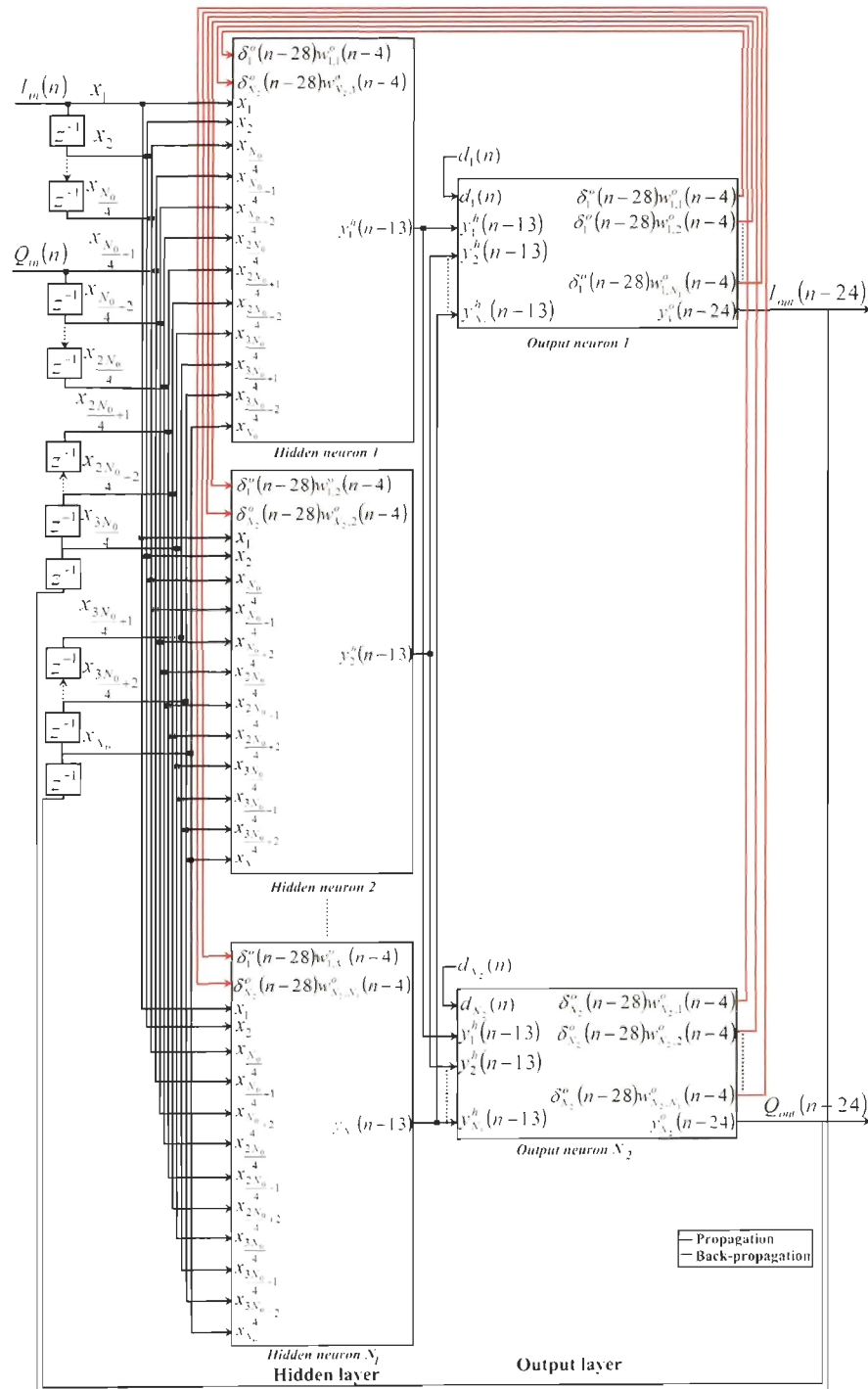


Figure 45: Architecture pipelinée du réseau NARX.

La structure des neurones de la couche de sortie du réseau NARX-8-pip est la même que celle du réseau RVTDDN-8-pip qui est représentée par la figure 40. Par contre la structure des neurones de la couche cachée du réseau NARX-8-pip est la même que celle du réseau RVRNN-8-pip qui est représentée par la figure 43. La structures des neurones de la couche cachée du réseau NARX-6-pip est également la même que celle du réseau RVRNN-8-pip représentée par la figure 43. La structure des neurones de la couche de sortie du réseau NARX-6-pip est également la même que celle du réseau RVRNN-6-pip représentée par la figure 44.

2.6 Implantation des architectures pipelinées des modèles RVTDDN, RVRNN et NARX en utilisant les blocs de l'outil XSG

L'implantation des architectures pipelinées en utilisant les blocs de l'outil XSG reste la même que celle des architectures pseudo-conventionnelles. Afin de réduire le chemin critique et ainsi augmenter considérablement la fréquence maximale d'opération seuls des délais supplémentaires sont ajoutés. Après plusieurs tests réalisés avec différentes valeurs du nombre de neurones à la couche cachée, du nombre de bits de données et de la taille des mémoires RAM, les architectures pipelinées retenues dans ce mémoire pour l'implantation matérielle sont le réseau NARX- N_1 -pip, le réseau RVRNN- N_1 -pip et le réseau RVTDDN- N_1 -pip, avec N_1 correspondant à 6 ou 8 neurones à la couche cachée. Le nombre d'entrées dépend de l'architecture en question tel qu'expliqué dans la section 2.4.

2.6.1 Implantation de l'architecture RVTDDN- N_1 -pip

Le système de test avec modulation 16-QAM à bande de base de l'architecture de modélisation du sous-système *Neural Networks* à base du réseau RVTDDN-8-pip, du réseau RVRNN- N_1 -pip ou du réseau NARX- N_1 -pip et du bloc correspondant *JTAG Hardware co-simulation* sont représentés sur la figure 46. Cependant, le système de test avec modulation 16-QAM à bande de base de l'architecture de modélisation du sous-système *Neural Networks* basé sur le réseau RVTDDN-6-pip et le bloc correspondant *JTAG Hardware co-simulation* est représenté sur la figure 47.

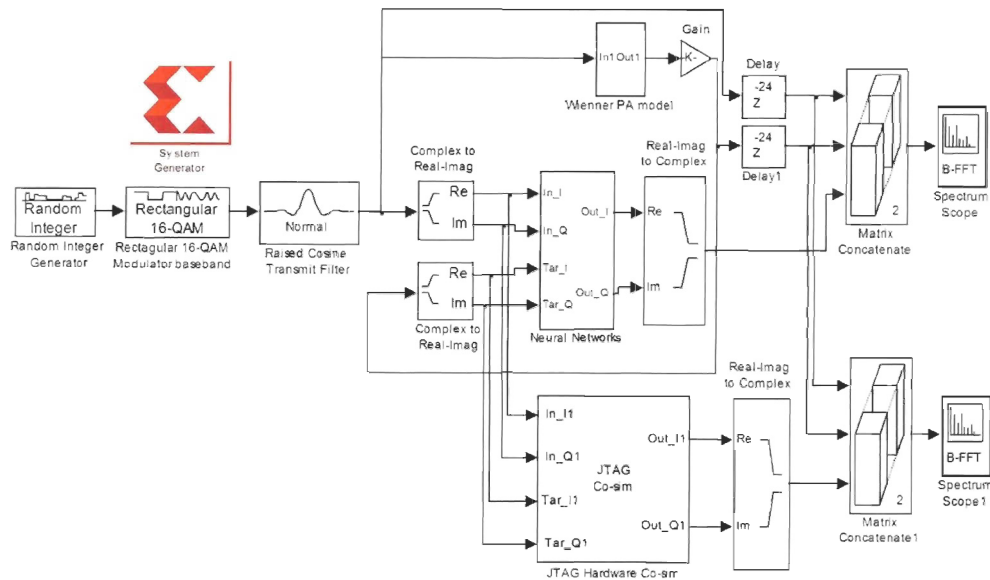


Figure 46: Système de test avec modulation 16-QAM à bande de base des architectures de modélisation par le sous-système *Neural Networks* basé sur le réseau RVTDDN-8-pip, le réseau RVRNN- N_1 -pip et le réseau NARX- N_1 -pip et le bloc correspondant *JTAG Hardware Co-simulation*.

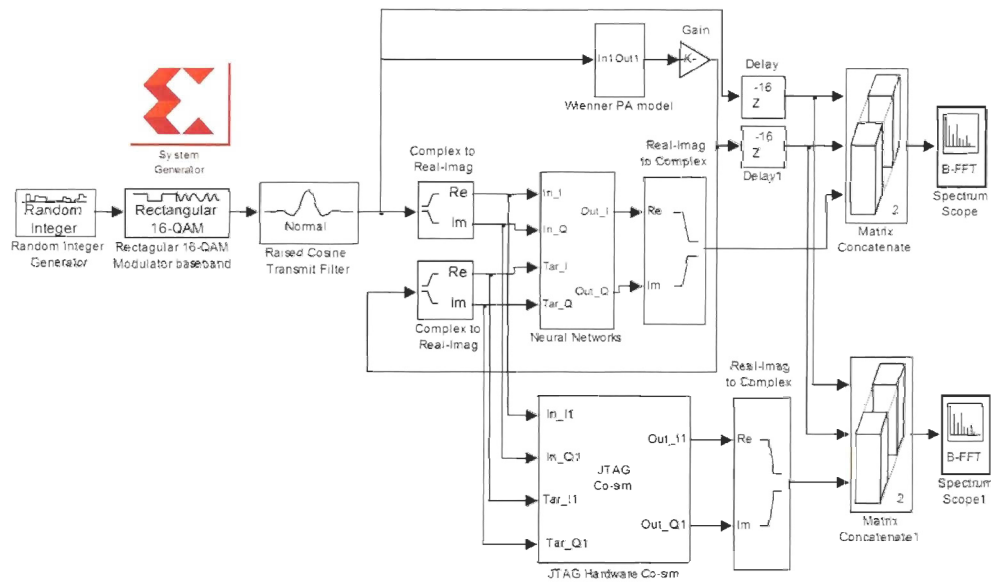


Figure 47: Système de test avec modulation 16-QAM à bande de base des architectures de modélisation par le sous-système *Neural Networks* basé sur le réseau RVTDDN-8-pip et le bloc correspondant *JTAG Hardware Co-simulation*

Dans le premier cas (figure 46), la sortie du réseau est retardée de 24 délais, alors que dans le second (figure 47) elle est retardée de 16 délais. En d'autres termes, le diagramme de la figure 46 est utilisé dans la modélisation par les réseaux RVRNN-8-pip, NARX-8-pip, RVRNN-6-pip et NARX-6-pip, alors que celui de la figure 47 est utilisé dans la modélisation par le réseau RVTDDN-6-pip seulement.

Les composants à base de blocs de l'outil XSG de ces systèmes de test sont identiques à ceux de la figure 23 et avec la même configuration, excepté les délais qui sont différents. Les architectures du réseau RVTDDN-6-pip et du réseau RVTDDN-8-pip sont respectivement similaires aux architectures du réseau RVTDDN-6-pse et du réseau RVTDDN-8-pse. Le réseau RVTDDN-8-pip correspond au réseau RVTDDN-6-pip mais avec deux neurones de plus à la couche cachée.

2.6.1.1 Implantation d'un neurone de la couche cachée de l'architecture RVTDDN- N_1 -pip en utilisant les blocs de l'outil XSG

La figure 48 représente l'implantation de la somme (équation 2.40) dans chaque neurone de la couche cachée de l'architecture RVTDDN-6-pip.

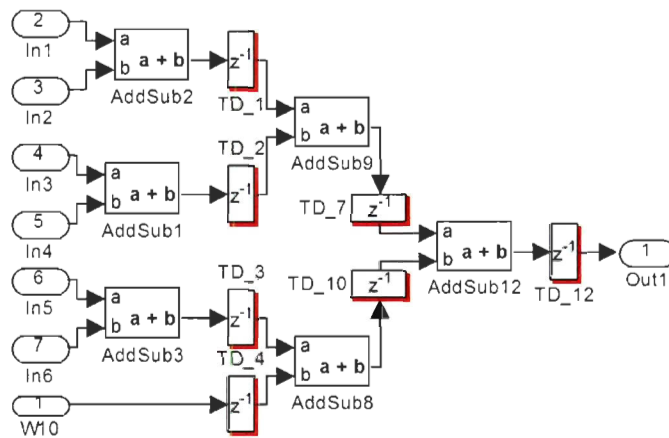


Figure 48: Implantation de la somme (équation 2.40) dans chaque neurone de la couche cachée de l'architecture RVTDDN-6-pip.

La figure 49 représente l'implantation d'un neurone de la couche cachée du réseau RVTDDN-6-pip en utilisant les blocs de l'outil XSG.

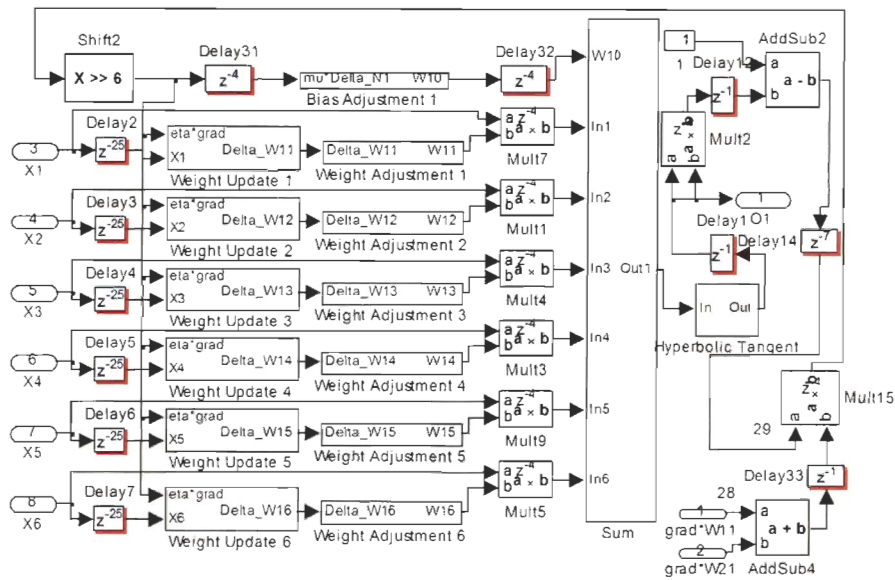


Figure 49: Implantation d'un neurone de la couche cachée de l'architecture RVTDNN-6-pip en utilisant les blocs de l'outil XSG.

La figure 50 représente l'implantation d'un neurone de la couche cachée du réseau RVTDNN-8-pip en utilisant les blocs de l'outil XSG.

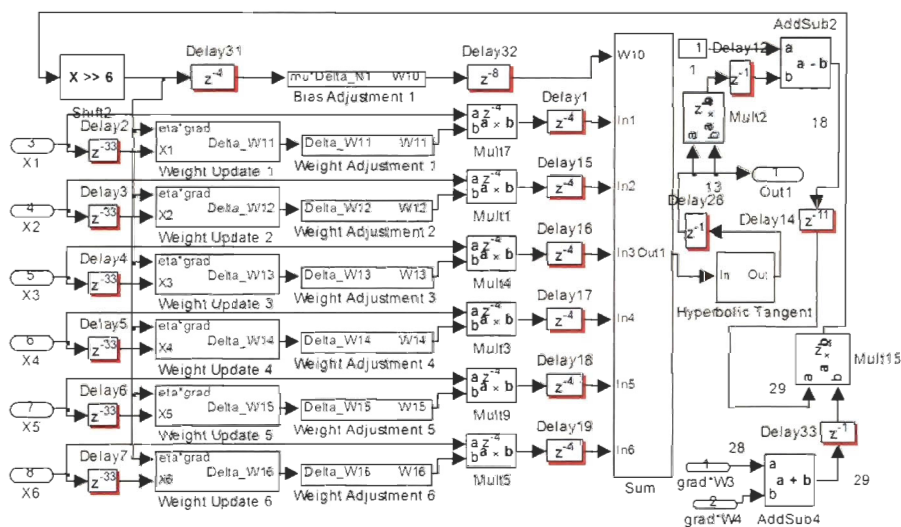


Figure 50: Implantation d'un neurone de la couche cachée de l'architecture RVTDNN-8-pip en utilisant les blocs de l'outil XSG.

L'implantation de la somme dans chaque neurone de la couche cachée de l'architecture RVTDDN-8-pip est identique à celle du réseau RVTDDN-6-pip de la figure 48.

2.6.1.2 Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pip en utilisant les blocs de l'outil XSG

La figure 51 illustre l'implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pip en utilisant les blocs de l'outil XSG.

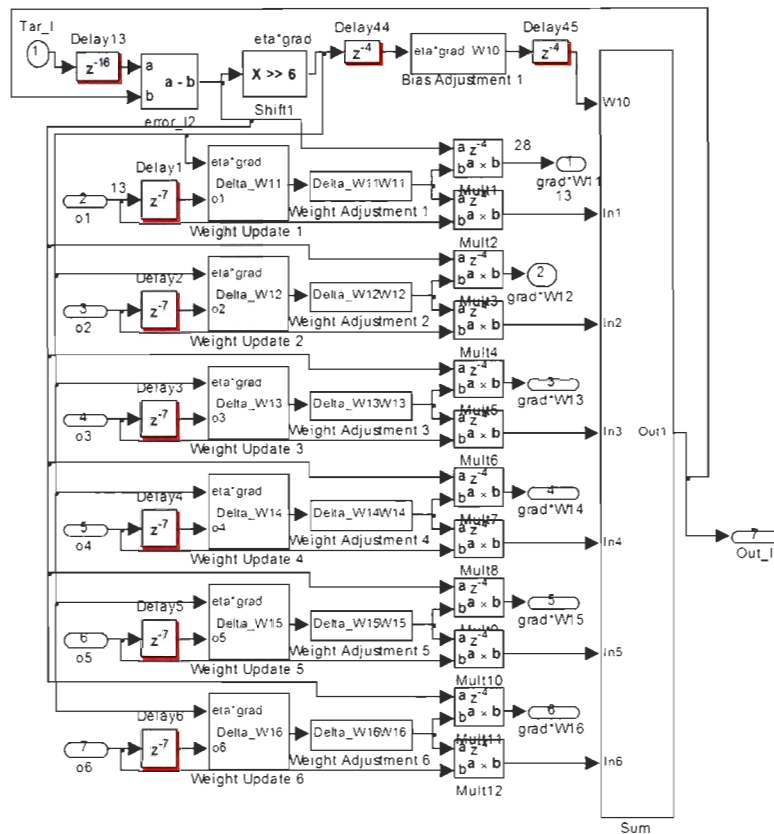


Figure 51: Implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-6-pip en utilisant les blocs de l'outil XSG.

L'implantation de la somme dans chaque neurone de la couche de sortie et de la couche cachée de l'architecture RVTDDN-6-pip est identique, parce que le nombre d'entrées est égal au nombre de neurones à la couche cachée. La figure 52 illustre l'implantation d'un neurone de la couche de sortie de l'architecture RVTDDN-8-pip en utilisant les blocs de l'outil XSG.

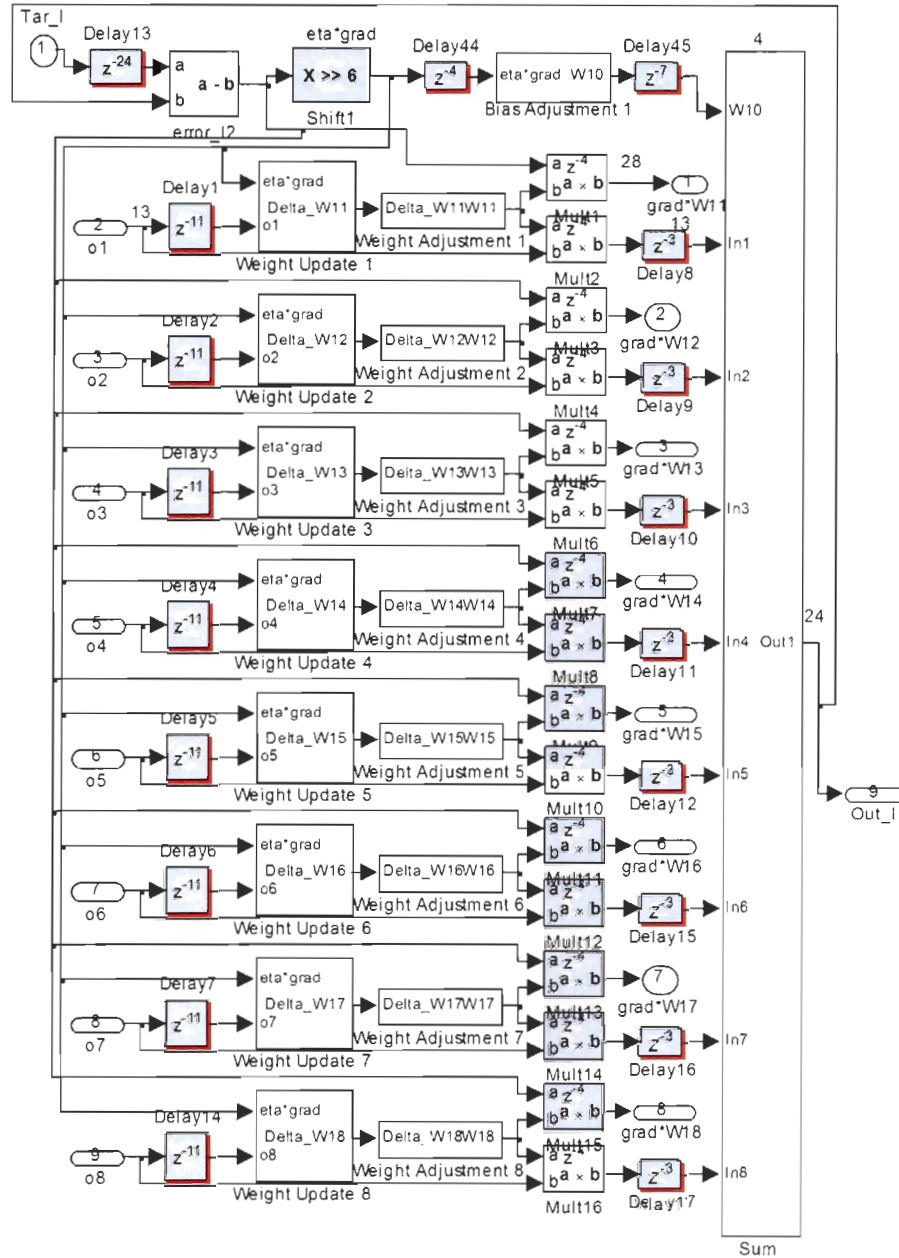


Figure 52: Implmentation d'un neurone de la couche de sortie de l'architecture RVTDDN-8-pip en utilisant les blocs de l'outil XSG.

La somme dans chaque neurone de la couche de sortie du réseau RVTDDN-8-pip a deux entrées de plus que celle de la couche de sortie du réseau RVTDDN-6-pip et est représentée sur la figure 53.

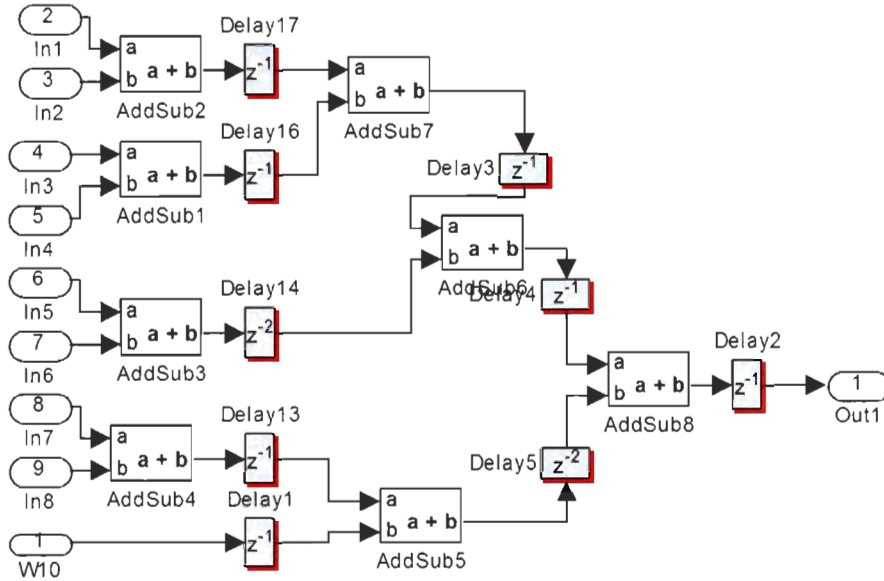


Figure 53: Implantation de la somme (équation 2.42) dans chaque neurone de la couche de sortie de l'architecture RVTDDN-6-pip.

2.6.2 Implantation de l'architecture RVRNN- N_1 -pip

Les architectures du réseau RVRNN-6-pip et du réseau RVRNN-8-pip sont respectivement identiques aux architectures du réseau RVTDDN-6-pse de la figure 32 et du réseau RVTDDN-8-pse. Le réseau RVRNN-8-pip correspond au réseau RVRNN-6-pip avec deux neurones de plus à la couche cachée. Les fonctions et blocs de l'environnement Simulink et de l'outil XSG de ces deux architectures sont similaires à celles de la section 2.4.1.

2.6.2.1 Implantation d'un neurone de la couche cachée de l'architecture RVRNN- N_1 -pip en utilisant les blocs de l'outil XSG

Un neurone de la couche cachée de l'architecture RVRNN- N_1 -pip a deux entrées de plus que le réseau RVTDDN- N_1 -pip, dues à la réaction des sorties. L'implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pip en utilisant les blocs de l'outil XSG est représentée sur la figure 54.

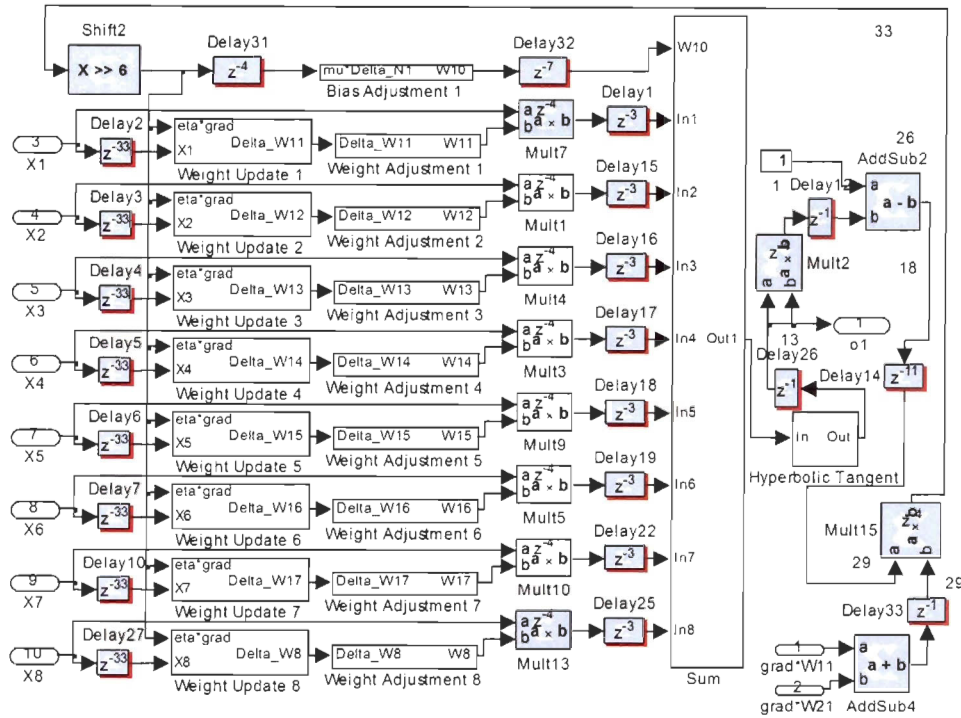


Figure 54: Implantation d'un neurone de la couche cachée de l'architecture RVRNN-6-pip en utilisant les blocs de l'outil XSG.

La somme dans chaque neurone de la couche cachée de l'architecture RVRNN-6-pip est la même que celle dans chaque neurone de la couche de sortie de l'architecture RVTDDN-8-pip de la figure 53. Chaque neurone de la couche cachée de l'architecture RVRNN-8-pip est identique à celle du réseau RVRNN-6-pip de la figure 54. La somme dans chaque neurone de la couche cachée de l'architecture RVRNN-8-pip est identique à celle dans chaque neurone de la couche de sortie de l'architecture RVTDDN-8-pip de la figure 53.

2.6.2.2 Implantation d'un neurone de la couche de sortie de l'architecture RVRNN- N_1 -pip en utilisant les blocs de l'outil XSG

Un neurone de la couche de sortie de l'architecture RVRNN-6-pip est différent de celui de l'architecture RVTDDN-6-pip de la figure 51 en terme de retard et est représenté sur la figure 55. Cependant, leur bloc somme est identique.

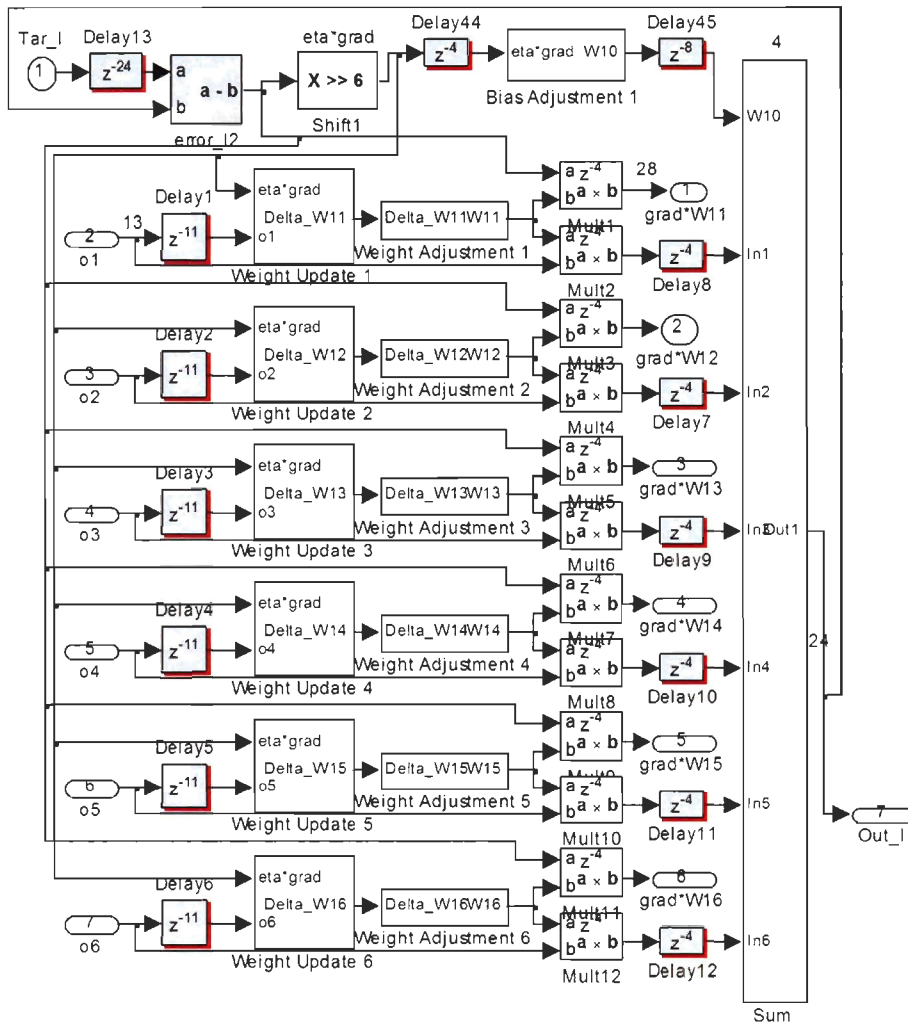


Figure 55: Implantation d'un neurone de la couche de sortie de l'architecture RVRNN-6-pip en utilisant les blocs de l'outil XSG.

Le neurone de la couche de sortie de l'architecture RVRNN-8-pip et sa somme sont les mêmes que ceux de l'architecture RVTDDN-8-pip de la figure 52.

2.6.3 Implantation de l'architecture NARX- N_1 -pip

Les architectures NARX-6-pip et NARX-8-pip sont respectivement identiques aux architectures NARX-6-pse de la figure 34 et NARX-8-pse. Les fonctions et blocs de l'environnement Simulink et de l'outil XSG de ces deux architectures sont similaires à celles de la section 2.4.1.

2.6.3.1 Implantation d'un neurone de la couche cachée de l'architecture NARX- N_1 -pip en utilisant les blocs de l'outil XSG

Un neurone de la couche cachée de l'architecture NARX-6-pip a 4 entrées de plus que celui du réseau RVRNN-6-pip et est représenté sur la figure 56.

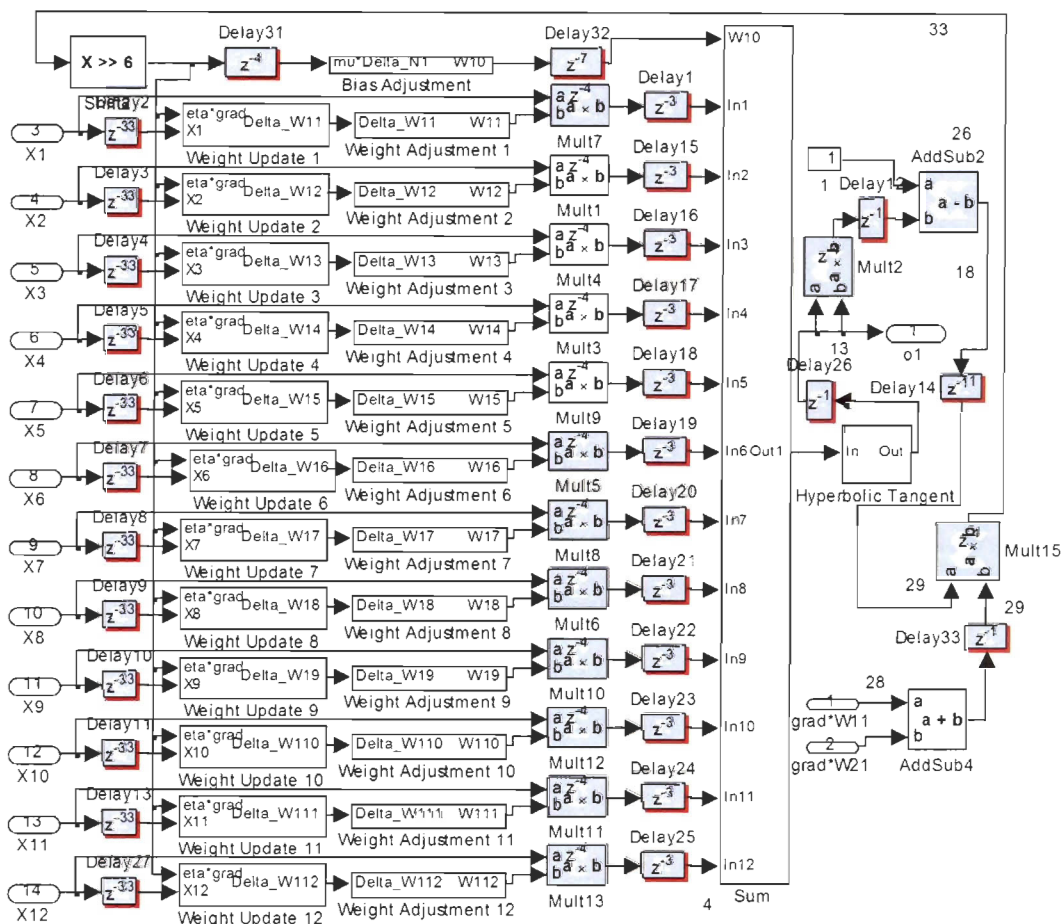


Figure 56: Implantation d'un neurone de la couche cachée de l'architecture NARX-6-pip en utilisant les blocs de l'outil XSG.

La somme dans chaque neurone de la couche cachée du réseau NARX-6-pip a 4 entrées de plus que celle de la couche cachée du réseau RVRNN-6-pip et est représentée sur la figure 57. La somme dans chaque neurone de la couche cachée du réseau NARX-8-pip est similaire à celle de la couche cachée du réseau NARX-6-pip de la figure 57, car leurs neurones à la couche cachée ont la même structure (voir figure 56).

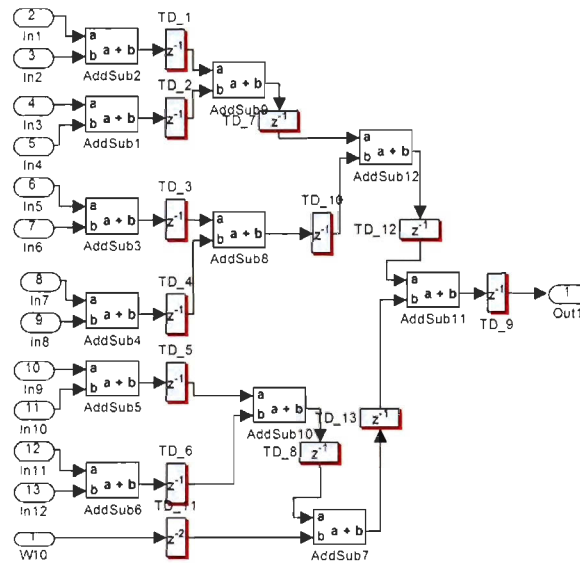


Figure 57: Implantation de la somme (équation 2.42) dans chaque neurone de la couche cachée de l'architecture NARX-6-pip.

2.6.3.2 Implantation d'un neurone de la couche de sortie de l'architecture NARX- N_1 -pip en utilisant les blocs de l'outil XSG

Les neurones des couches de sortie des architectures du réseau NARX-6-pip et du réseau NARX-8-pip sont respectivement identiques à ceux du réseau RVRNN-6-pip de la figure 55 et du réseau RVTDDN-8-pip de la figure 52. Cela implique également que la somme de chaque neurone de la couche de sortie de l'architecture du réseau NARX-6-pip et celle du réseau NARX-8-pip sont respectivement identiques à celles du réseau RVRNN-6-pip et du réseau RVRNN-8-pip de la figure 53.

2.7 Implantation des architectures de modélisation sur la puce FPGA

La carte d'évaluation FPGA Virtex-6 XC6VLX240T-1FFG1156 ML605 est la plate-forme de base de l'environnement Xilinx pour le développement des applications très performantes dans le domaine des télécommunications sans fil (Xilinx, 2012b). Le type de puce FPGA de la famille Virtex-6 est XC6VLX240T, -1 est le niveau de vitesse, FF est le type de boîtier et 1156 est le nombre de pins (Xilinx, 2012a). Chaque tranche (*slice*) logique de la puce FPGA Virtex-6 contient 4 tables LUT et 8 bascules (Flip-Flop). Chaque tranche DSP48E1 contient

un multiplieur de 25×18 bits, un additionneur et un accumulateur de 48 bits (Bahoura et Park, 2012; Xilinx, 2012a). Chaque bloc RAM est fondamentalement à 36 Kb (Xilinx, 2012a), en l'occurrence RAMB36E1, mais il peut être utilisé comme deux blocs indépendants de 18 Kb (RAMB18E1). La puce Virtex-6 de la carte ML605 utilisée contient 37680 tranche (*slices*) logiques, 768 DSP48E1 et 416 blocs RAM de 36 Kb chacun. Le bloc *System Generator* permet de générer les fichiers VHDL (ou Verilog) correspondant au diagramme conçu à l'aide des composants de l'outil XSG. Il permet également de créer un composant *Hardware Co-simulation* associé au fichier *bitstream* qui permet d'exécuter le diagramme sur la puce FPGA à partir de l'environnement Matlab/Simulink. La figure 58 représente le système de co-simulation matérielle contenant un ordinateur, la puce FPGA Virtex-6 de la carte ML605 et le câble USB.

Après génération du bloc *Hardware Co-simulation* des architectures de modélisation NARX- N_1 -pip, RVRNN- N_1 -pip et RVTDNN- N_1 -pip, ce bloc pourra être exécuté en mode *single-step* ou *free-running*. En mode *single-step*, il y a synchronisation entre la puce FPGA et l'environnement Simulink, car l'horloge de la puce FPGA provient de l'environnement Simulink. C'est donc le mode choisi. En mode *free-running*, la puce FPGA fonctionne avec son horloge interne en échantillonnant de manière asynchrone les signaux provenant de l'environnement Simulink. Les données de toutes les architectures sont quantifiées en utilisant le format FIX24RS (Bahoura et Park, 2012). Cependant, un format non signé de 1 bit au total, sans fraction, est utilisé pour représenter respectivement la constante 0 de la tangente hyperbolique et la constante 1 de la dérivé de la tangente hyperbolique. Une table de correspondance (LUT) basée sur un bloc RAM de 2^{15} mots de 24 bits est utilisée pour l'implantation de la fonction tangente hyperbolique, tirant profit de ces caractéristiques asymétriques.

2.7.1 Implantation des architectures pseudo-conventionnelles

La compilation de chacun de ces algorithmes pour la génération du fichier *Bitstream* s'est faite en allant dans le bloc *System Generator*, puis dans l'option *compilation*, en choisissant l'option *Bitstream*, ensuite dans le champ *Part* en sélectionnant la carte Virtex-6 XC6VLX240T-1FFG1156 et enfin dans l'onglet *Clocking*, en mettant dans l'option *FPGA clock period* la valeur 100 ns. Cette configuration a permis la génération, avec succès, des fichiers *bitstream* pour toutes ces architectures de modélisation. À la fin de la compilation, une fenêtre comportant les détails sur les opérations effectuées lors de la génération des ressources

de ces architectures s'ouvre. Dans cette fenêtre se trouvent les ressources estimées de chacune de ces architectures de modélisation pseudo-conventionnelles telles que représentées dans le tableau 1.

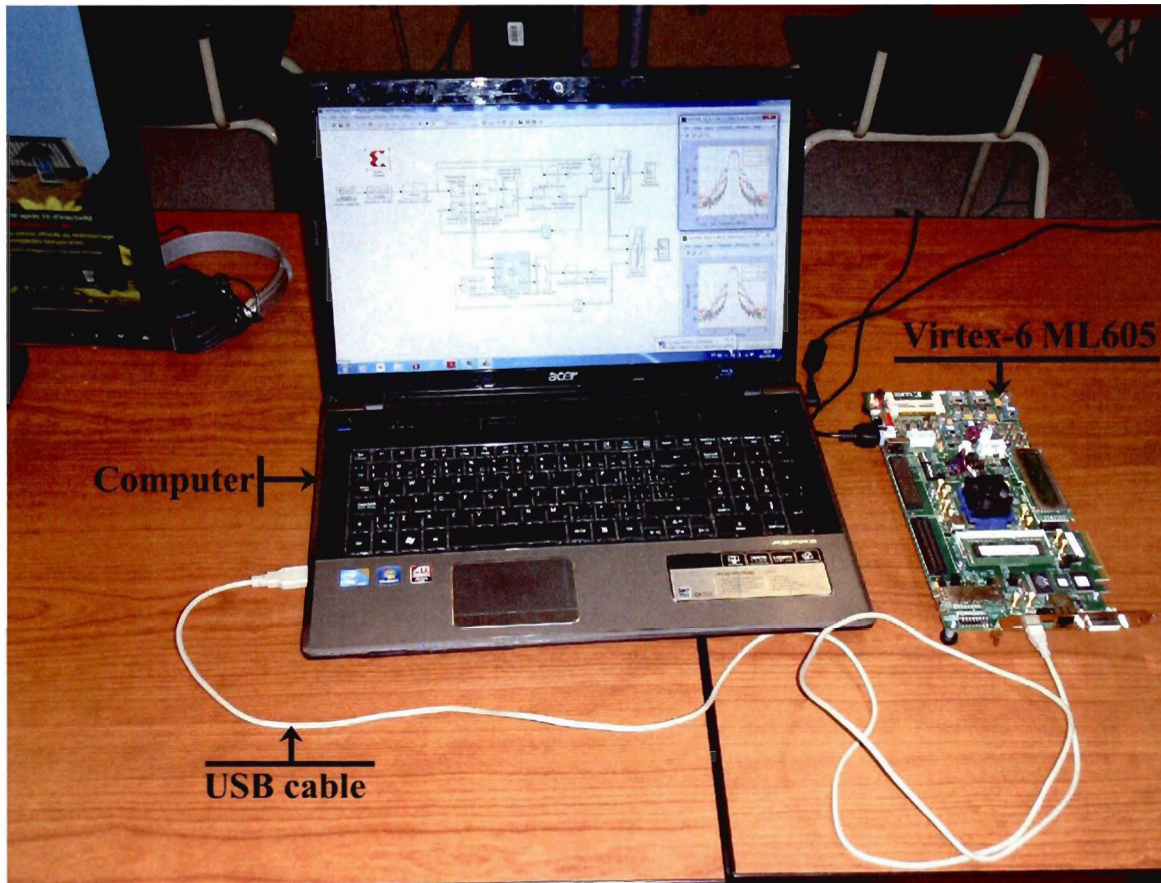


Figure 58: Système de co-simulation matérielle contenant un ordinateur, la puce FPGA Virtex-6 de la carte ML605 et le câble USB.

Pour le même type d'architecture, les ressources requises augmentent avec le nombre de neurone à la couche cachée N_2 . Il est à noter que les architectures NARX nécessitent plus de ressources que leurs équivalents RVRNN et RVTDDN, car elles présentent plus d'entrées N_0 . La fréquence maximale d'opération (MOF) pour ces architectures pseudo-conventionnelles est d'environ 37.4 MHz.

Tableau 1: Utilisation des ressources et fréquence maximale d’opération des architectures de modélisation du réseau RVTDDN- N_1 -pse, du réseau RVRNN- N_1 -pse et du réseau NARX- N_1 -pse dans la carte Virtex-6 XC6VLX240T. Les ressources disponibles sont données entre parenthèse, ainsi que le pourcentage de ressources estimées de chaque architecture de modélisation.

Modelisation Architecture	RVTDDN-6-pse	RVRNN-6-pse	NARX-6-pse	RVTDDN-8-pse	RVRNN-8-pse	NARX-8-pse
Resource utilization						
Flip-Flop (301.440)	2.376 (0.78 %)	3.000 (0.99 %)	4.248 (1.40 %)	3.104 (1.02 %)	3.920 (1.30 %)	5.552 (1.84 %)
LUT (150.720)	10.552 (7.00 %)	12.616 (8.37 %)	16.472 (10.92 %)	13.937 (9.24 %)	16.708 (11.08 %)	22.135 (14.68 %)
Bonded IOB (600)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)
RAMB36E1 (416)	132 (31.73 %)	132 (31.73 %)	132 (31.73 %)	176 (42.30 %)	176 (42.30 %)	176 (42.30 %)
DSP48E1 (768)	240 (31.25 %)	288 (37.50 %)	384 (50.00 %)	320 (41.66 %)	384 (50.00 %)	512 (66.66 %)
Max. Opera. Freq. (MHz)	37.440	37.429	37.429	37.418	37.407	37.407

Tableau 2: Ressources utilisées et fréquence maximale d’opération des architectures de modélisation du réseau RVTDDN- N_1 -pip, du réseau RVRNN- N_1 -pip et du réseau NARX- N_1 -pip dans la carte Virtex-6 XC6VLX240T. Les ressources disponibles sont données entre parenthèse, ainsi que le pourcentage des ressources estimées de chaque architecture de modélisation.

Modelisation Architecture	RVTDDN-6-pip	RVRNN-6-pip	NARX-6-pip	RVTDDN-8-pip	RVRNN-8-pip	NARX-8-pip
Resource utilization						
Flip-Flop (301.440)	10.752 (3.56 %)	14.424 (4.78 %)	19.368 (6.42 %)	15.808 (5.24 %)	19,088 (6.33 %)	25.648 (8.50 %)
LUT (150.720)	13.354 (8.86 %)	17,521 (11.62 %)	22,956 (15.23 %)	19.252 (12.77 %)	23,434 (15.54 %)	30,608 (20.30 %)
Bonded IOB (600)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)	145 (24.16 %)
RAMB36E1 (416)	132 (31.73 %)	132 (31.73 %)	132 (31.73 %)	176 (42.30 %)	176 (42.30 %)	176 (42.30 %)
DSP48E1 (768)	240 (31.25 %)	288 (37.50 %)	384 (50.00 %)	320 (41.66 %)	384 (50.00 %)	512 (66.66 %)
Max. Opera. Freq. (MHz)	156.128	156.128	156.128	156.128	156.128	156.128

2.7.2 Implantation des architectures pipelinées

Les architectures de modélisation RVTDDN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip ont les mêmes ressources IOB, RAMB36E1 et DSP48E1 (tableau 2) que leurs correspondantes pseudo-conventionnelles (tableau 1). Cependant les architectures pipelinées nécessitent plus de ressources de la table LUT et de bascules (Flip-Flop) à cause des retards insérés (tableau 2). La fréquence maximale d'opération (MOF) a augmentée de plus de 4 fois pour toutes les architectures de modélisation du réseau RVTDDN- N_1 -pip, du réseau RVRNN- N_1 -pip et du réseau NARX- N_1 -pip, soit 156.128 MHz (voir le tableau 2) par rapport aux architectures de modélisation RVTDDN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse qui ont chacune une fréquence MOF d'environ 37.4 MHz (tableau 1). Cette augmentation est due à la diminution du chemin critique. L'uniformité de la fréquence MOF pour les architectures avec pipeline justifie une bonne optimisation et une bonne synchronisation des retards. Dans (Ntouné Ntouné et al., 2012a), le réseau RVTDDN avec pipeline a une fréquence MOF de 139.391 MHz alors que dans le tableau 2, elle est de 156.128 MHz, d'où une augmentation de 12 %. Cette différence de modélisation provient principalement du choix du format de données, au lieu du format FIX36TW (Bahoura et Park, 2012) utilisé dans (Ntouné Ntouné et al., 2012a), le format FIX24RS est utilisé dans ce mémoire. Le format FIX36TW est un format complément à 2 signé, à virgule fixe avec 36 bits dont une partie fractionnaire de 32 bits et comme option de débordement *wrap* et de quantification *truncate*.

2.8 Expérimentation et analyse des résultats de modélisation par des architectures pseudo-conventionnelles et pipelinées

Les architectures pseudo-conventionnelles et pipelinées de modélisation sont évaluées avec un signal de test à bande de base modulé 16-QAM en utilisant la boîte à outils de communication de Matlab. Dans ce système de test, les 16 points de constellation de la modulation 16-QAM sont analysés lors du processus de modélisation avant et après leur amplification par le modèle de référence de Wiener. Pour mener à bien cette évaluation, différentes courbes sont également tracées, dont les caractéristiques AM/AM et AM/PM, le spectre du signal à l'entrée et à la sortie du modèle de référence de Wiener et les différentes architectures de modélisation proposées, l'évolution des composants cartésiens I et Q en fonction du temps, l'erreur quadratique moyenne (MSE) relative à la convergence et les diagrammes de constella-

tion. L'amplitude du vecteur d'erreur (EVM) est aussi calculée pour exprimer le pourcentage de fidélité de modélisation de la constellation des différentes architectures. Pour le tracé du spectre, de la constellation, des conversions AM/AM et AM/PM et de l'évolution des composants cartésiens I et Q en fonction du temps, la simulation est d'abord faite sur une durée de 20 s pour une fréquence d'échantillonnage de 8 MHz. Puis, les poids optimisés sont réinjectés dans les différents réseaux de neurones au niveau du registre à décalage de l'ajustement des biais et des poids (voir section 2.4.1.2). Enfin, la simulation est reprise sur une durée d'une seconde. Pour l'estimation du paramètre MSE relatif à la convergence, les poids des programmes de connexion des architectures de modélisation sont initialisés avec la fonction *randn* du logiciel Matlab.

2.8.1 Expérimentation des architectures pseudo-conventionnelles et pipelinées

Les systèmes de test sont représentés à la figure 23 pour les architectures de modélisation RVTDDN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse, à la figure 46 pour les architectures de modélisation RVTDDN-8-pip, RVRNN- N_1 -pip et NARX- N_1 -pip et à la figure 47 pour l'architecture de modélisation RVTDDN-6-pip. Le bloc *JTAG Co-simulation* sert à vérifier le bon fonctionnement des différentes architectures dans la puce FPGA Virtex-6 de la carte ML605. Les courbes sont uniquement tracées avec le bloc de simulation *Neural Networks* pouvant être n'importe laquelle des architectures dépendamment du fait que le système de test soit pseudo-conventionnel ou avec pipeline. Le fichier *Bitstream*, quant à lui, est généré pour toutes les architectures afin d'évaluer les différentes ressources estimées de chaque architecture comparativement à celles disponibles sur le FPGA Virtex-6 de la carte ML605 (voir la section 2.7.1 et la section 2.7.2).

2.8.2 Analyse des résultats de modélisation par des architectures pseudo-conventionnelles et pipelinées

Les résultats de modélisation montrent clairement la bonne performance des différentes architectures de modélisation des non-linéarités et des effets mémoire de l'amplificateur de puissance par réseaux de neurones. Cependant, certains modèles sont plus performants que d'autres. La constellation permet de visualiser les distorsions à l'intérieur de la bande de

fréquence EVM dues au modèle de référence de Wiener. Pour la caractérisation du modèle de référence de Wiener et des modèles neuronaux, les caractéristiques de conversion AM/AM et de compression AM/PM sont utilisées. L'évolution temporelle des composants cartésiens réel (I) et imaginaire (Q) permet de montrer le degré de similitude entre le modèle de référence de Wiener et les architectures proposées. Le spectre de puissance illustre, lui aussi, la précision des différentes architectures par rapport au modèle de référence de Wiener. Le spectre permet aussi de visualiser les distorsions à l'extérieur de la bande de fréquence ainsi que la fidélité des architectures à modéliser ces distorsions. Le paramètre EVM en pourcent est calculé pour spécifier le pourcentage de modélisation des paramètres EVM par les architectures développées par rapport à un amplificateur idéal. La fonction MSE est utilisée pour évaluer la fidélité du spectre, des conversions AM/AM et AM/PM, et des courbes I et Q en fonction du temps des architectures proposées par rapport au modèle de référence de Wiener. Les fonctions MSE pour les conversions AM/AM et AM/PM et les courbes I et Q en fonction du temps sont également calculées avec ces échantillons des composants cartésiens I et Q. Les fonctions MSE pour l'évaluation de la fidélité du spectre sont calculées par la moyenne des paramètres MSE des composants I et Q de chaque architecture. Le calcul du paramètre MSE relatif à la vitesse de convergence des architectures de modélisation est obtenu à partir de la moyenne de 10 différents tests dans lesquels est faite varier la constante x de la fonction $randn('state',x)$, où x varie de 1 à 10. La variation de x implique une initialisation différente des poids et biais dans les registres à décalage de chaque neurone (voir section 2.4.1.2). Dans chaque test, le paramètre MSE relatif à la vitesse de convergence des architectures de modélisation est calculé sur 20 trames de 4096 échantillons chacune.

2.8.3 Performances des architectures de modélisation RVTDDNN-6-pse, RVRNN-6-pse et NARX-6-pse

La figure 59 représente les diagrammes de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDNN-6-pse. La caractérisation quantitative est faite par calcul du paramètre EVM en pourcent. Le modèle de référence de Wiener a 33.205 % du paramètre EVM tandis que les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDNN-6-pse ont respectivement 33.197 %, 33.199 % et 33.181 % du paramètre EVM.

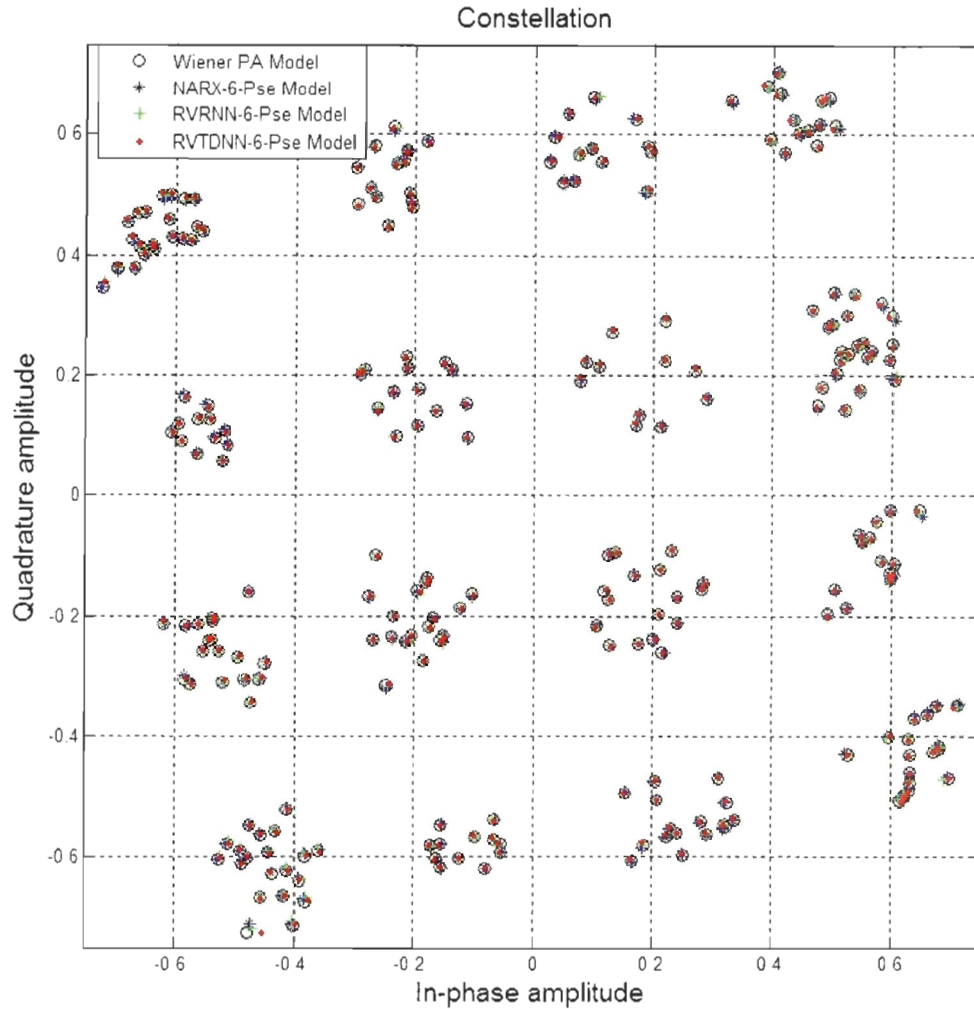


Figure 59: Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse.

Les résultats de simulation de la conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse représentés sur la figure 60 sont quantitativement caractérisés par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse ont respectivement comme paramètre MSE de la conversion AM/AM 1.379×10^{-5} , 0.670×10^{-5} et 1.675×10^{-5} par rapport au modèle de référence de Wiener.

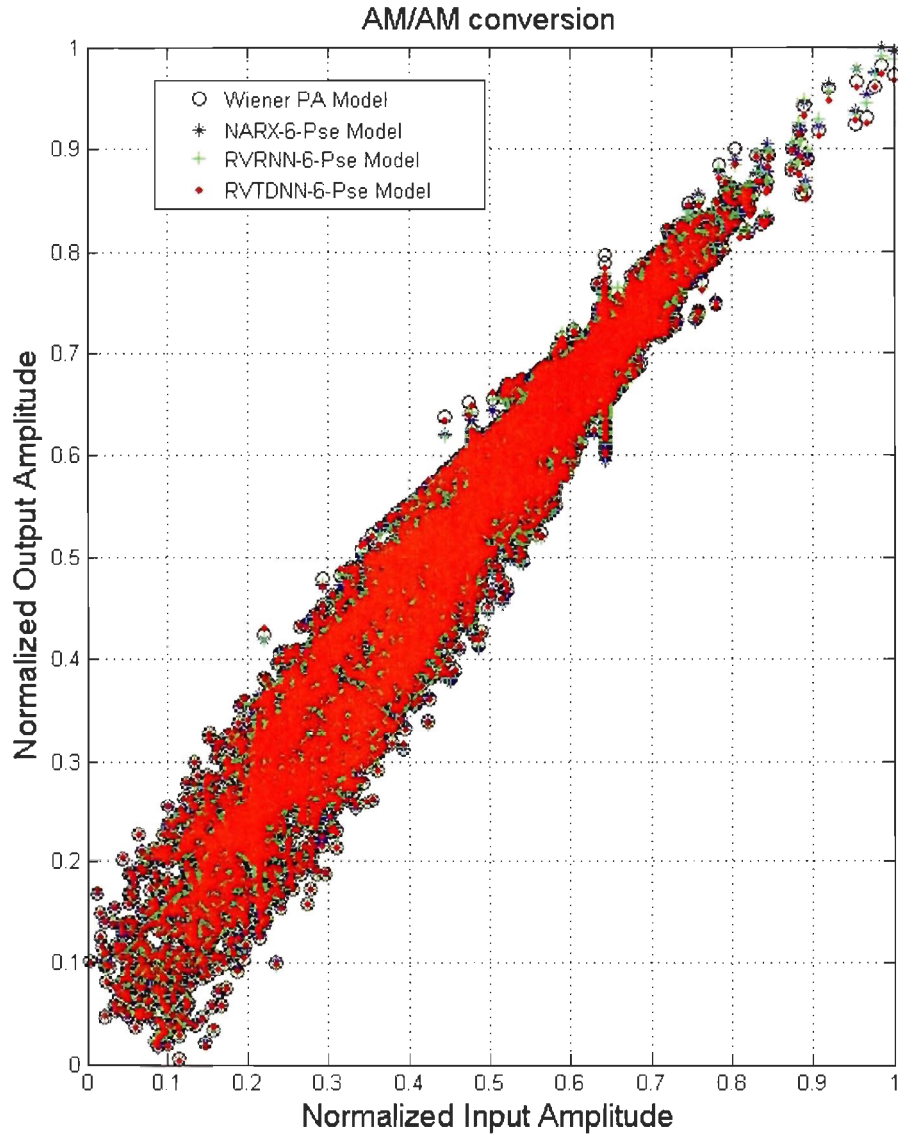


Figure 60: Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.

Le calcul du paramètre MSE de la conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse, représentées sur la figure 61, est utilisé comme méthode d'analyse quantitative. Les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse ont respectivement comme paramètre MSE de la conversion AM/PM 0.141, 0.125 et 0.150 par rapport au modèle de référence de Wiener.

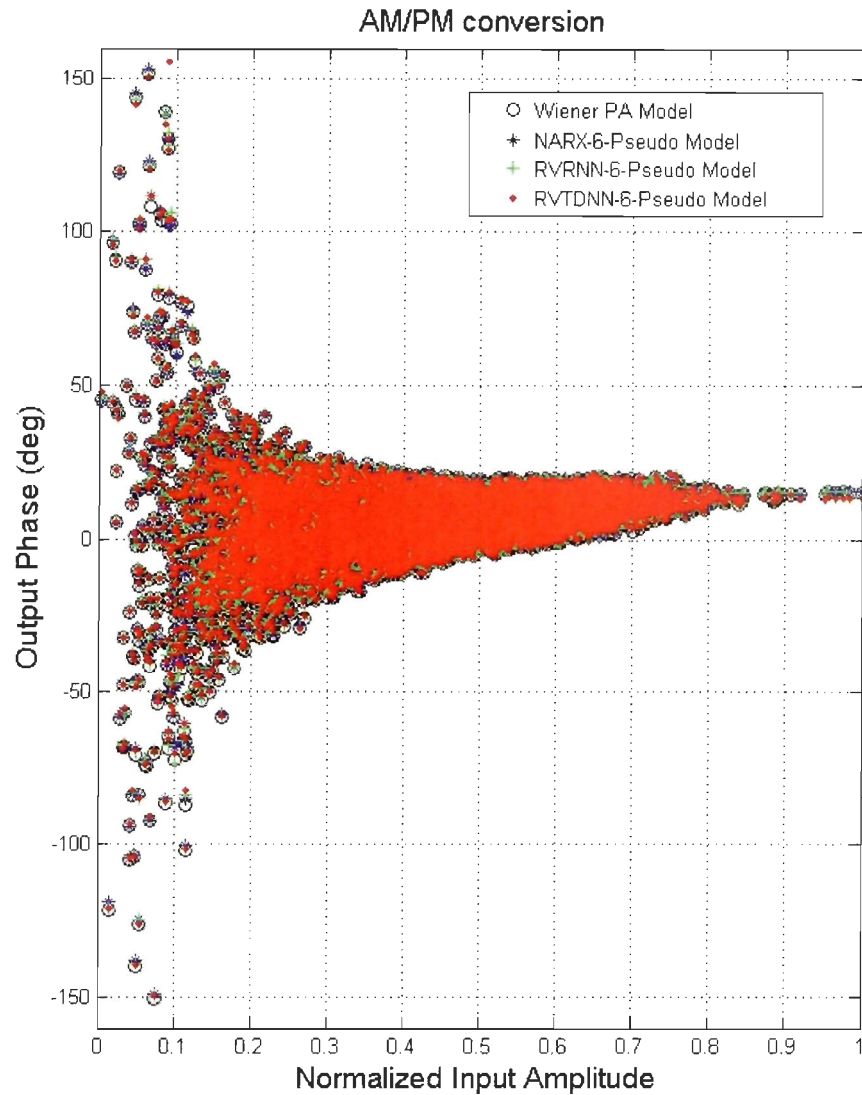


Figure 61: Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.

La comparaison des caractéristiques des composants cartésiens I et Q du modèle de référence de Wiener des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse, représentée sur la figure 62, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 1.175×10^{-5} , 0.459×10^{-5} et 1.040×10^{-5} et en quadrature (Q) 1.924×10^{-5} , 0.186×10^{-5} et 2.118×10^{-5} par rapport au modèle de référence de Wiener.

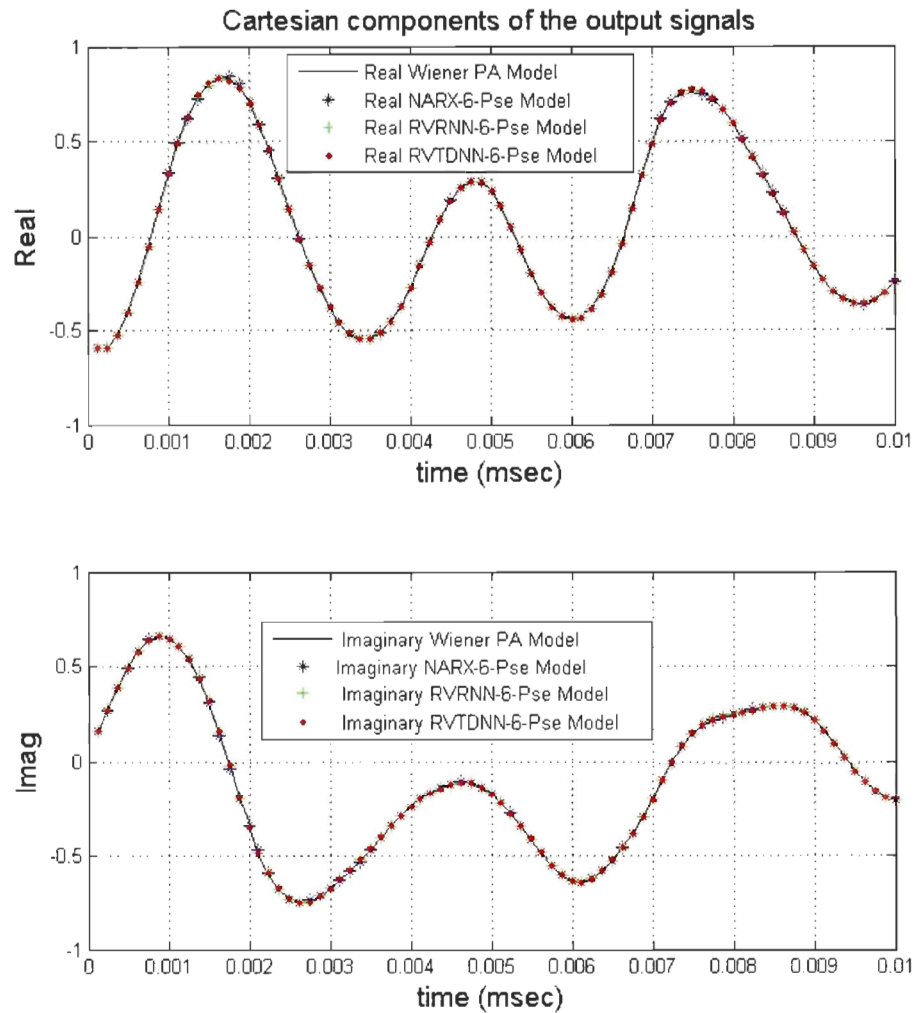


Figure 62: Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse.

La comparaison des résultats de simulation des spectres du signal d'entrée et des signaux de sortie du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse représentés sur la figure 63, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse ont respectivement comme paramètre MSE des spectres 1.549×10^{-5} , 0.116×10^{-5} et 1.579×10^{-5} par rapport au modèle de référence de Wiener.

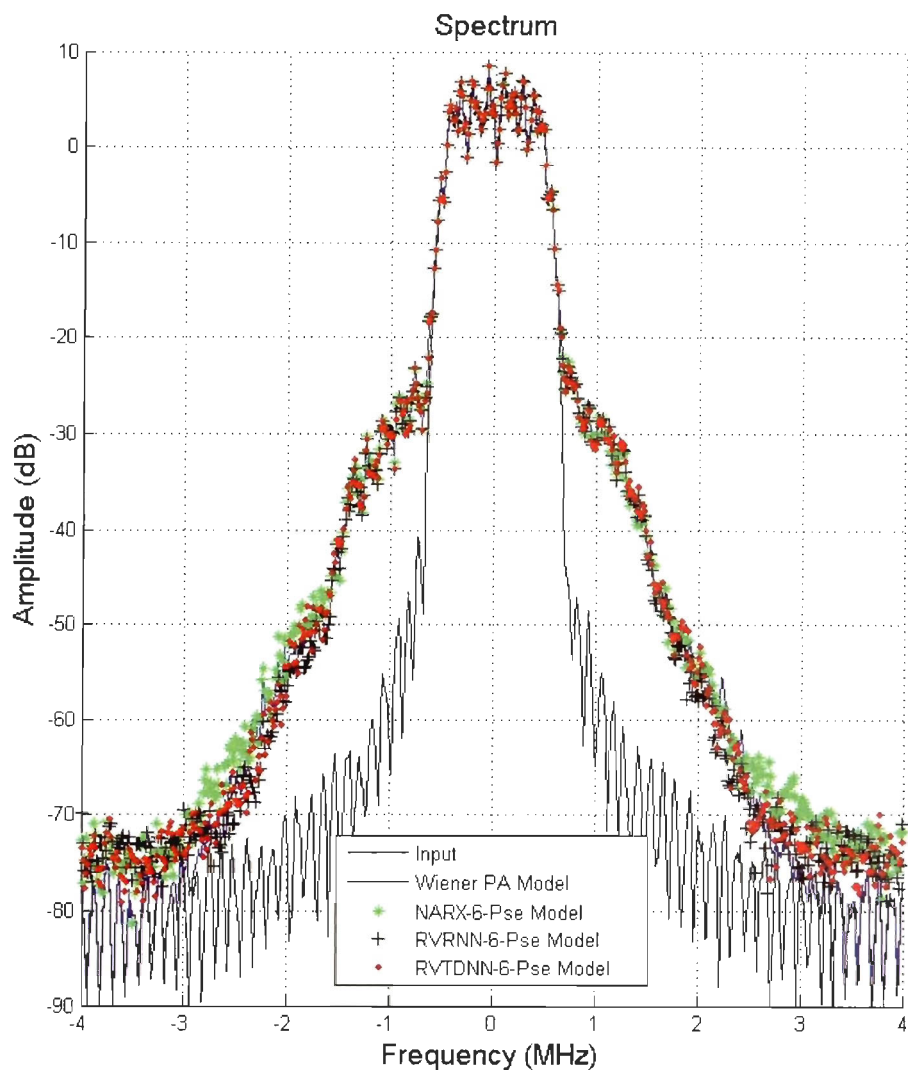


Figure 63: Spectres du signal d'entrée et des signaux de sorties du modèle de référence de Wiener et des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse.

Les convergences des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse mesurées par les paramètres MSE de leurs sorties relativement à celle du modèle de référence de Wiener sont représentées sur la figure 64. Les architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0174, 0.0135 et 0.0105.

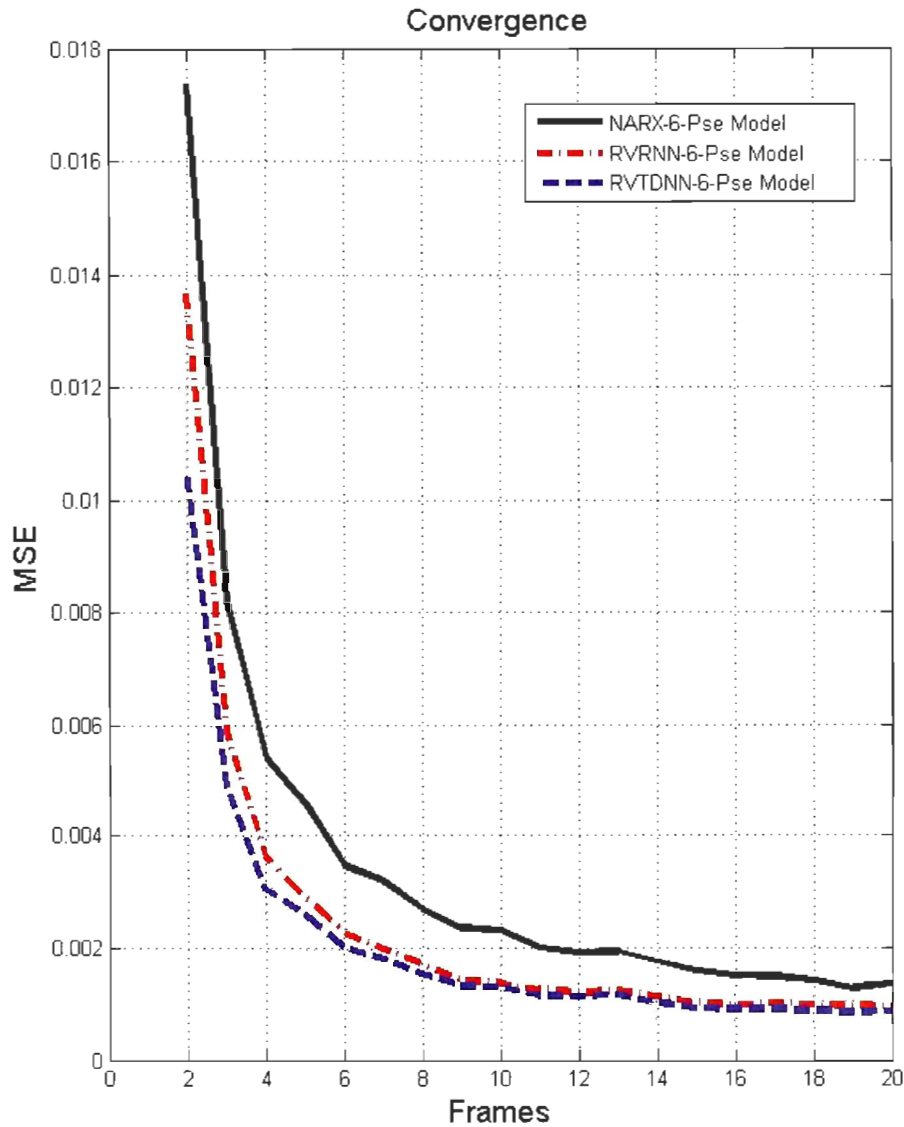


Figure 64: Convergence des architectures de modélisation NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.

De l'analyse des résultats de simulation des différentes courbes servant à la validation des résultats, découle que l'architecture de modélisation RVRNN-6-pse modélise mieux le modèle de référence de Wiener que celle du réseau NARX-6-pse qui, elle-même, le modélise mieux que le réseau RVTDNN-6-pse. En effet, l'architecture de modélisation RVRNN-6-pse a 33.199 % du paramètre EVM, valeur qui est plus proche du paramètre EVM du modèle de référence de Wiener (33.205 %) que celle des architectures de modélisation NARX-6-pse et RVTDNN-

6-pse qui ont respectivement 33.197 % et 33.181 % pour la caractérisation quantitative des diagrammes de constellation. Le calcul des paramètres EVM se fait par rapport au signal d'entrée.

Pour les conversions AM/AM et AM/PM, le paramètre MSE de l'architecture de modélisation RVRNN-6-pse a respectivement 0.670×10^{-5} et 0.125 valeurs qui sont respectivement plus petites que celles du réseau NARX-6-pse (1.379×10^{-5} et 0.141) qui sont elles-mêmes respectivement plus petites que celles du réseau RVTDDN-6-pse (1.675×10^{-5} et 0.150) par rapport au modèle de référence de Wiener.

Le calcul du paramètre MSE pour la comparaison des caractéristiques des composants cartésiens I et Q, montre que le réseau RVRNN-6-pse a une valeur inférieure à celle du réseau NARX-6-pse et du réseau RVTDDN-6-pse. Cependant, pour le composant Q, le réseau NARX-6-pse a 1.175×10^{-5} , qui est supérieure au paramètre MSE du réseau RVTDDN-6-pse qui a 1.040×10^{-5} par rapport au modèle de référence de Wiener.

Pour le spectre, le réseau RVRNN-6-pse (0.116×10^{-5}) a un paramètre MSE inférieur à celui du réseau NARX-6-pse (1.549×10^{-5}) dont le paramètre MSE est inférieur au réseau RVTDDN-6-pse (1.579×10^{-5}) par rapport au modèle de référence de Wiener. Le paramètre MSE de la vitesse de convergence à la 2^{ème} trame du réseau RVTDDN-6-pip (0.0105) étant inférieur à celui du réseau RVRNN-6-pip (0.0135) qui est lui-même inférieur à celui du réseau NARX-6-pip (0.0174) par rapport au modèle de référence de Wiener. Il convient de dire que le réseau RVTDDN-6-pip converge plus vite que le réseau RVRNN-6-pip qui lui-même converge plus vite que le réseau NARX-6-pip (voir figure 64).

2.8.4 Performances des architectures de modélisation RVTDDN-8-pse, RVRNN-8-pse et NARX-8-pse

La figure 65 représente les diagrammes de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse. La caractérisation quantitative est faite par calcul du paramètre EVM en pourcent. Le modèle de référence de Wiener a 33.205 % du paramètre EVM tandis que les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse ont respectivement 33.199 %, 33.192 % et 33.198 % du paramètre EVM.

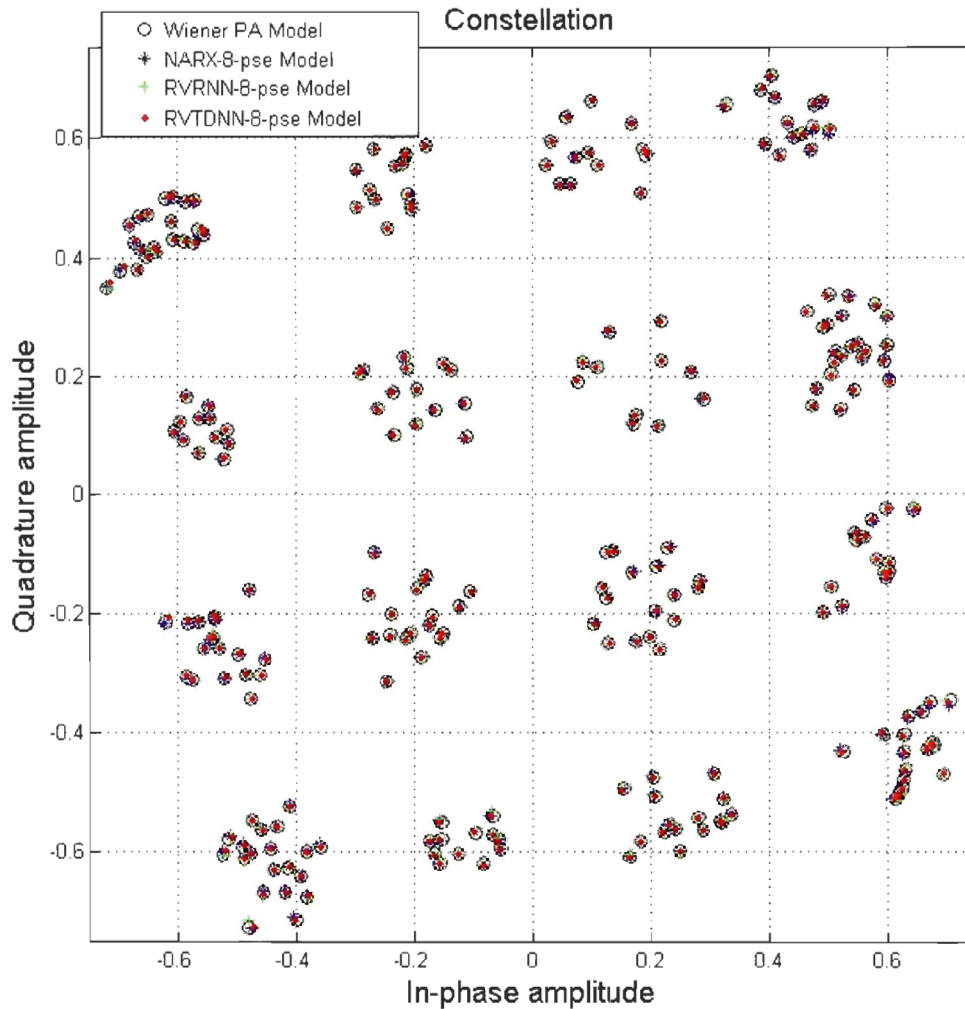


Figure 65: Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse.

Les résultats de simulation de la conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse représentées sur la figure 66 sont quantitativement caractérisés par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse ont respectivement comme paramètre MSE de la conversion AM/AM 0.581×10^{-5} , 1.116×10^{-5} et 0.776×10^{-5} par rapport au modèle de référence de Wiener.

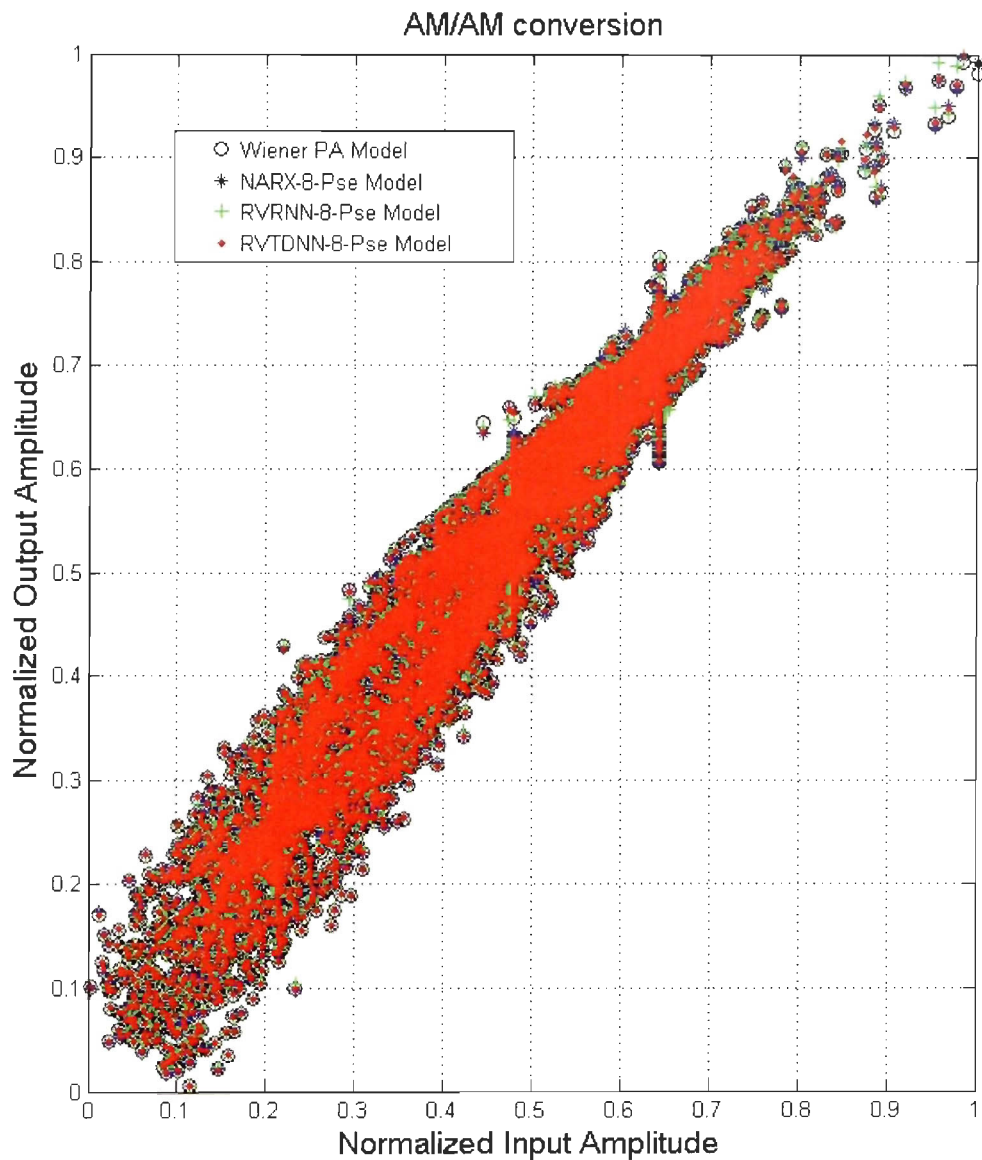


Figure 66: Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.

Le calcul du paramètre MSE de la conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse représentées sur la figure 67, est utilisé comme méthode d'analyse quantitative. Les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse ont respectivement comme paramètre MSE de la conversion AM/PM 0.100, 0.080 et 0.029 par rapport au modèle de référence de Wiener.

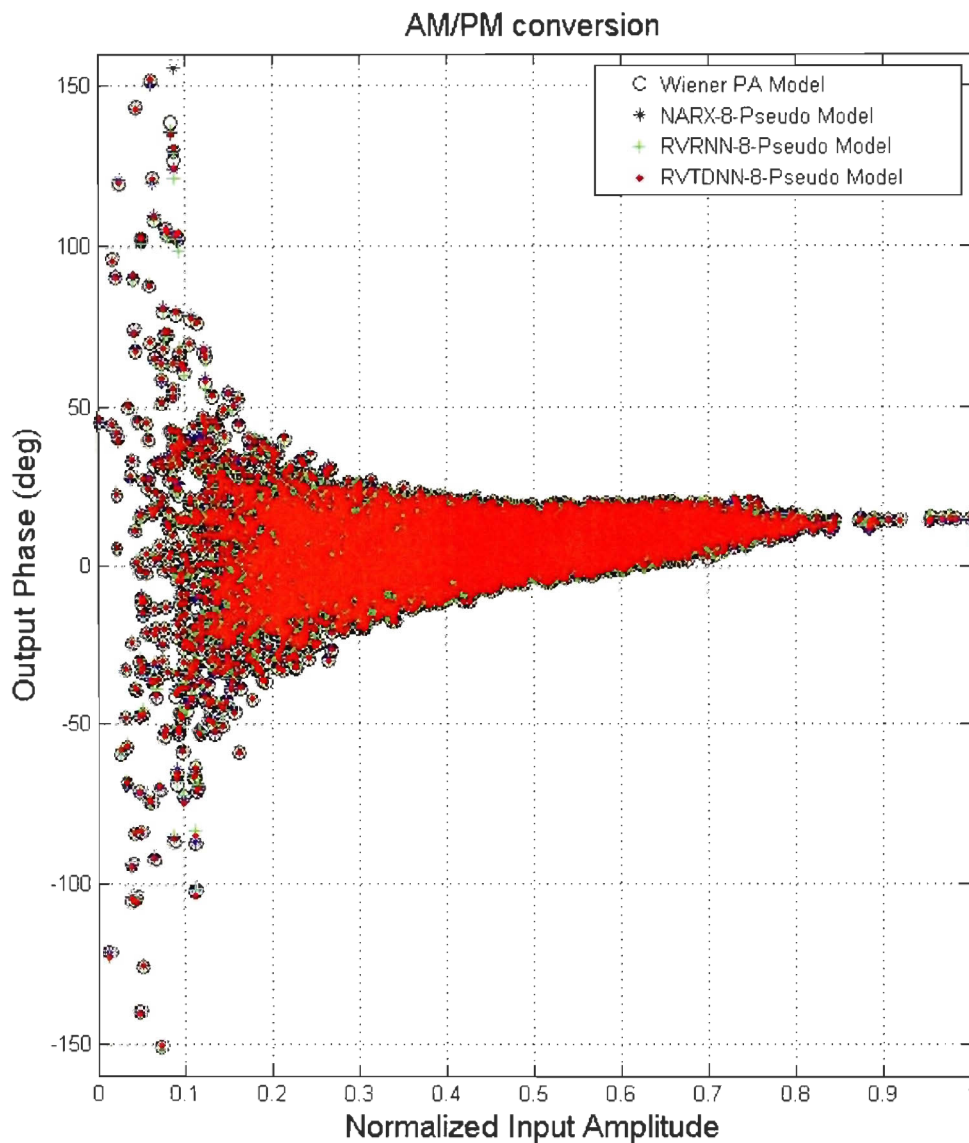


Figure 67: Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse.

La comparaison des caractéristiques des composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse, représentée sur la figure 68, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 0.485×10^{-5} , 0.741×10^{-5} et 0.571×10^{-5} et en quadrature (Q) 1.436×10^{-5} , 1.507×10^{-5} et 0.501×10^{-5} par rapport au modèle de référence de Wiener.

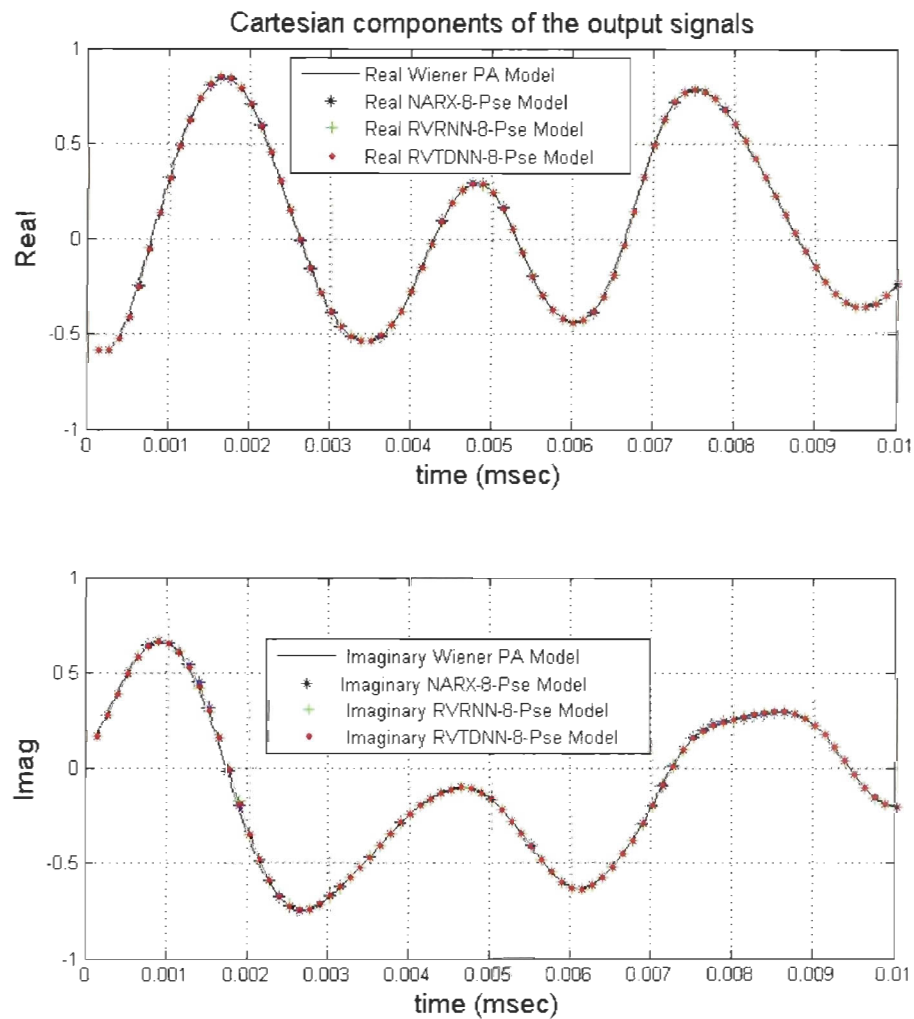


Figure 68: Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse.

La comparaison des résultats de simulation des spectres du signal d'entrée et des signaux de sortie du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse

RVRNN-8-pse et RVTDDN-8-pse représentés sur la figure 69, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse ont respectivement comme paramètre MSE des spectres 0.960×10^{-5} , 1.124×10^{-5} et 0.536×10^{-5} par rapport au modèle de référence de Wiener.

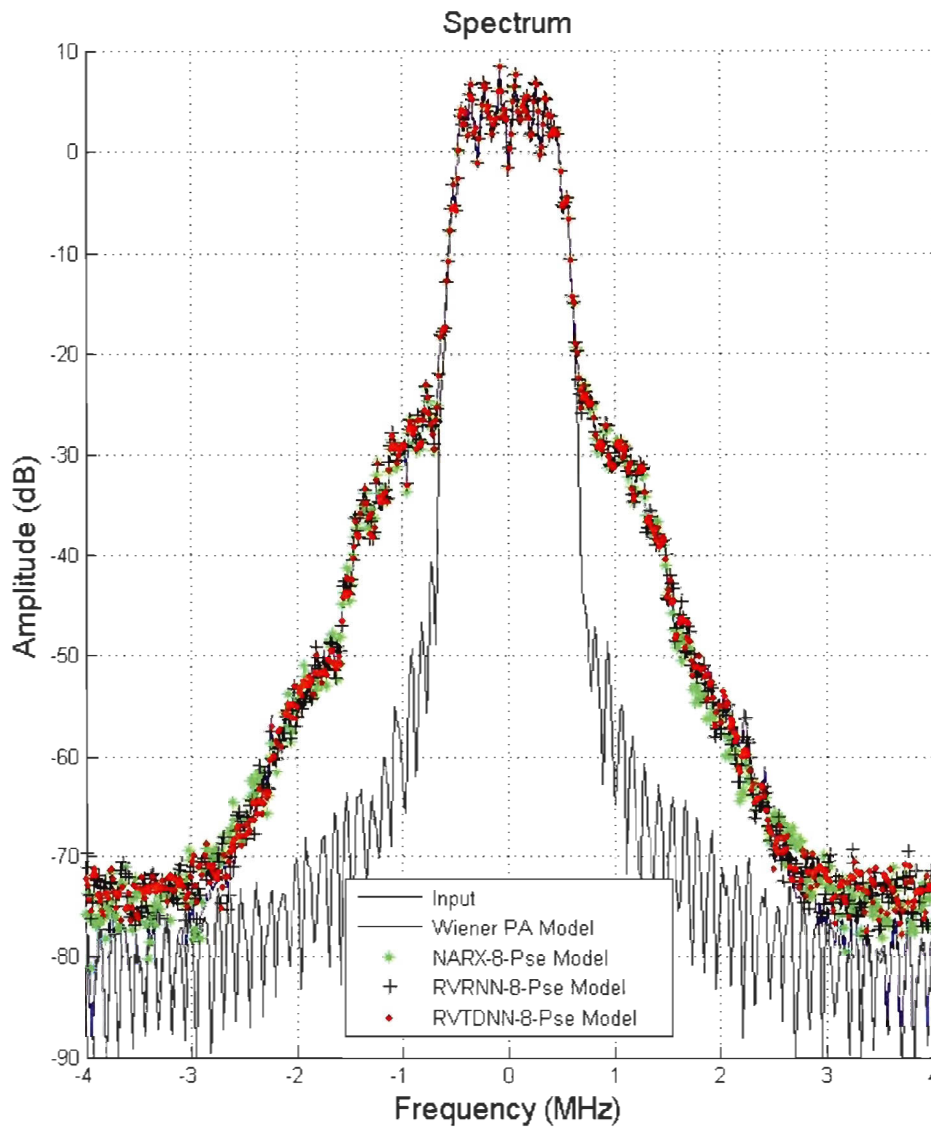


Figure 69: Spectres du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse.

Les convergences des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse mesurées par les paramètres MSE de leurs sorties relativement à celle du modèle de

référence de Wiener sont représentées sur la figure 70. Les architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse ont respectivement comme paramètre MSE à la 2^{ième} trame environ 0.0125, 0.0105 et 0.0096.

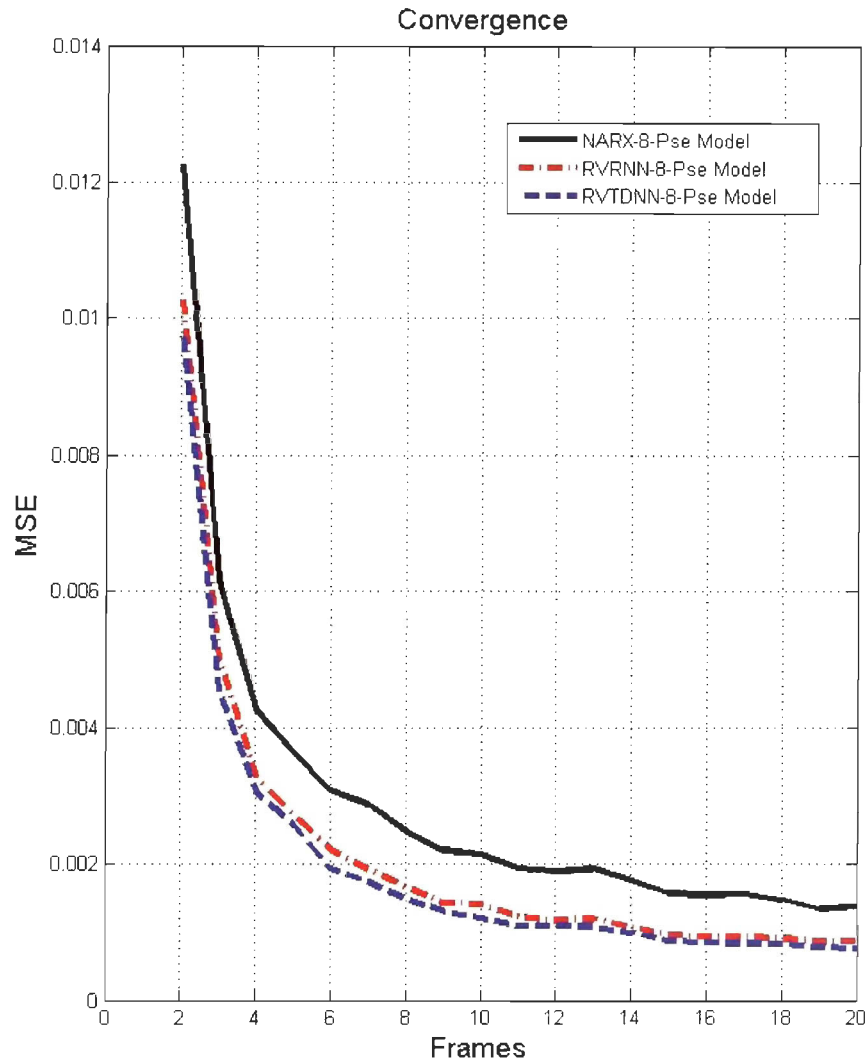


Figure 70: Convergence des architectures de modélisation NARX-8-pse, RVRNN-8-pse et RVTDDN-8-pse mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.

À l'issue de l'analyse des résultats, l'architecture de modélisation RVTDDN-8-pse modélise légèrement mieux le modèle de référence de Wiener que celle du réseau NARX-8-pse qui elle-même le modélise mieux que le réseau RVRNN-8-pse. Effectivement, l'architecture de modélisation NARX-8-pse a 33.199 % du paramètre EVM, valeur qui est plus proche du

paramètre EVM du modèle de référence de Wiener (33.205 %) que celles des architectures de modélisation RVTDDN-8-pse et RVRNN-8-pse qui ont respectivement 33.198 % et 33.192 % pour les caractéristiques quantitatives des diagrammes de constellation par rapport au signal d'entrée. Pour les conversions AM/AM, le paramètre MSE de l'architecture de modélisation NARX-8-pse est 0.581×10^{-5} qui est plus petit que celui du réseau RVTDDN-8-pse 0.776×10^{-5} qui est lui-même plus petit que celui du réseau RVRNN-8-pse 1.116×10^{-5} . Pour les conversions AM/PM, le paramètre MSE de l'architecture de modélisation RVTDDN-8-pse a 0.029 qui est plus petit que celui du réseau RVRNN-8-pse 0.080 qui est lui-même plus petit que celui du réseau NARX-8-pse 0.100 par rapport au modèle de référence de Wiener. Le calcul du paramètre MSE pour la caractérisation du composant cartésien I montre que le réseau NARX-8-pse (0.485×10^{-5}) a une valeur inférieure à celle du réseau RVTDDN-8-pse (0.571×10^{-5}) et du réseau RVRNN-8-pse (0.741×10^{-5}) par rapport au modèle de référence de Wiener. Par contre, le calcul du paramètre MSE pour la caractérisation du composant cartésien Q montre que le réseau RVTDDN-8-pse (0.501×10^{-5}) a une valeur inférieure à celle du réseau NARX-8-pse (1.436×10^{-5}) et du réseau RVRNN-8-pse (1.507×10^{-5}) par rapport au modèle de référence de Wiener. Pour le spectre, le réseau RVTDDN-8-pse (0.536×10^{-5}) a un paramètre MSE inférieur au réseau NARX-8-pse (0.960×10^{-5}) dont le paramètre MSE est inférieur au réseau RVRNN-8-pse (1.124×10^{-5}) par rapport au modèle de référence de Wiener. Le paramètre MSE de la vitesse de convergence à la 2^{ème} trame du réseau RVTDDN-6-pip (0.0096) est inférieur à celui du réseau RVRNN-6-pip (0.0105) qui est lui-même inférieur à celui du réseau NARX-6-pip (0.0125) par rapport au modèle de référence de Wiener. Il convient de dire que le réseau RVTDDN-6-pip converge plus vite que le réseau RVRNN-6-pip qui lui-même converge plus vite que le réseau NARX-6-pip (voir figure 70).

2.8.5 Performances des architectures de modélisation RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip

La figure 71 représente les diagrammes de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip. La caractérisation quantitative est faite par calcul du paramètre EVM en pourcent. Le modèle de référence de Wiener a 33.205 % du paramètre EVM tandis que les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip ont respectivement 33.142 %, 33.152 % et 33.214 % de paramètre EVM.

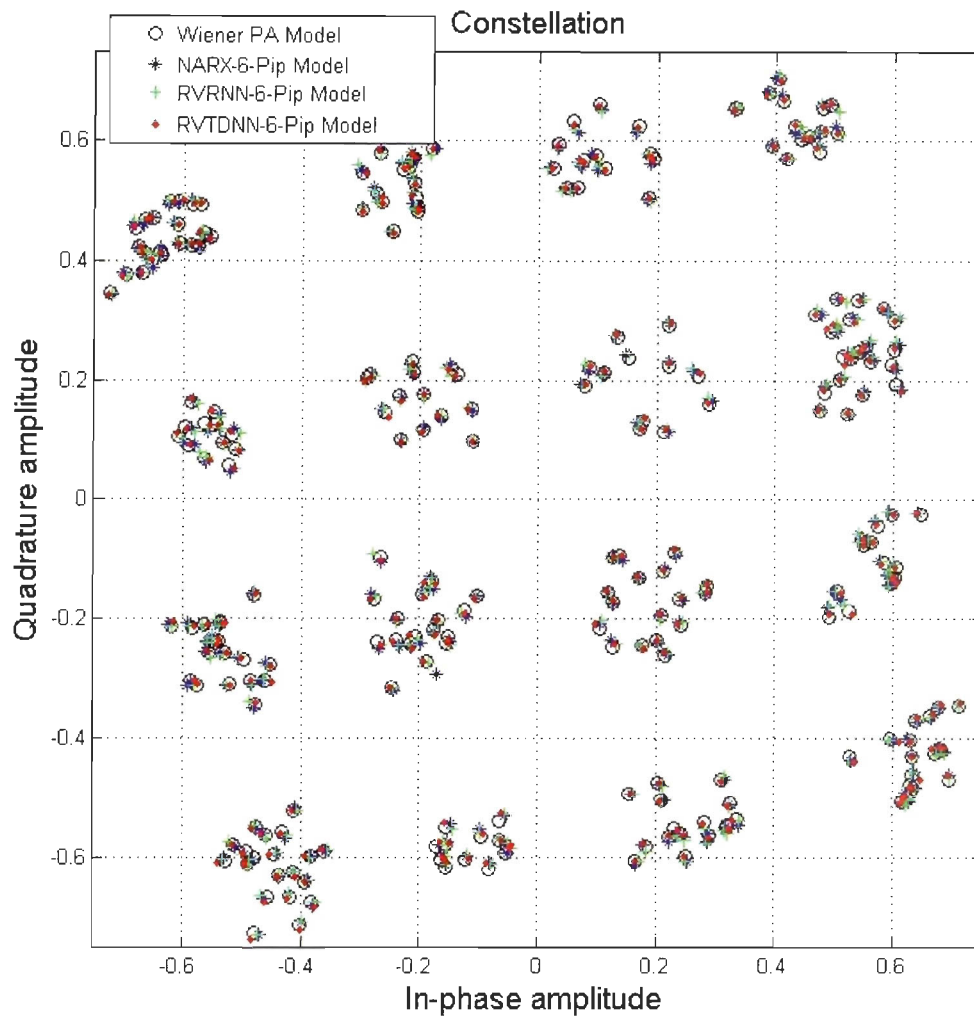


Figure 71: Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.

Les résultats de simulation de la conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip représentés sur la figure 72 sont quantitativement caractérisés par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip ont respectivement comme paramètre MSE de la conversion AM/AM 3.223×10^{-5} , 4.753×10^{-5} et 7288.781×10^{-5} par rapport au modèle de référence de Wiener.

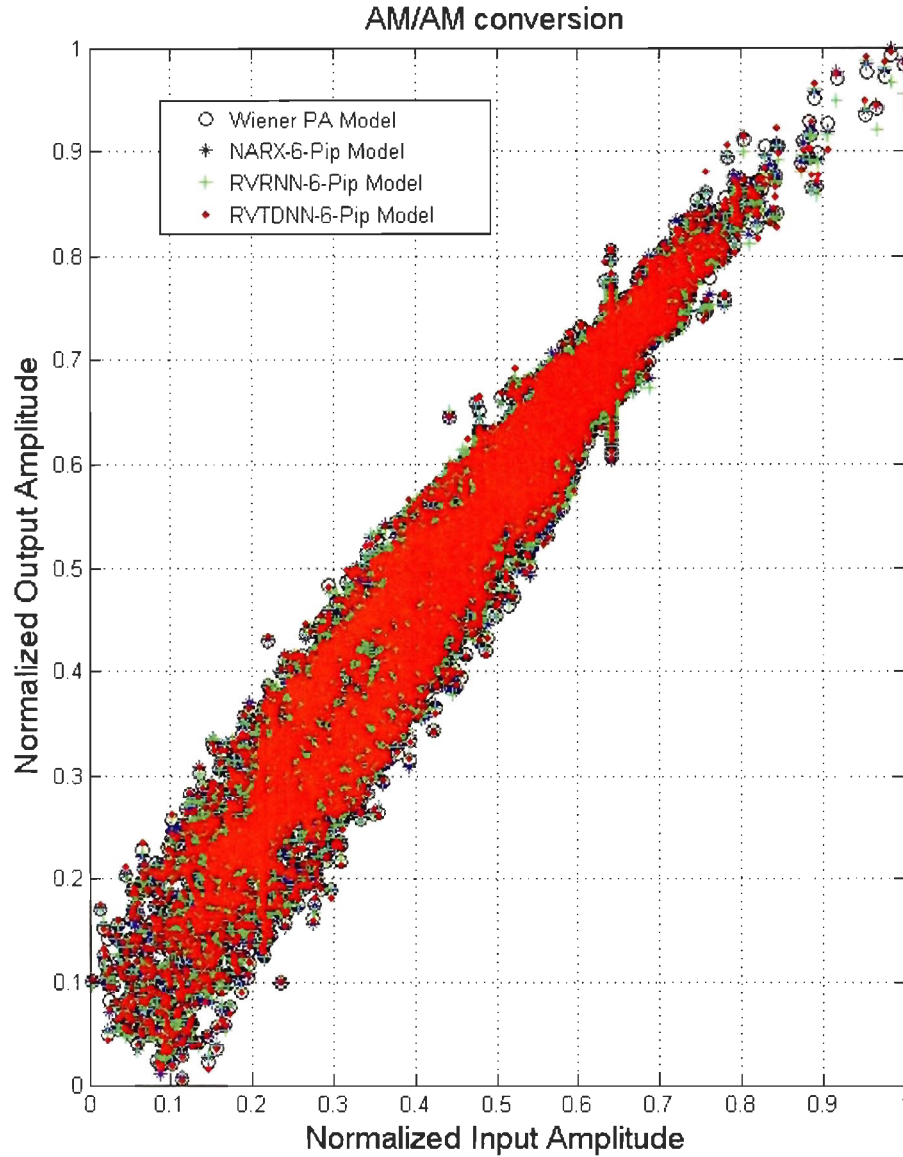


Figure 72: Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.

Le calcul du paramètre MSE de la conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip représentées sur la figure 73, est utilisé comme méthode d'analyse quantitative. Les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip ont respectivement comme paramètre MSE de la conversion AM/PM 0.560, 0.932 et 2.046 par rapport au modèle de référence de Wiener.

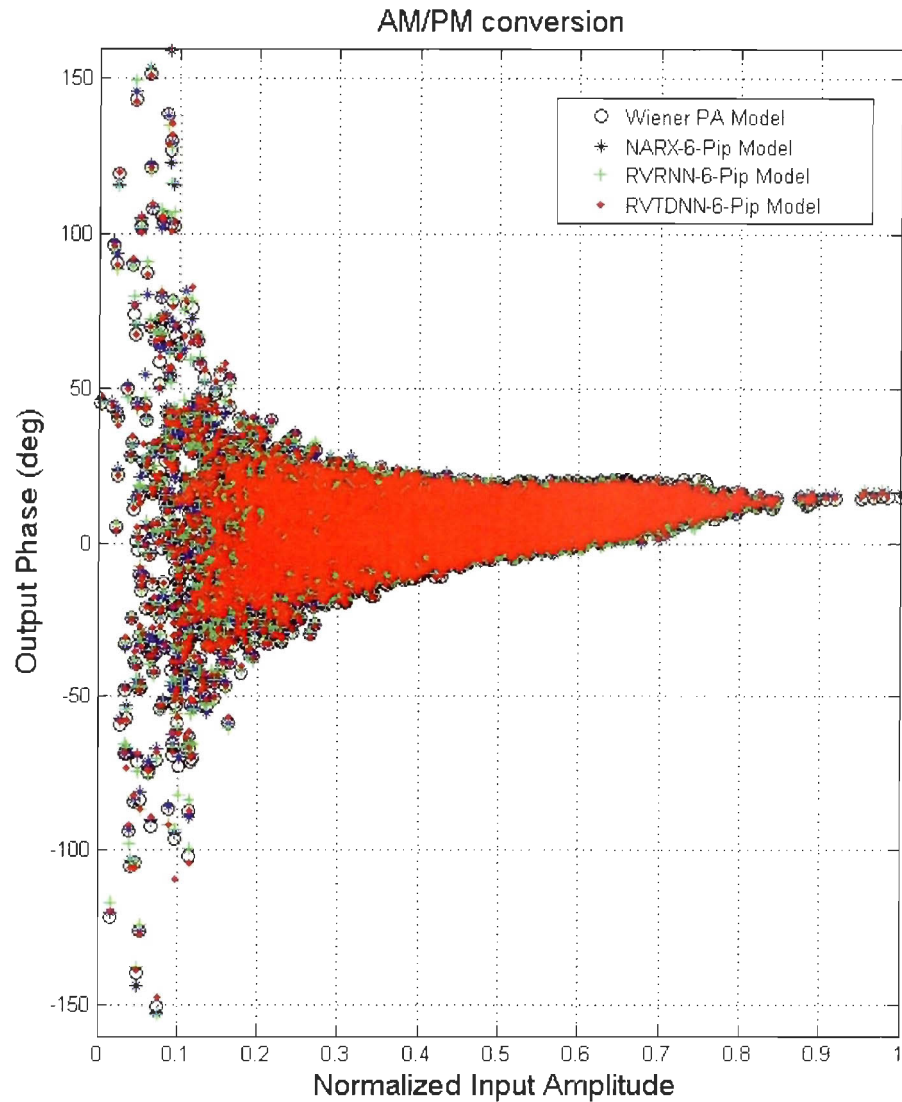


Figure 73: Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.

La comparaison des caractéristiques des composants cartésiens I et Q du modèle de référence de Wiener des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip, représentée sur la figure 74, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 2.983×10^{-5} , 3.796×10^{-5} et 51453.434×10^{-5} et en quadrature (Q) 5.975×10^{-5} , 11.464×10^{-5} et 26916.484×10^{-5} par rapport au modèle de référence de Wiener.

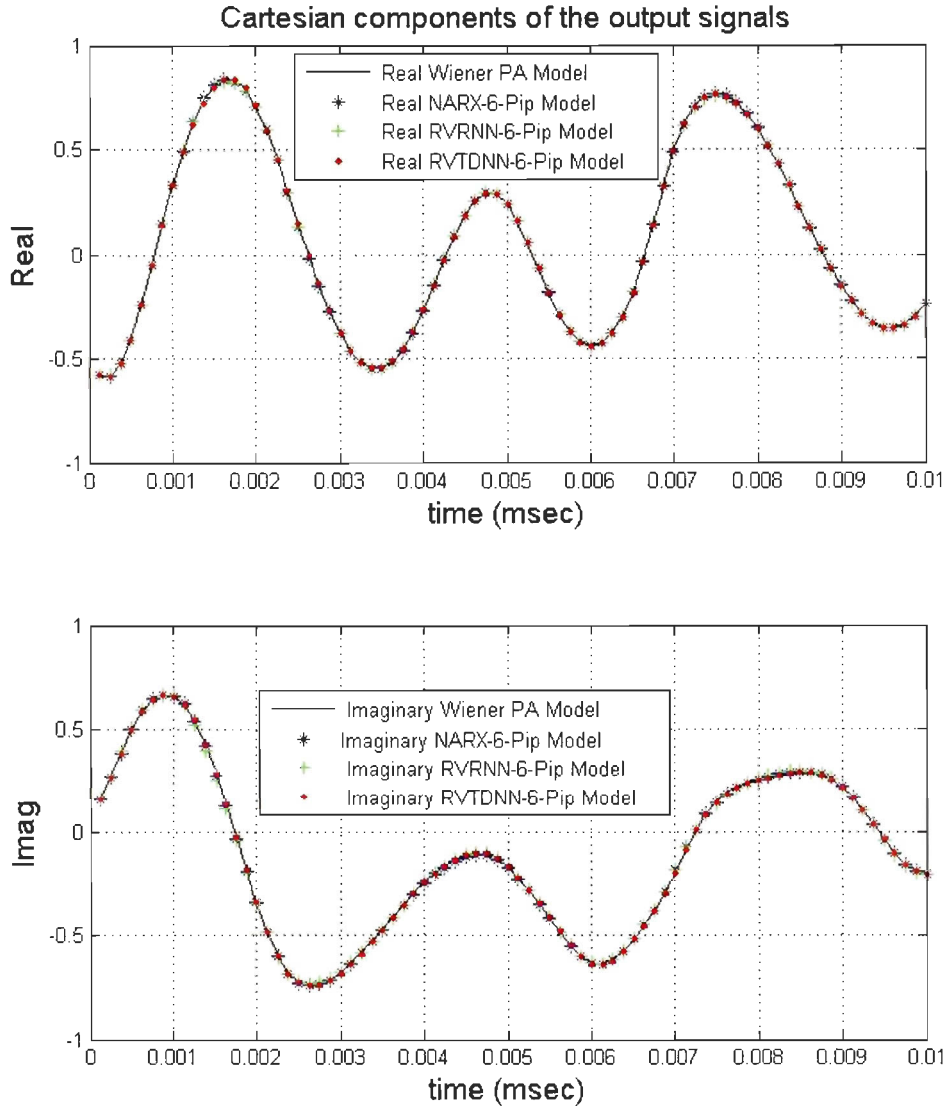


Figure 74: Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.

La comparaison des résultats de simulation des spectres du signal d'entrée et des signaux de sortie du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip représentés sur la figure 75, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip ont respectivement comme paramètre MSE des spectres 4.479×10^{-5} , 7.630×10^{-5} et 39184.959×10^{-5} par rapport au modèle de référence de Wiener.

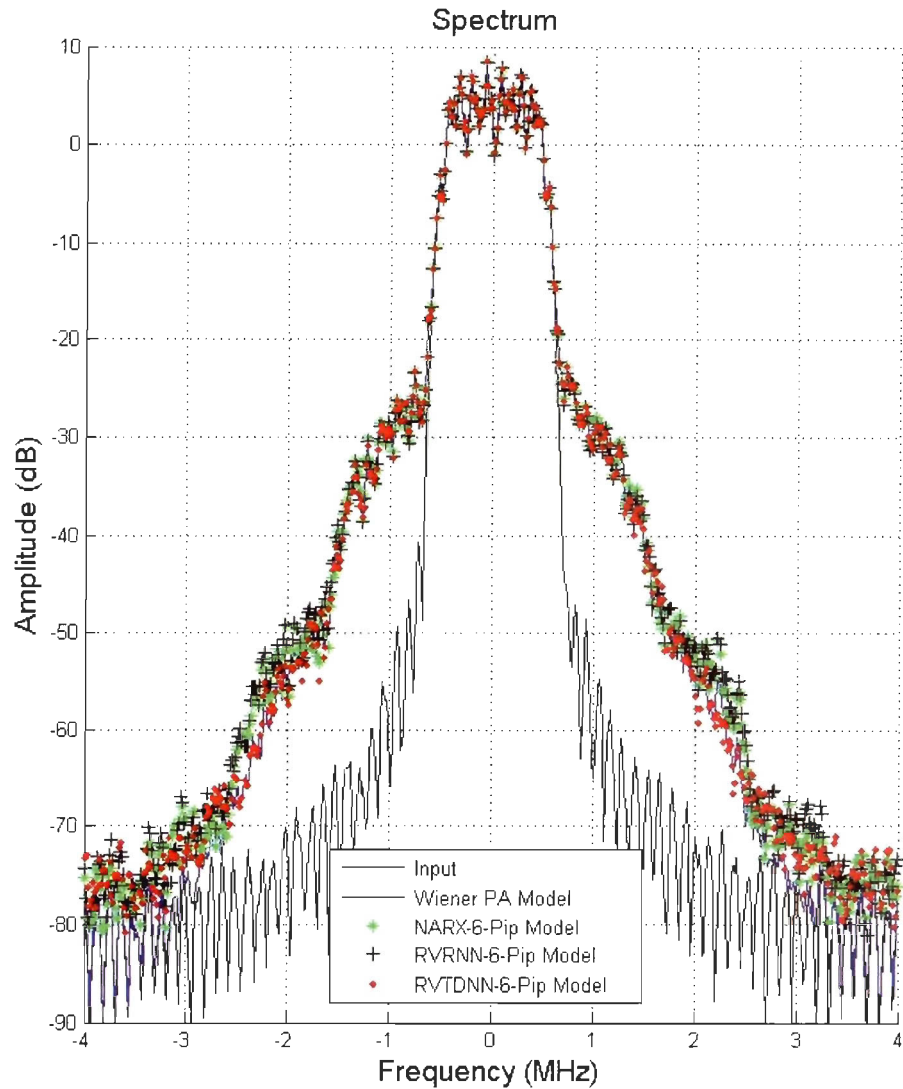


Figure 75: Spectre du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip.

Les convergences des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip mesurées par les paramètres MSE de leurs sorties relativement à celle du modèle de référence de Wiener sont représentées sur la figure 76. Les architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0460, 0.0285 et 0.0135.

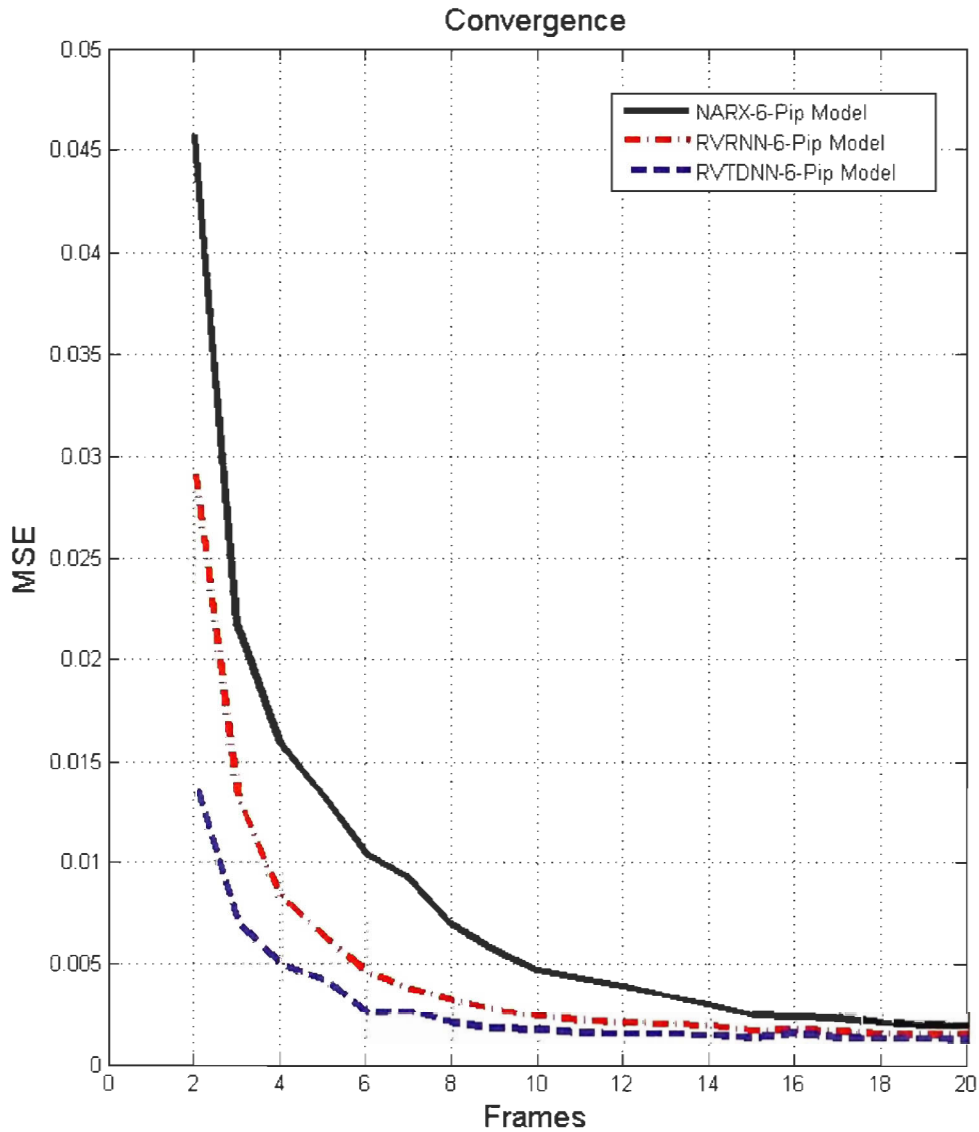


Figure 76: Convergence des architectures de modélisation NARX-6-pip, RVRNN-6-pip et RVDNN-6-pip mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.

Après l'analyse des résultats de simulation des différentes courbes, le constat est que l'architecture de modélisation NARX-6-pip modélise mieux le modèle de référence de Wiener que celle du réseau RVRNN-6-pip qui elle-même modélise mieux que le réseau RVDNN-6-pip. Pour les conversions AM/AM et AM/PM, le paramètre MSE de l'architecture de modélisation NARX-6-pip a respectivement 3.223×10^{-5} et 0.560 valeurs qui sont respectivement plus petites

que celles du réseau RVRNN-6-pip, 4.753×10^{-5} et 0.932, qui sont elles-mêmes respectivement plus petites que celles du réseau RVTDDN-6-pip, 7288.781×10^{-5} et 2.046, par rapport au modèle de référence de Wiener. Le calcul du paramètre MSE pour la caractérisation des composants cartésiens I et Q, montre que le réseau NARX-6-pip a une valeur inférieure à celle du RVRNN-6-pip et du RVTDDN-6-pip par rapport au modèle de référence de Wiener. De même, pour le spectre le réseau NARX-6-pip a un paramètre MSE inférieur au réseau RVRNN-6-pip dont le paramètre MSE est inférieur au réseau RVTDDN-6-pip par rapport au modèle de référence de Wiener. Le paramètre MSE de la vitesse de convergence à la 2^{ième} trame du réseau RVTDDN-6-pip (0.0460) est inférieur à celui du réseau RVRNN-6-pip (0.0285) qui est lui-même inférieur à celui du réseau NARX-6-pip (0.0135) par rapport au modèle de référence de Wiener. Il convient de dire que le réseau RVTDDN-6-pip converge plus vite que le réseau RVRNN-6-pip qui lui même converge plus vite que le réseau NARX-6-pip (voir figure 76).

2.8.6 Performances des architectures de modélisation RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip

La figure 77 représente les diagrammes de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip. La caractérisation quantitative est faite par calcul du paramètre EVM en pourcent. Le modèle de référence de Wiener a 33.205 % du paramètre EVM tandis que les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip ont respectivement 33.179 %, 33.198 % et 33.556 % de paramètre EVM.

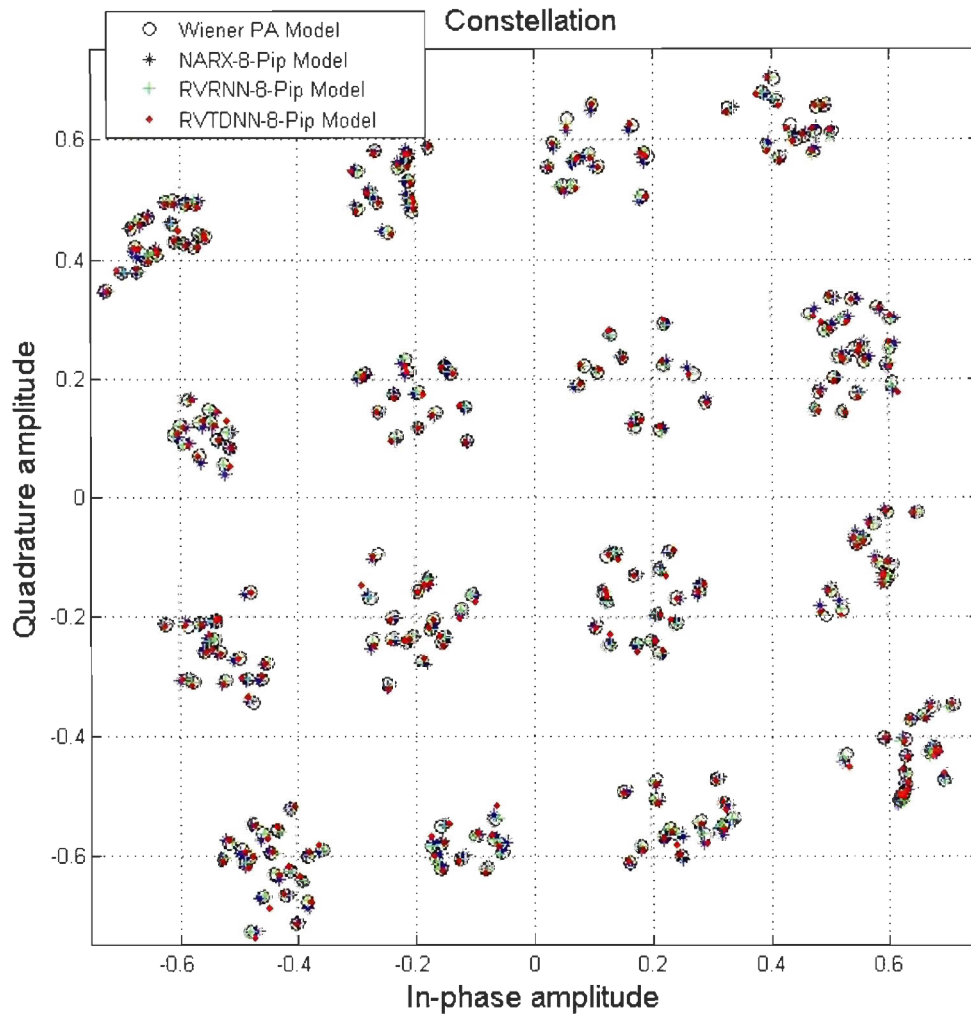


Figure 77: Diagramme de constellation du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip.

Les résultats de simulation de la conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip représentées sur la figure 78 sont quantitativement caractérisés par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip ont respectivement comme paramètre MSE de la conversion AM/AM 4.695×10^{-5} , 0.926×10^{-5} et 2.849×10^{-5} par rapport au modèle de référence de Wiener.

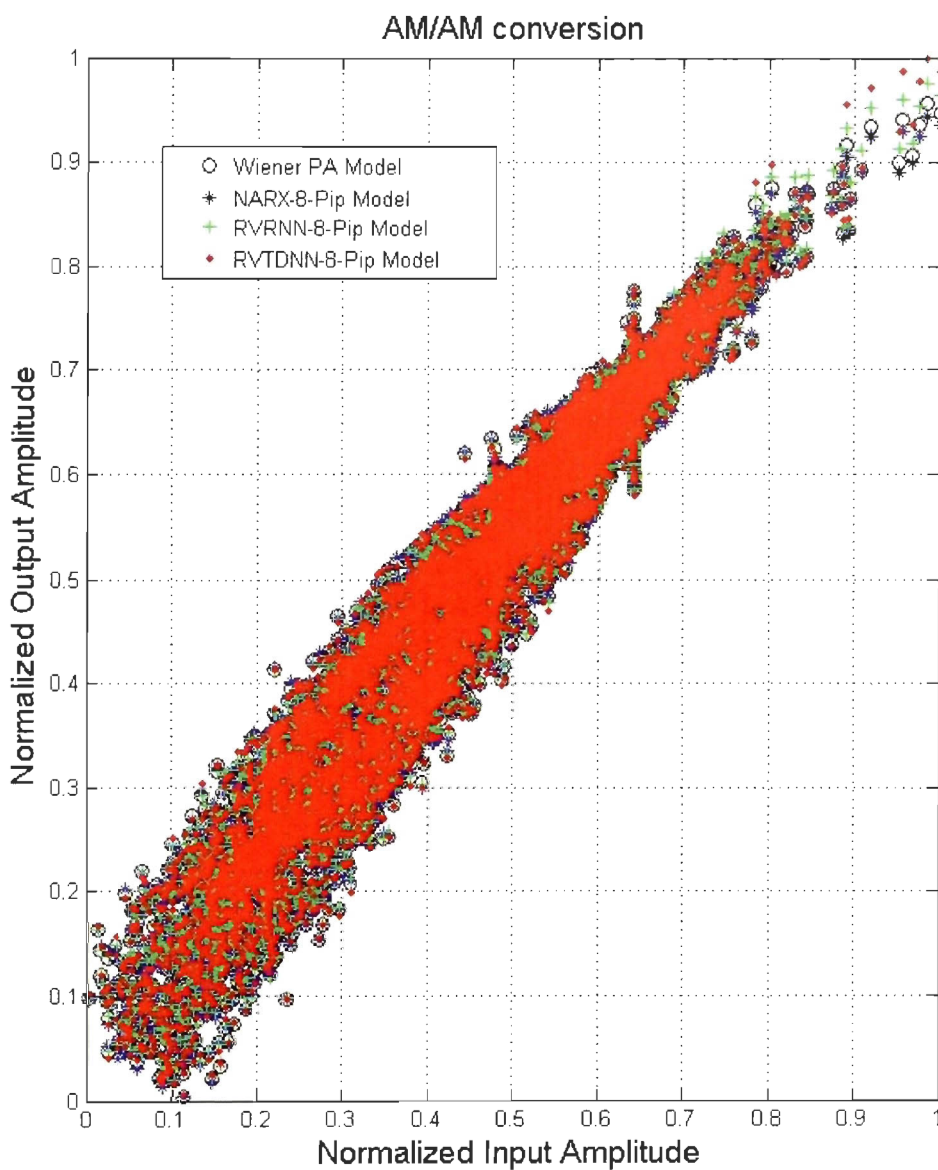


Figure 78: Conversion AM/AM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.

Le calcul du paramètre MSE de la conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip représentées sur la figure 79, est utilisé comme méthode d'analyse quantitative. Les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip ont respectivement comme paramètre MSE de la conversion AM/PM 0.811, 0.068 et 0.328 par rapport au modèle de référence de Wiener.

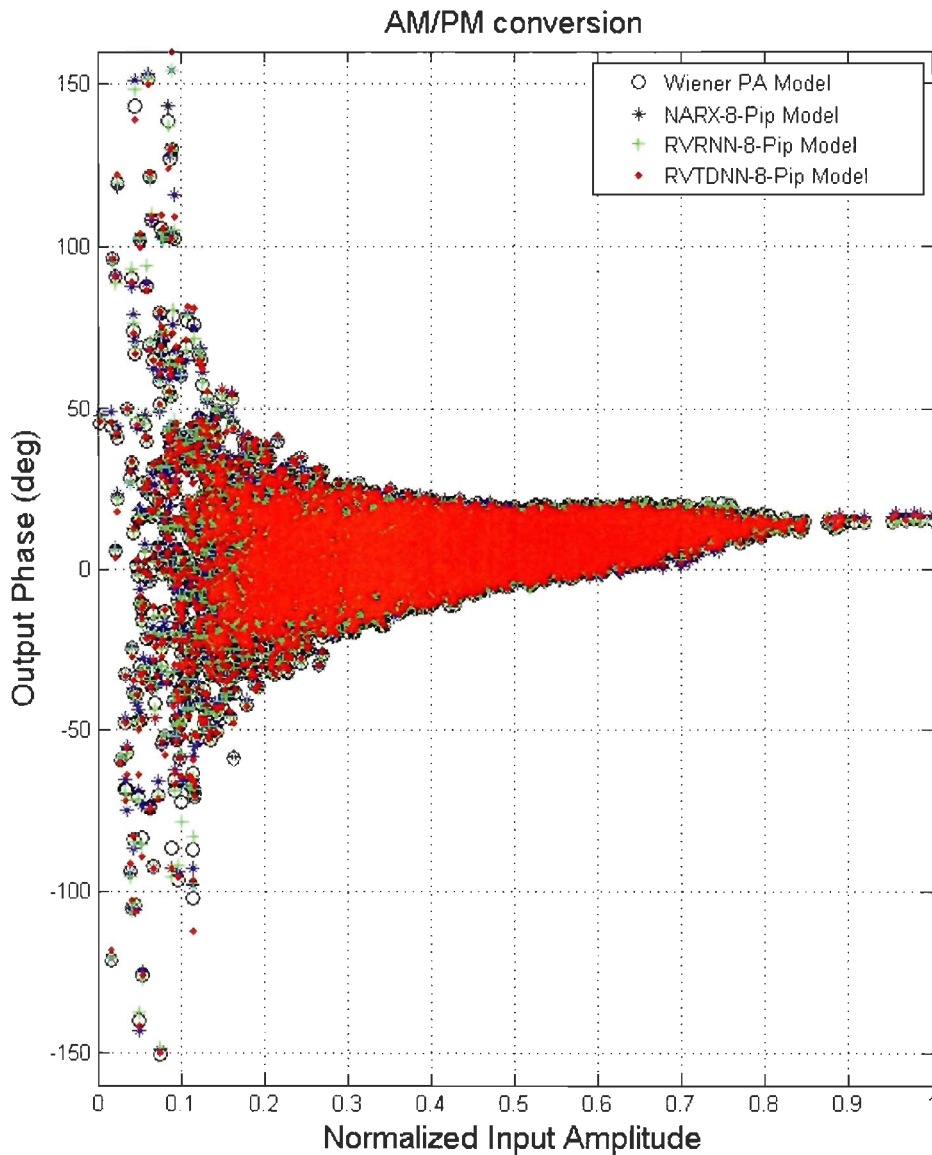


Figure 79: Conversion AM/PM du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip.

La comparaison des caractéristiques des composants cartésiens I et Q du modèle de référence de Wiener des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip, représentée sur la figure 80, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 4.603×10^{-5} , 1.169×10^{-5} et 3.525×10^{-5} et en quadrature (Q) 9.629×10^{-5} , 0.544×10^{-5} et 2.413×10^{-5} par rapport au modèle de référence de Wiener.

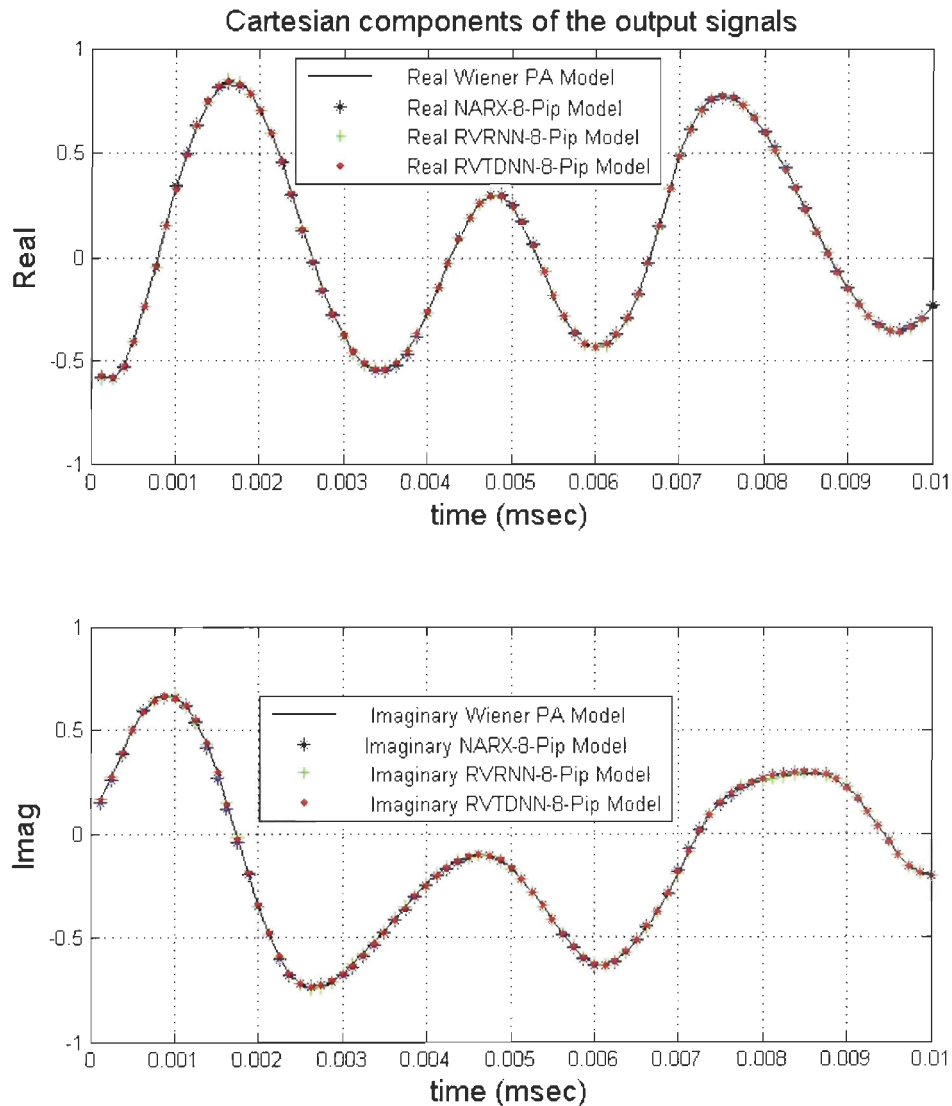


Figure 80: Composants cartésiens I et Q du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.

La comparaison des résultats de simulation des spectres du signal d'entrée et des signaux de sortie du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip représentés sur la figure 81, est aussi faite par calcul du paramètre MSE. Les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip ont respectivement comme paramètre MSE des spectres 7.116×10^{-5} , 0.857×10^{-5} et 2.969×10^{-5} par rapport au modèle de référence de Wiener.

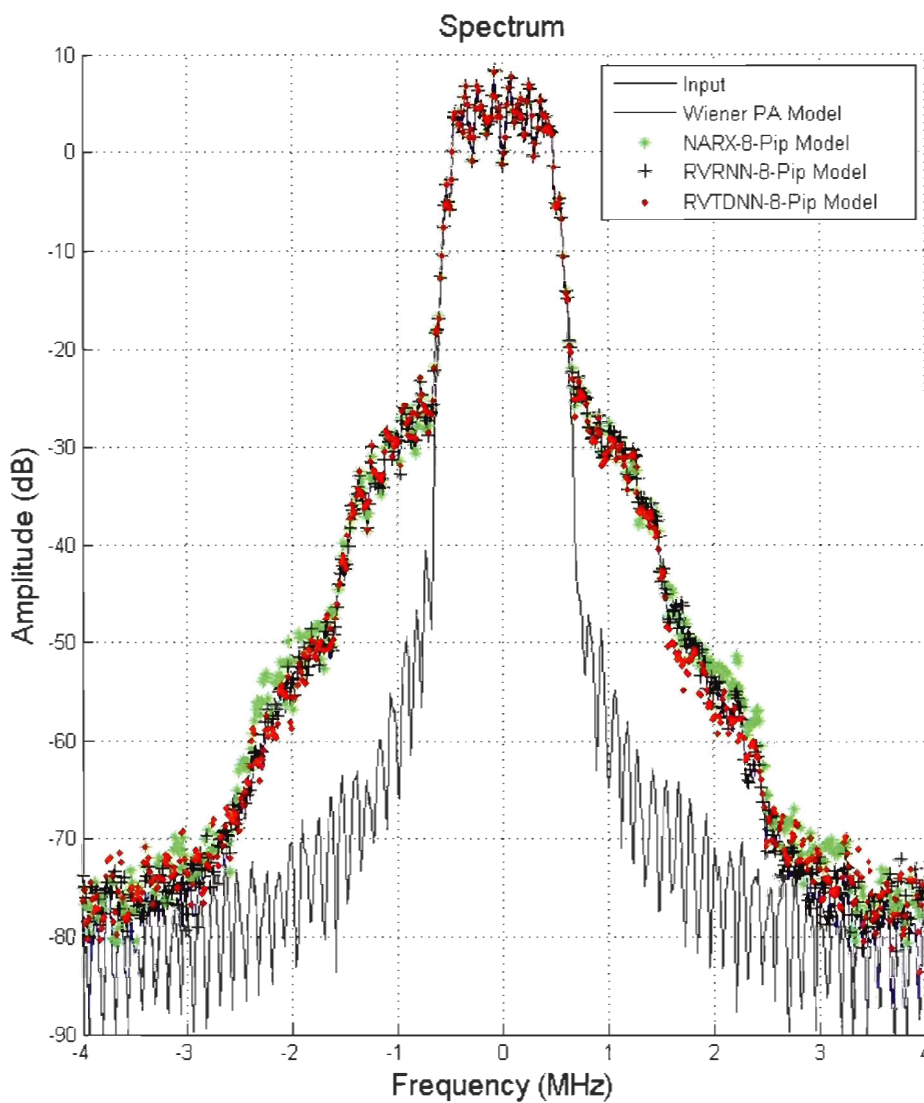


Figure 81: Spectre du signal d'entrée et des sorties du modèle de référence de Wiener et des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip.

Les convergences des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip mesurées par les paramètres MSE de leurs sorties relativement à celle du modèle de référence de Wiener sont représentées sur la figure 82. Les architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.059, 0.064 et 0.016.

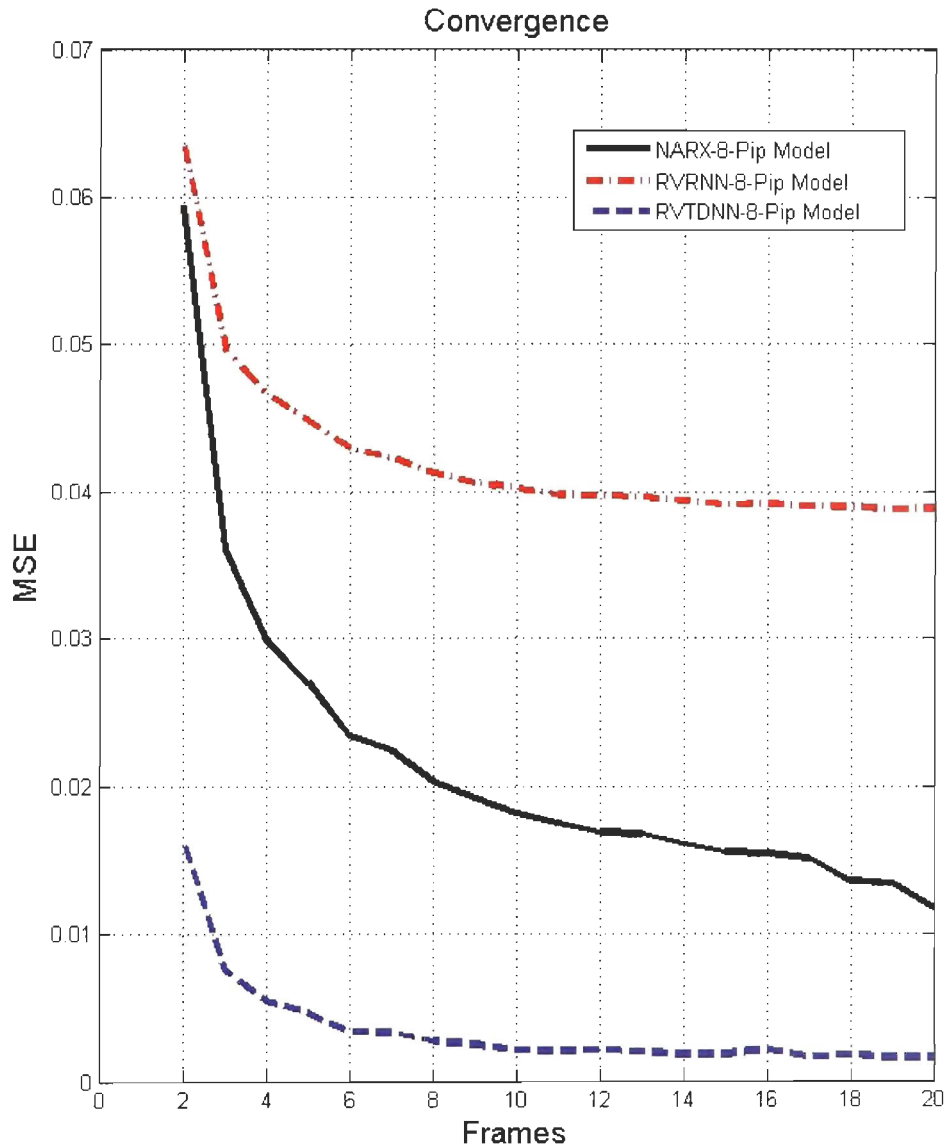


Figure 82: Convergence des architectures de modélisation NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip mesurée par les paramètres MSE de leurs sorties relativement à celle du modèle de référence.

De l'analyse des résultats de simulation des différentes courbes servant à la validation des résultats, découle que l'architecture de modélisation RVRNN-8-pip modélise mieux le modèle de référence de Wiener que celle du réseau RVTDDNN-8-pip qui elle-même modélise mieux que le réseau NARX-8-pip. Pour les conversions AM/AM, le paramètre MSE de l'architecture de modélisation RVRNN-8-pip a $9.269e^{-006}$ qui est plus petit que celle du réseau RVTDDNN-8-pip (2.849×10^{-5}) qui est elle-même plus petite que celle du réseau NARX-8-pip (4.695×10^{-5}) par rapport au modèle de référence de Wiener. Pour les conversions AM/PM, le paramètre MSE de l'architecture de modélisation RVTDDNN-8-pip a 0.328 qui est plus petite que celle du réseau RVRNN-8-pip 0.068 qui est elle-même plus petite que celle du réseau NARX-8-pip 0.811 par rapport au modèle de référence de Wiener. Le calcul du paramètre MSE pour la caractérisation des composants cartésiens I montre que le réseau RVRNN-8-pip ($1,169 \times 10^{-5}$) a une valeur inférieure à celle du réseau RVTDDNN-8-pip (3.525×10^{-5}) et du réseau NARX-8-pip (4.603×10^{-5}) par rapport au modèle de référence de Wiener. Le calcul du paramètre MSE pour la caractérisation des composants cartésiens Q montre également que le réseau RVRNN-8-pip (0.544×10^{-5}) a une valeur inférieure à celle du réseau RVTDDNN-8-pip (2.413×10^{-5}) et du réseau NARX-8-pip (9.629×10^{-5}) par rapport au modèle de référence de Wiener. Pour le spectre, le réseau RVRNN-8-pip (0.857×10^{-5}) a un paramètre MSE inférieur au réseau RVTDDNN-8-pip (2.969×10^{-5}) dont le paramètre MSE est inférieur au réseau NARX-8-pip (7.116×10^{-5}) par rapport au modèle de référence de Wiener. Le paramètre MSE relatif à la vitesse de convergence du réseau RVTDDNN-8-pip (0.016) est inférieur à celui du réseau RVRNN-8-pip (0.064) qui est lui-même inférieur à celui du réseau NARX-8-pip (0.059) par rapport au modèle de référence de Wiener. Il convient de dire que le réseau RVTDDNN-8-pip converge plus vite que le réseau NARX-8-pip qui lui-même converge plus vite que le réseau RVRNN-8-pip. Les résultats de modélisation par le réseau RVTDDNN- N_1 -pse, le réseau RVRNN- N_1 -pse, le réseau NARX- N_1 -pse, le réseau NARX- N_1 -pip, le réseau RVRNN- N_1 -pip et le réseau RVTDDNN- N_1 -pip montrent qu'ils modélisent fidèlement le modèle de référence de Wiener.

Tableau 3: Résultats de simulation en termes de constellation, caractéristiques AM/AM et AM/PM, composants cartésiens I et Q et spectre des différentes architectures de modélisation.

Architecture de Modélisation	Constellations	Caractéristiques		Composants Cartésiens		Spectre
	EVM (%)	MSE d'AM/AM ($\times 10^{-5}$)	MSE d'AM/PM	MSE de Phase (I) ($\times 10^{-5}$)	MSE de Quadrature (Q) ($\times 10^{-5}$)	MSE ($\times 10^{-5}$)
Modèle de référence	33.205	-	-	-	-	-
RVTDNN-6-pse	33.181	1.675	0.150	1.040	2.118	1.579
RVRNN-6-pse	33.199	0.670	0.125	0.459	0.186	0.322
NARX-6-pse	33.197	1.379	0.141	1.175	1.924	1.549
RVTDNN-8-pse	33.198	0.776	0.029	0.571	0.501	0.536
RVRNN-8-pse	33.192	1.116	0.080	0.741	1.507	1.124
NARX-8-pse	33.199	0.581	0.100	0.485	1.436	0.960
RVTDNN-6-pip	33.214	7288.781	2.046	51453.434	26916.484	39184.959
RVRNN-6-pip	33.152	4.753	0.932	3.796	11.464	7.630
NARX-6-pip	33.142	3.223	0.560	2.983	5.975	4.479
RVTDNN-8-pip	33.556	2.840	0.328	3.525	2.413	2.969
RVRNN-8-pip	33.198	0.926	0.068	1.169	0.544	0.857
NARX-8-pip	33.179	4.695	0.811	4.603	9.629	7.116

Dépendamment du fait que les architectures de modélisation soit pseudo-conventionnelles ou pipelinées avec 6 ou 8 neurones à la couche cachée, elles modélisent plus ou moins mieux les distorsions liées à la non-linéarité et aux effets mémoire des PAs. Du fait que ces modèles adaptatifs modélisent les PA avec précision, ils peuvent donc être utilisés pour la prédistorsion adaptative. L'évaluation des ressources montre également que les architectures de modélisation NARX- N_1 -pip, RVRNN- N_1 -pip et RVTDNN- N_1 -pip sont implantables dans la carte FPGA Virtex-6 XC6VLX240T-1FFG1156 ML605. Tel que représenté dans le tableau 3, pour les architectures de modélisation pseudo-conventionnelles, le réseau RVRNN-6-pse et le réseau NARX-8-pse modélisent mieux les distorsions à l'intérieur de la bande (EVM) dues au modèle de référence de Wiener que toutes les autres architectures de modélisation pseudo-conventionnelles, car elles ont toutes les deux 33.199 % du paramètre EVM, qui est la valeur la plus proche du paramètre EVM du modèle de référence de Wiener (33.205 %). Elles sont suivies de près par le réseau RVTDNN-8-pse qui a 33.198 % de paramètre EVM. De même, pour les caractéristiques AM/AM, le réseau NARX-8-pse a le paramètre MSE le plus bas (0.581×10^{-5}) suivi du réseau RVRNN-6-pse (0.670×10^{-5}) et du réseau RVTDNN-8-pse (0.776×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau RVTDNN-8-pse a le paramètre MSE le plus bas (0.029) suivi du réseau RVRNN-8-pse (0.080) et du réseau NARX-8-pse (0.100). Pour la modélisation du composant cartésien en phase I, le réseau RVRNN-6-pse (0.459×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-8-pse (0.485×10^{-5}) qui est lui-même meilleur que le réseau RVTDNN-8-pse (0.571×10^{-5}). Pour la modélisation du composant cartésien en quadrature Q, le réseau RVRNN-6-pse a un paramètre MSE de 0.186×10^{-5} qui est meilleur que ceux du réseau RVTDNN-8-pse (0.501×10^{-5}) et du réseau NARX-8-pse (1.436×10^{-5}). Par conséquent, le réseau RVRNN-6-pse, avec un paramètre MSE de (0.322×10^{-5}), modélise mieux le spectre du modèle de référence de Wiener que le réseau RVTDNN-8-pse (0.536×10^{-5}), qui lui-même modélise mieux le réseau NARX-8-pse (0.960×10^{-5}).

Également, dans le tableau 3, pour les architectures de modélisation avec pipeline, le réseau RVRNN-8-pip modélise mieux les distorsions à l'intérieur de la bande (EVM) que toutes les autres architectures de modélisation avec pipeline, car elle a 33.198 % de paramètre EVM, qui est la valeur la plus proche le paramètre EVM du modèle de référence de Wiener (33.205 %). De même, pour les caractéristiques AM/AM, le réseau RVRNN-8-pip a le paramètre MSE le plus bas (0.926×10^{-5}) suivi du réseau RVTDNN-8-pip (2.849×10^{-5}). Par contre, pour les

caractéristiques AM/PM, le réseau RVRNN-8-pip a le paramètre MSE le plus bas (0.068) suivi du RVTDDN-8-pip 0.328. Pour la modélisation du composant cartésien en phase I, le réseau RVRNN-8-pip (1.169×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-6-pip (2.983×10^{-5}). Aussi, pour la modélisation du composant cartésien en quadrature Q, le réseau RVRNN-8-pip a un paramètre MSE de 0.544×10^{-5} qui est meilleur que celui du réseau RVTDDN-8-pip (2.413×10^{-5}). Par conséquent, le réseau RVRNN-8-pip avec un paramètre MSE de (0.857×10^{-5}) modélise mieux le spectre du modèle de référence de Wiener que le réseau RVTDDN-8-pip (2.969×10^{-5}).

Les architectures de modélisation pseudo-conventionnelles donnent de meilleurs résultats de simulation que celles avec pipeline, mais ne sont malheureusement pas implantables sur la puce FPGA Virtex-6 de la carte ML605, malgré le fait qu'elles nécessitent des quantités de ressources inférieures à celles disponibles sur cette puce FPGA. Leur défaut est qu'elles ont une fréquence maximale d'opération qui est d'environ 37.4 MHz, ce qui est inférieure à la fréquence minimale de traitement d'un échantillon dans cette carte ML605. Celles avec pipeline sont implantables sur la puce FPGA Virtex-6 de la carte ML605, car elles ont une fréquence maximale d'opération de 156.128 MHz et des quantités de ressources requises inférieures à celles disponibles sur la cette puce FPGA, mais sont moins performantes que celles pseudo-conventionnelles à cause des retards insérés dans la chaîne pour le pipeline.

CHAPITRE 3

LINÉARISATION DES AMPLIFICATEURS DE PUISSANCE PAR PRÉDISTORSION NUMÉRIQUE

3.1 Principe de la prédistorsion numérique

Il existe plusieurs algorithmes de prédistorsion parmi lesquels le modèle polynomial (Aladren et al., 2012; Bensmida et al., 2012; Braithwaite, 2012; Choi et al., 2007; Chun et al., 2010; Ding et al., 2012; Djamai et al., 2008; Fehri et Boumaiza, 2011), les séries de Volterra (Gandhi et Abbott, 2010; Guan et Zhu, 2010, 2011a,b; Liszewski et al., 2011), et les réseaux de neurones telles que RVFTDNN (Rawat et al., 2010), TDNN (Hwangbo et al., 2006) et RVRNN (Rawat et al., 2010), RVTDNN (Boumaiza et Mkaem, 2009; Doufana, 2009; Doufana et al., 2010; Mkaem et al., 2010). Dans ce mémoire, seules les architectures de prédistorsion numérique (DPD) à base de réseaux de neurones RVTDNN, RVRNN et NARX seront développées. Pour la prédistorsion, il existe l'architecture d'apprentissage direct et celle d'apprentissage indirect. Dans ce projet, seule la prédistorsion avec apprentissage direct est développée afin de pouvoir implémenter un système capable de corriger les distorsions des PA en temps-réel. Le diagramme de l'apprentissage direct (Choi et al., 2007; Doufana, 2009) représentée dans la figure 83 consiste à placer la prédistorsion numérique avant le PA représenté par le modèle de référence de Wiener. Les erreurs en phase $e_I(n)$ et en quadrature de phase $e_Q(n)$ sont respectivement la différence entre les entrées de prédistorsion numérique, $x_I(n)$ et $x_Q(n)$, et les sorties désirées du modèle de référence de Wiener $d_I(n)$ et $d_Q(n)$ divisé par le gain G du PA. Ces erreurs sont injectées dans la prédistorsion numérique pour la mise à jour des poids et des biais du réseau de neurones par l'algorithme de rétro-propagation.

Le diagramme de l'apprentissage indirect (Choi et al., 2007; Djamai et al., 2008; Doufana, 2009) représentée dans la figure 84 nécessite deux modules de calcul, dont un (*Inverse PA model*) pour calculer le modèle inverse du PA, et l'autre (DPD) utilise les poids du modèle inverse pour réaliser la prédistorsion.

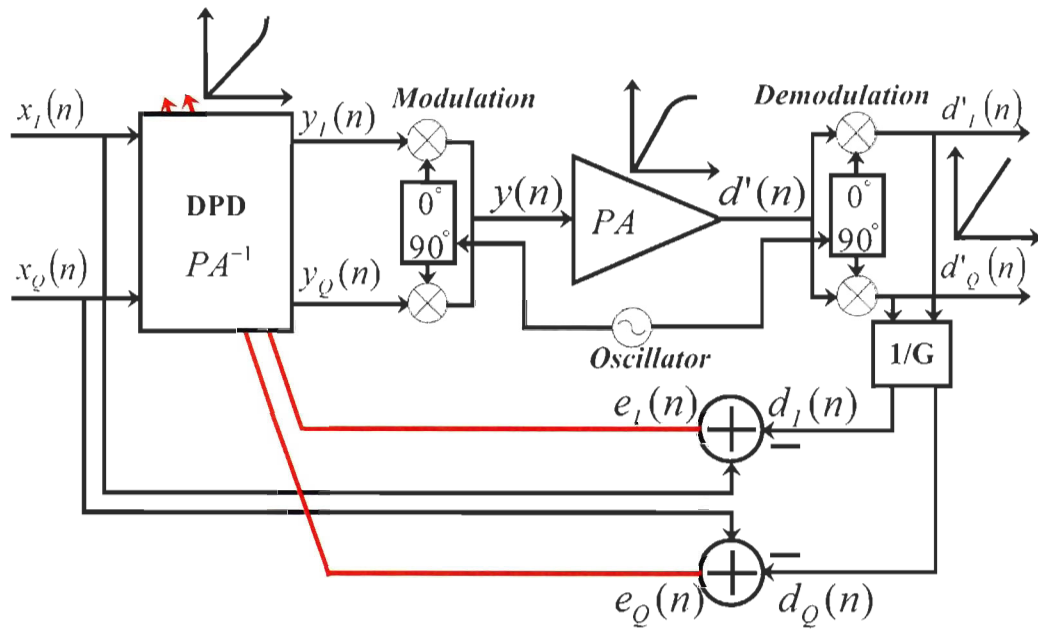


Figure 83: Diagramme de l'apprentissage direct pour la prédistorsion numérique (DPD) en bande de base.

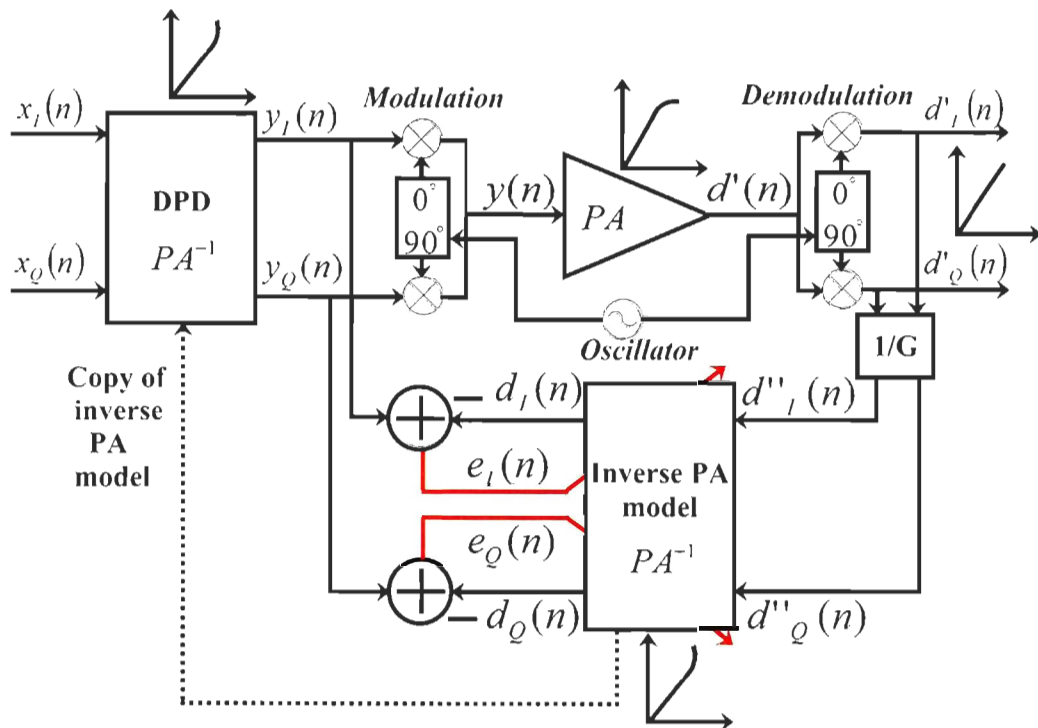


Figure 84: Diagramme de l'apprentissage indirect pour la prédistorsion numérique (DPD) en bande de base.

Les erreurs en phase $e_I(n)$ et en quadrature de phase $e_Q(n)$ sont respectivement la différence entre les sorties de la prédistorsion numérique, $y_I(n)$ et $y_Q(n)$, et les sorties du modèle inverse de PA $d_I(n)$ et $d_Q(n)$. Ces erreurs sont injectées dans le modèle inverse pour la mise à jour des poids et des biais à l'aide de l'algorithme de rétro-propagation. $x_I(n)$ et $x_Q(n)$ sont respectivement les entrées en phase et en quadrature de phase du modèle inverse de la prédistorsion numérique. Les composantes $y_I(n)$ et $y_Q(n)$ sont les parties réelle et imaginaire de l'entrée $y(n)$ du PA. Cette architecture d'apprentissage indirect utilise plus de ressources parce quelle nécessite deux modules (modèle inverse et DPD) et la mise à jour des poids ne peut pas être en temps-réel, mais elle peut être faite de façon périodique ou dès la convergence du modèle inverse.

3.2 Prédistorsion par architectures conventionnelles, pseudo-conventionnelles et pipelinées des réseaux de neurones

Les différentes architectures à base de réseaux de neurones (RVTDNN, RVRNN et NARX) utilisées précédemment dans la modélisation des PA seront également utilisées comme module de prédistorsion. Les différentes équations décrivant la propagation et la rétro-propagation dans ces réseaux seront conservées. La seule différence réside dans la définition de l'erreur $e_k(n)$. L'erreur est définie par $e_k(n) = d_k(n) - y_k(n)$ pour les architectures de modélisation conventionnelles et par $e_k(n) = x_k(n) - d_k(n)$ pour les architectures de prédistorsion conventionnelles. Elle est définie par $e_k(n-1) = d_k(n-1) - y_k(n-1)$ pour les architectures de modélisation pseudo-conventionnelles et par $e_k(n-1) = x_k(n-1) - d_k(n-1)$ pour les architectures de prédistorsion pseudo-conventionnelles. Elle est définie par $e_k(n-24) = d_k(n-24) - y_k(n-24)$ pour les architectures de modélisation pipelinées et par $e_k(n-24) = x_k(n-24) - d_k(n-24)$ pour les architectures de prédistorsion pipelinées excepté le réseau RVTDNN-6-pip. Elle est définie par $e_k(n-16) = d_k(n-16) - y_k(n-16)$ pour l'architecture de modélisation RVTDNN-6-pip et par $e_k(n-16) = x_k(n-16) - d_k(n-16)$ pour l'architecture de prédistorsion RVTDNN-6-pip.

3.3 Implantation des architectures de prédistorsion en utilisant les blocs de l'outil XSG

3.3.1 Implantation des architectures pseudo-conventionnelles

XSG est aussi l'outil utilisé pour programmer et générer les fichiers *bitstream* à charger dans la puce FPGA Virtex-6 de la carte ML605. Le système de test avec modulation 16-QAM des architectures de prédistorsion NARX- N_1 -pse, RVRNN- N_1 -pse et RVTDNN- N_1 -pse est représenté sur la figure 85. Ce système de test avec modulation 16-QAM a la même constitution que dans la section 2.4.1. La période d'échantillonnage du générateur est 1 μ s.

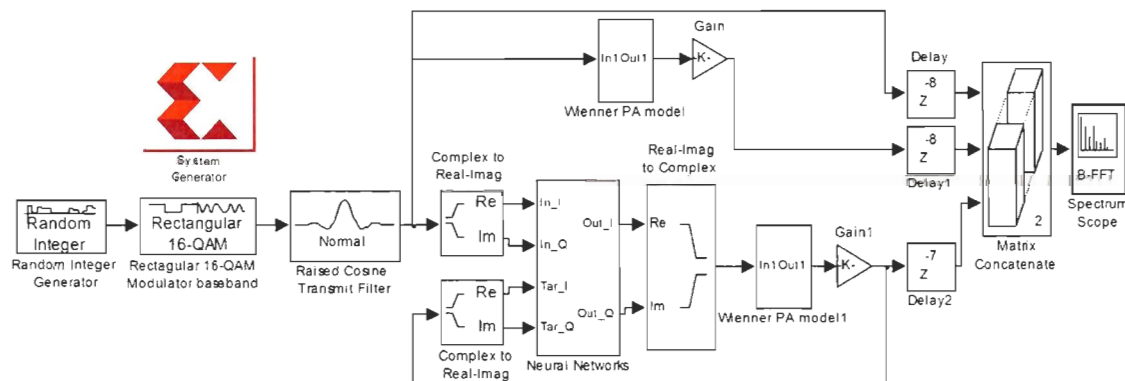


Figure 85: Système de test avec modulation 16-QAM des architectures pseudo-conventionnelles de prédistorsion à bande de base.

3.3.2 Implantation des architectures pipelinées

Le système de test avec modulation 16-QAM à bande de base de l'architecture de prédistorsion RVTDNN-8-pip, RVRNN- N_1 -pip et NARX- N_1 -pip est représenté sur la figure 86. Le système de test avec modulation 16-QAM à bande de base de l'architecture de prédistorsion RVTDNN-6-pip est représenté sur la figure 87. Les 16 retards de la figure 87 et les 24 retards de la figure 86 à la sortie du bloc *neural networks* de chacune de ces figures sont justifiés dans la section 2.6.1.

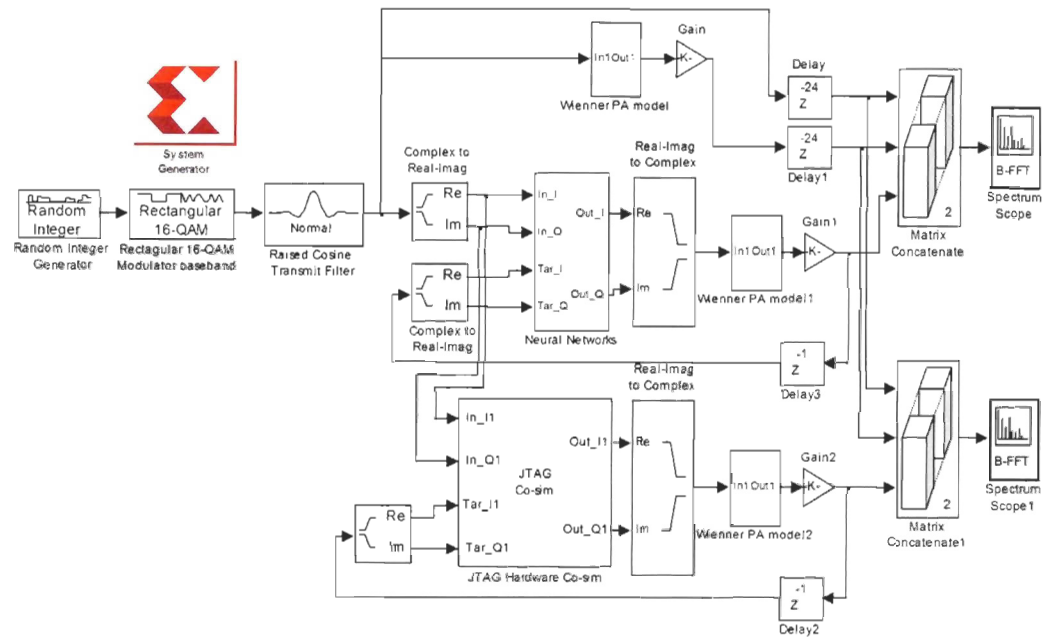


Figure 86: Système de test avec modulation 16-QAM à bande de base des architectures de prédistorsion RVTDNN-8-pip, RVRNN- N_1 -pip et NARX- N_1 -pip et du bloc *JTAG Hardware Co-simulation*.

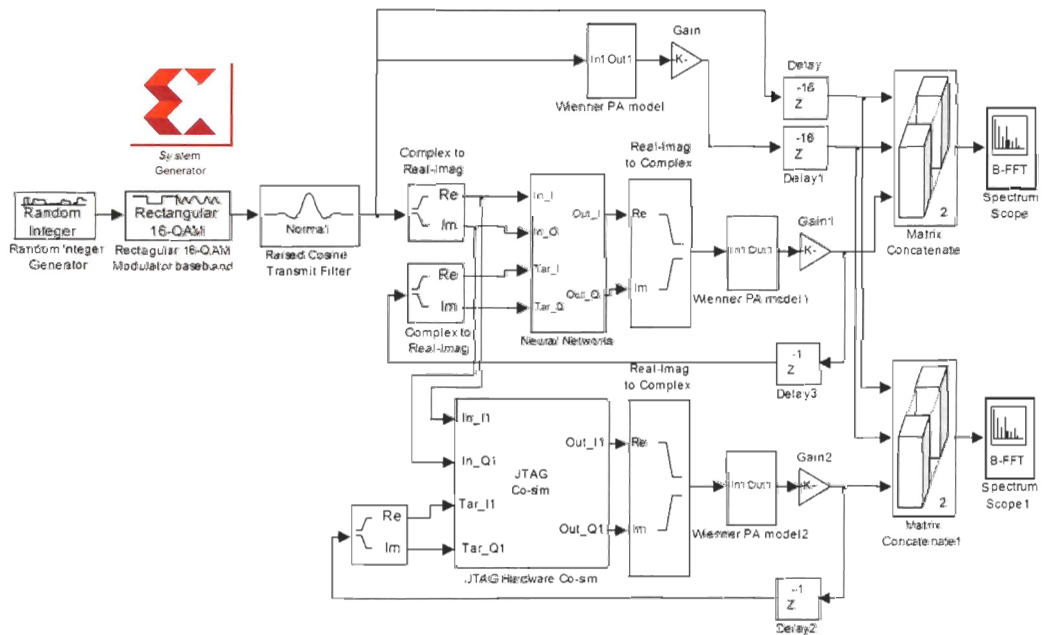


Figure 87: Système de test avec modulation 16-QAM à bande de base des architectures de prédistorsion RVTDNN-6-pip et du bloc *JTAG Hardware Co-simulation*.

3.4 Implantation des architectures de prédistorsion sur la puce FPGA

Pour les architectures de prédistorsion pseudo-conventionnelles, la fréquence maximale d'opération (MOF) est d'environ 42.2 MHz (voir le tableau 4), mais reste encore inférieure à la fréquence minimale de traitement d'un échantillon acceptable pour la génération d'un bloc *JTAG Co-simulation* avec succès. C'est la raison pour laquelle les architectures de prédistorsion pseudo-conventionnelles développées ne peuvent être implantées avec succès par l'interface *Hardware Co-simulation*. Cependant, les architectures de prédistorsion avec pipeline ont une fréquence MOF de 156.128 MHz (voir le tableau 5). Cette fréquence étant largement supérieure à la fréquence maximale de co-simulation (100 MHz) disponible sur la carte ML605, ces architectures de prédistorsion avec pipeline peuvent être implantées avec succès par l'interface *Hardware Co-simulation*. La fréquence de 100 MHz a été choisie lors de la génération du bloc JTAG co-simulation. Les différentes ressources estimées dans ce mémoire ont été évaluées lors de la génération directe du fichier *bitstream*.

3.4.1 Implantation des architectures pseudo-conventionnelles

Les architectures de prédistorsion RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse ont une estimation de ressources totales inférieure aux ressources disponibles sur la puce FPGA en utilisant le bloc System Generator (tableau 4) pour la génération du fichier *bitstream*. Les architectures de prédistorsion RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse ont chacune une fréquence MOF d'environ 42.2 MHz. Leur pourcentage de ressources estimées ne dépasse pas 100 %, ce qui les rend implantables dans la puce FPGA Virtex-6 de la carte ML605. La compilation de chacune de ces architectures pour la génération du fichier *Bitstream* s'est faite comme dans la section 2.7.1. Cette configuration a permis la génération avec succès du fichier *Bitstream* pour toutes les architectures de prédistorsion. À la fin de la compilation d'une architecture, une fenêtre comportant les détails sur les opérations effectuées lors de la génération des ressources de cette architecture s'ouvre. Dans cette fenêtre se trouvent les ressources estimées de chacune de ces architectures tel que représentées dans le tableau 4. Les architectures de prédistorsion pseudo-conventionnelles ont le même nombre de bascules (Flips-Flop), de *Bonded IOB*, de RAMB36E1 et de DSP48E1 que celles de modélisation pseudo-conventionnelles. Seuls les tables LUT ont augmenté. Par exemple, pour l'architecture DPD RVTDNN-6-pse, les ressources de la table LUT sont estimées à 7.45% (tableau 4),

alors que pour l'architecture de modélisation RVTDDN-6-pse, les ressources de la table LUT sont estimées à 7.00% (tableau 1). La fréquence MOF a été améliorée dans les architectures de prédistorsion pseudo-conventionnelles par rapport à celles de modélisation pseudo-conventionnelles. Dans le tableau 4, la fréquence MOF de l'architecture DPD RVTDDN-6-pse est de 43.148 MHz, alors que dans le tableau 1 la fréquence MOF de l'architecture de modélisation RVTDDN-6-pse est de 37.440 MHz. Cette variation de la fréquence MOF est due au calcul de l'erreur des architectures de prédistorsion qui est différent de celles de modélisation (voir section 3.2).

3.4.2 Implantation des architectures pipelinées

Les architectures de prédistorsion RVTDDN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip du tableau 5 ont le même nombre de ressources bascules, RAMB36E1, IOB et DSP48E1 que celles de modélisation du tableau 2. Seules les ressources de la table LUT sont différentes. La fréquence MOF des architectures pipelinées de prédistorsion est restée la même (156.128 MHz) que celle des architectures pipelinées de modélisation. L'uniformité de la fréquence MOF pour les architectures de prédistorsion avec pipeline justifie une bonne optimisation et une bonne synchronisation des retards.

3.5 Expérimentation et analyse des résultats de prédistorsion par des architectures pseudo-conventionnelles et pipelinées

Comme pour la modélisation, les architectures de prédistorsions pseudo-conventionnelles et avec pipeline sont évaluées avec un signal de test à bande de base modulé 16-QAM en utilisant la boîte à outils de communication du logiciel Matlab. Dans ce système de test, les 16 points de constellation sont analysés lors du processus de prédistorsion numérique avant et après leur correction avec les architectures de prédistorsion.

Pour mener à bien l'évaluation de la prédistorsion, différentes courbes sont tracées, dont les caractéristiques AM/AM et AM/PM, le spectre du signal à l'entrée et à la sortie du modèle de référence de Wiener et des différentes architectures de prédistorsion, l'évolution des composants cartésiens I et Q en fonction du temps. Le paramètre EVM est aussi calculé pour exprimer le pourcentage de fidélité de la constellation des signaux à la sortie des architectures de prédistorsion par rapport à celle du signal d'entrée. Pour le tracé du spectre, de la constellation, des conversions AM/AM et AM/PM et de l'évolution des composants cartésiens I et Q en fonction du temps, la simulation est d'abord faite sur une durée de 20 s à la période d'échantillonnage $0.125 \mu s$. Puis, les poids optimisés sont réinjectés dans les différents réseaux de neurones au niveau du registre à décalage de l'ajustement des biais et des poids (voir section 2.4.1.2). Enfin, la simulation est reprise sur une durée d'une seconde. Pour la simulation du paramètre MSE, les poids des programmes d'architectures DPD sont initialisés avec la fonction *randn* du logiciel Matlab.

3.5.1 Expérimentation des architectures de prédistorsion pseudo-conventionnelles et pipelinées

Les systèmes de test et les différents blocs de simulation des architectures de prédistorsion RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse, est représenté à la figure 85, celui de prédistorsion RVTDNN-8-pip, RVRNN- N_1 -pip et NARX- N_1 -pip est représenté à la figure 86 et finalement celui de prédistorsion RVTDNN-6-pip sont respectivement représentés à la figure 87. Le bloc *JTAG Hardware Co-simulation* dans chacune de ces figures sert à vérifier le bon fonctionnement des différentes architectures de prédistorsion sur la puce FPGA Virtex-6 de la carte ML605, mais les courbes sont uniquement tracées avec le bloc de simulation *Neural Networks* pouvant être n'importe laquelle des architectures. Le fichier *Bitstream*, quant à lui, est généré pour toutes les architectures de prédistorsion afin d'évaluer les différentes ressources estimées de chaque architecture comparativement à celles disponibles sur la carte ML605 (voir la section 3.4.1 et la section 3.4.2).

3.5.2 Analyse des résultats de prédistorsion par des architectures pseudo-conventionnelles et pipelinées

Les résultats montrent clairement une bonne prédistorsion des différentes architectures. Cependant, certaines ont un meilleur paramètre EVM et une meilleure réduction des paramètres ACPR que d'autres.

La constellation permet de visualiser la correction des distorsions, due au modèle de PA de Wiener à l'intérieur de la bande de fréquence (EVM), par rapport au signal d'entrée. Pour la caractérisation du modèle de référence de Wiener et des architectures de prédistorsion, les caractéristiques de conversion AM/AM et de compression AM/PM sont utilisées. L'évolution temporelle des composants cartésiens réel (I) et imaginaire (Q) permet de montrer le degré de similitude entre le signal d'entrée et le signal de sortie du système d'amplification incluant les techniques DPD neuronales développées. Le spectre illustre le niveau de réduction des paramètres ACPR par rapport au modèle de référence de Wiener. Le spectre permet aussi de visualiser les distorsions à l'extérieur de la bande de fréquence

ainsi que la capacité des architectures proposées à compenser ces distorsions. Le paramètre EVM en pourcent est calculée pour spécifier le pourcentage de distorsion de la chaîne d'amplification incluant les architectures développées par rapport au signal d'entrée. La fonction MSE est utilisée pour évaluer la fidélité des conversions AM/AM et AM/PM et des composants cartésiens I et Q en fonction du temps par rapport au signal d'entrée.

Le calcul du paramètre MSE relatif à la vitesse de convergence des architectures de prédistorsion est obtenu à partir de 10 différents tests dans lesquels est faite varier la constante x de la fonction $randn('state',x)$, où x varie de 1 à 10. La variation de x implique une initialisation différente des poids et biais dans les registres à décalage de chaque neurone (voir section 2.4.1.2). Du fait que le signal d'entrée soit à bande de base, la fréquence centrale du canal principal est 0. La largeur de bande du canal principal est 1.35 MHz. Le décalage du canal adjacent gauche est -1 MHz et celui du canal adjacent droit est 1 MHz.

3.5.3 Performances des architectures de prédistorsion RVTDDN-6-pse, RV-RNN-6-pse et NARX-6-pse

La caractérisation quantitative des diagrammes de constellation des architectures de linéarisation à base du réseau RVTDDN-6-pse, du réseau RVRNN-6-pse et du réseau NARX-6-pse en série

avec le modèle de référence de Wiener et de l'entrée représentés sur la figure 88, est faite par calcul du paramètre EVM en pourcent. Les architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence de Wiener présentent respectivement de 4.050 %, 1.170 % et 0.913 % du paramètre EVM par rapport à l'entrée.

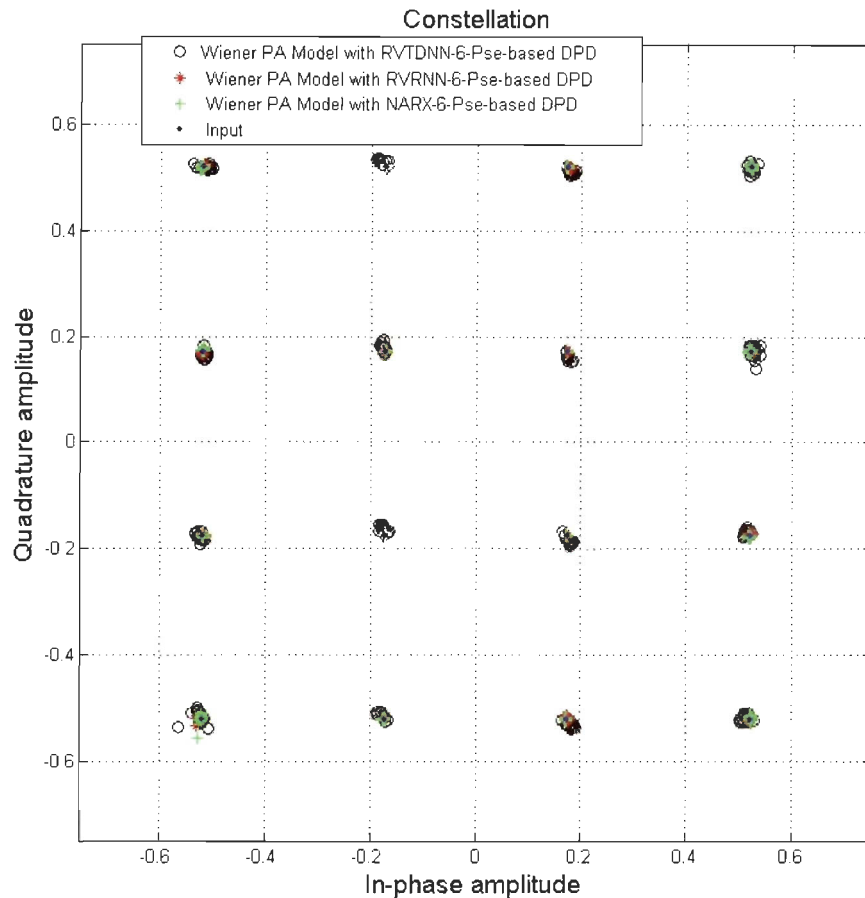


Figure 88: Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence de Wiener

L'évaluation quantitative des caractéristiques de conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence de Wiener, représentées sur la figure 89, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse ont respectivement comme paramètre MSE 13.153×10^{-5} , 0.790×10^{-5} et 0.727×10^{-5} par rapport à l'entrée.

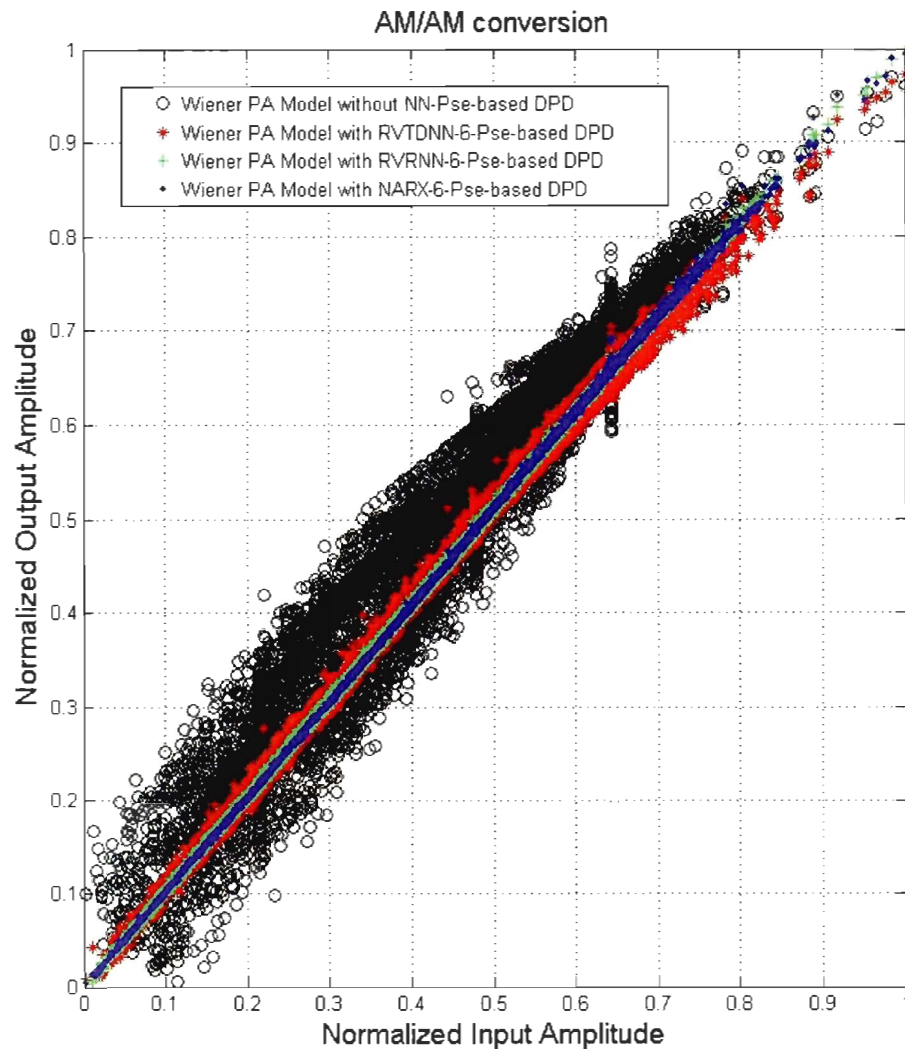


Figure 89: Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.

L'évaluation quantitative des caractéristiques de conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence, représentées sur la figure 90 est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVDNN-6-pse, RVRNN-6-pse et NARX-6-pse ont respectivement comme paramètre MSE 4.781, 0.095 et 0.076 par rapport à l'entrée.

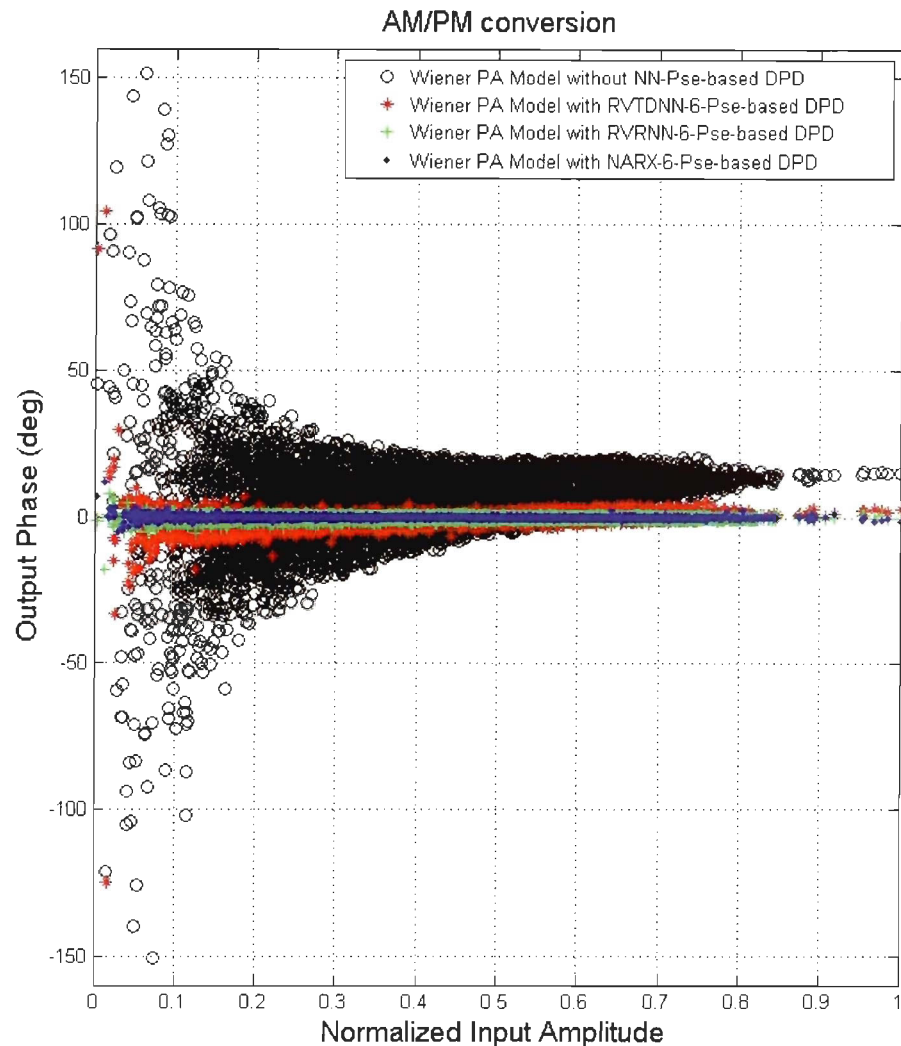


Figure 90: Conversion AM/PM du modèle de référence de Wiener seul et des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.

La caractérisation quantitative des composants cartésiens I et Q de l'entrée et des architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence de Wiener prédistorsion, représentées sur la figure 91, est aussi faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDNN-6-pse, RVRNN-6-pse et NARX-6-pse ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 10.057×10^{-5} , 0.763×10^{-5} et 0.488×10^{-5} et en quadrature (Q) 37.973×10^{-5} , 0.795×10^{-5} et 1.174×10^{-5} par rapport au signal d'entrée.

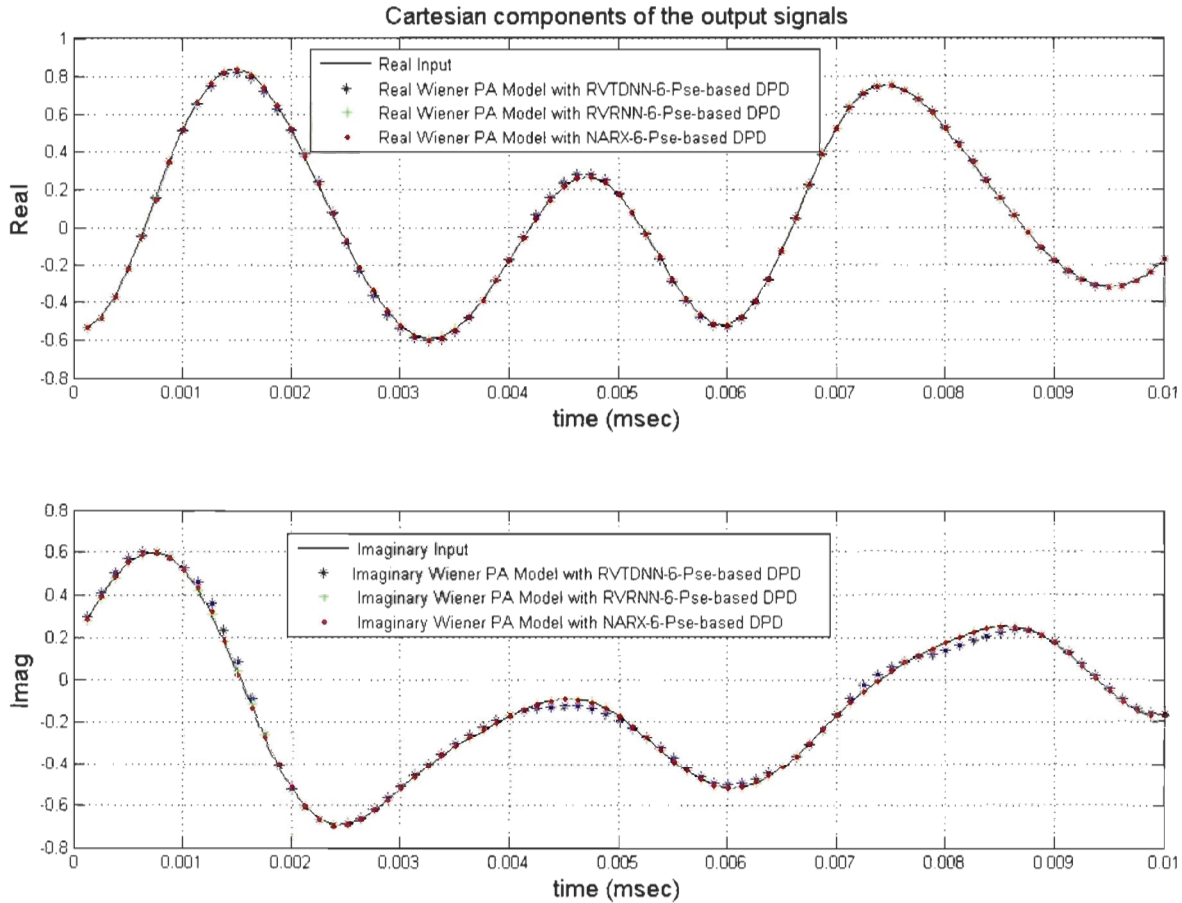


Figure 91: Composants cartésiens I et Q de l’entrée et des sorties des architectures de prédistorsion RVTDDN-6-pse, RVRNN-6-pse et NARX-6-pse en série avec le modèle de référence.

Les spectres de l’entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDDN-6-pse en série avec le modèle de référence de Wiener sont représentés sur la figure 92. Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, la diminution des paramètres ACPR gauche et droit du signal à la sortie du réseau NARX-6-pse, du réseau RVRNN-6-pse et du réseau RVTDDN-6-pse par rapport au signal d’entrée est respectivement de (14.570 dB et 14.376 dB), (14.670 dB et 14.848) et (6.127 dB et 5.335).

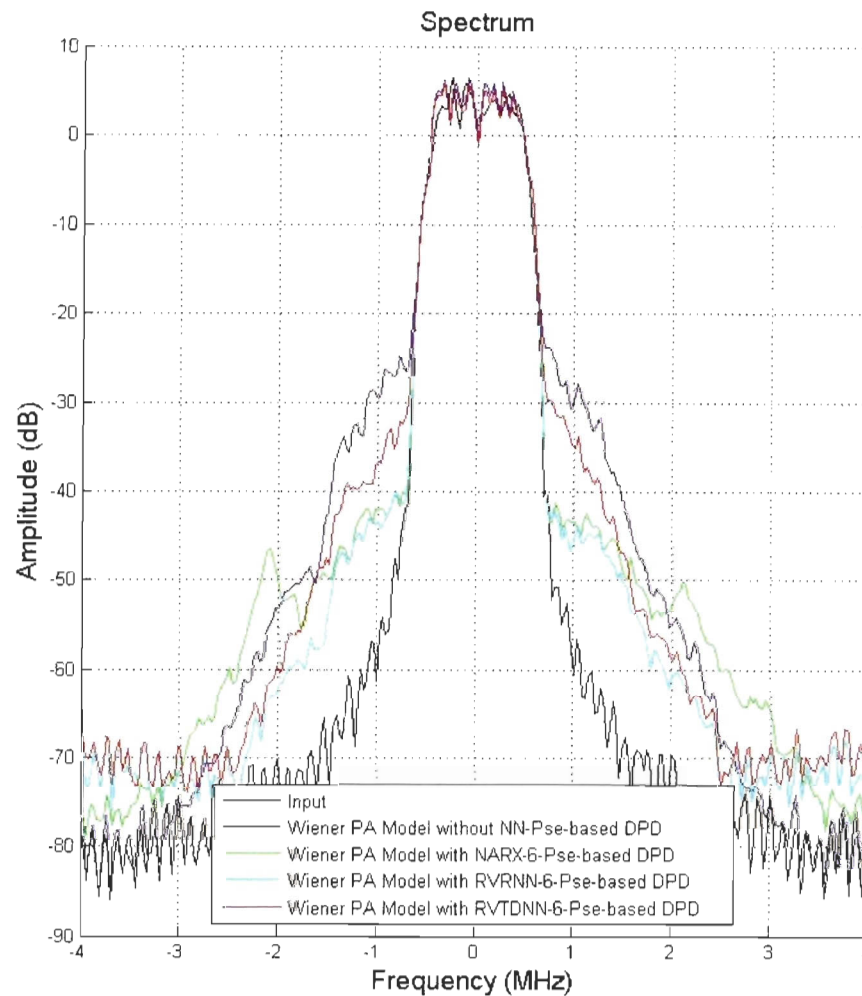


Figure 92: Spectre de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse en série avec le modèle de référence.

Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse en série avec le modèle de référence sont représentés sur la figure 93. Les architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVTDNN-6-pse en série avec le modèle de référence ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0167, 0.0191 et 0.0146.

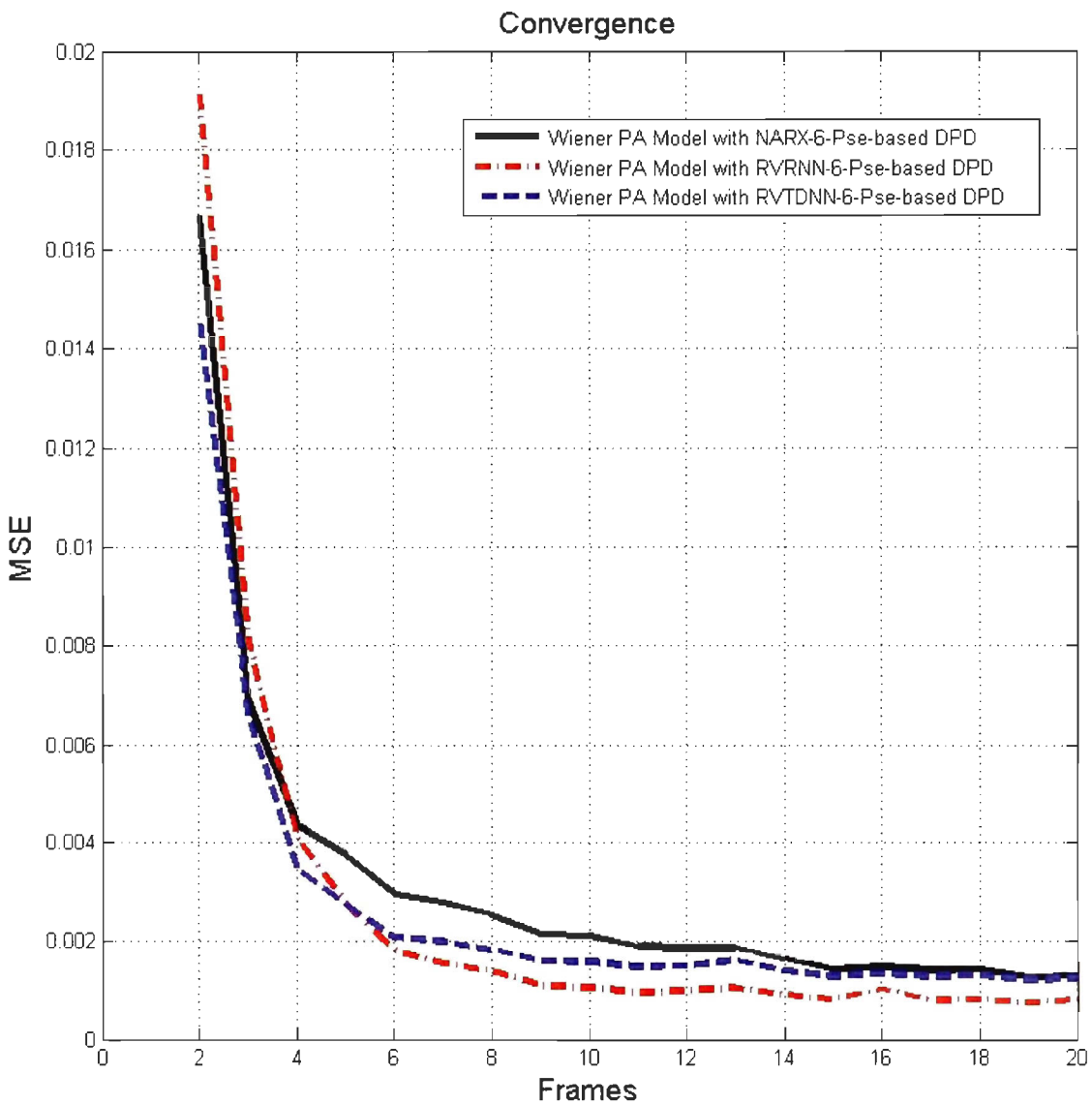


Figure 93: Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pse, RVRNN-6-pse et RVDNN-6-pse en série avec le modèle de référence.

De l'analyse des résultats de simulation des courbes découle que l'architecture de prédistorsion NARX-6-pse en série avec le modèle de référence de Wiener corrige en général mieux les distorsions dues à ce modèle de référence que celles du réseau RVRNN-6-pse qui lui-même corrige mieux les distorsions que le réseau RVDNN-6-pse. L'architecture de prédistorsion NARX-6-pse a un pourcentage de paramètre EVM (0.913 %) qui est inférieur à celui du

réseau RVRNN-6-pse (1.170 %) qui est lui-même inférieur à celui du réseau RVTDNN-6-pse (4.050 %). Lors de la conversion AM/AM, le paramètre MSE de l'architecture de prédistorsion NARX-6-pse en série avec le modèle de référence de Wiener a 0.727×10^{-5} qui est plus petit que celui du réseau RVRNN-6-pse (0.790×10^{-5}) qui est lui-même plus petit que celui du réseau RVTDNN-6-pse (13.153×10^{-5}). Pour les conversions AM/PM, le paramètre MSE de l'architecture de prédistorsion NARX-6-pse en série avec le modèle de référence de Wiener a 0.076 qui est plus petit que celui du réseau RVRNN-6-pse (0.095) qui est lui-même plus petit que celui du réseau RVTDNN-6-pse (4.781). Le calcul du paramètre MSE du composant cartésien I montre que le réseau NARX-6-pse en série avec le modèle de référence de Wiener a un paramètre MSE de 0.488×10^{-5} , inférieur à celui du réseau RVRNN-6-pse qui est de 0.763×10^{-5} et qui est aussi plus petit que le réseau RVRNN-6-pse (10.057×10^{-5}). Le calcul du paramètre MSE pour la caractérisation du composant cartésien Q montre que le réseau RVRNN-6-pse en série avec le modèle de référence de Wiener a un paramètre MSE de 0.795×10^{-5} inférieur à ceux du réseau NARX-6-pse qui est de 1.174×10^{-5} et du réseau RVTDNN-6-pse d'une valeur de 37.973×10^{-5} . Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, le réseau RVRNN-6-pse (14.670 dB et 14.848) a une meilleure diminution des paramètres ACPR gauche et droit par rapport au modèle de référence de Wiener que le réseau NARX-6-pse (14.570 dB et 14.376 dB) qui est meilleure que le réseau RVTDNN-6-pse (6.127 dB et 5.335). Cependant, à la 2^{ème} trame de la figure 93 le réseau RVTDNN-6-pse converge plus vite que le réseau NARX-6-pse qui lui-même converge plus vite que le réseau RVRNN-6-pse, car ils ont respectivement comme paramètre MSE 0.0146, 0.0167 et 0.0191. Par contre, à partir de la 6^{ème} trame, le réseau RVRNN-6-pse converge plus vite que le réseau RVRNN-6-pse qui, à son tour, converge plus vite que le réseau NARX-6-pse.

3.5.4 Performances des architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse

La caractérisation quantitative des diagrammes de constellation de l'entrée et des sorties des architectures de linéarisation à base du réseau RVTDNN-8-pse, du réseau RVRNN-8-pse et du réseau NARX-8-pse en série avec le modèle de référence de Wiener représentés sur la figure 94, est faite par calcul du paramètre EVM en pourcent. Les architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse présentent respectivement 0.934 %, 0.509 % et 0.505 % de paramètre EVM par rapport à l'entrée.

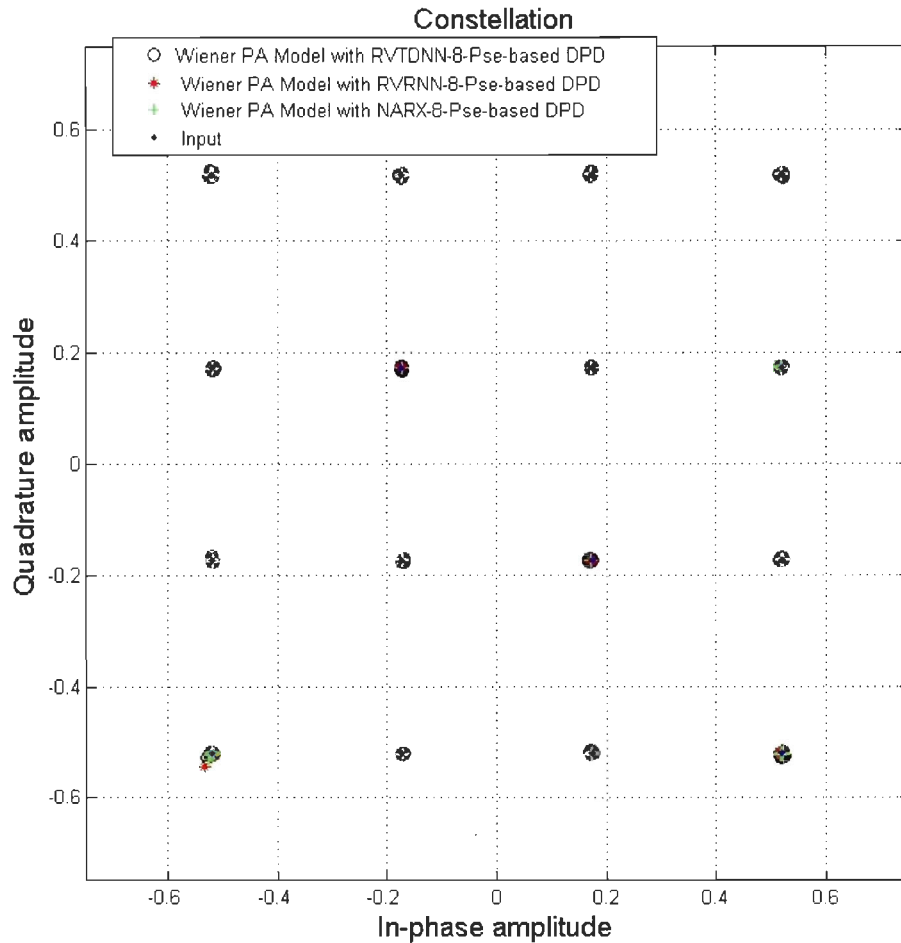


Figure 94: Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.

L'évaluation quantitative des caractéristiques de conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener, représentées sur la figure 95, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse ont respectivement comme paramètre MSE 0.775×10^{-5} , 0.115×10^{-5} et 0.170×10^{-5} par rapport à l'entrée.

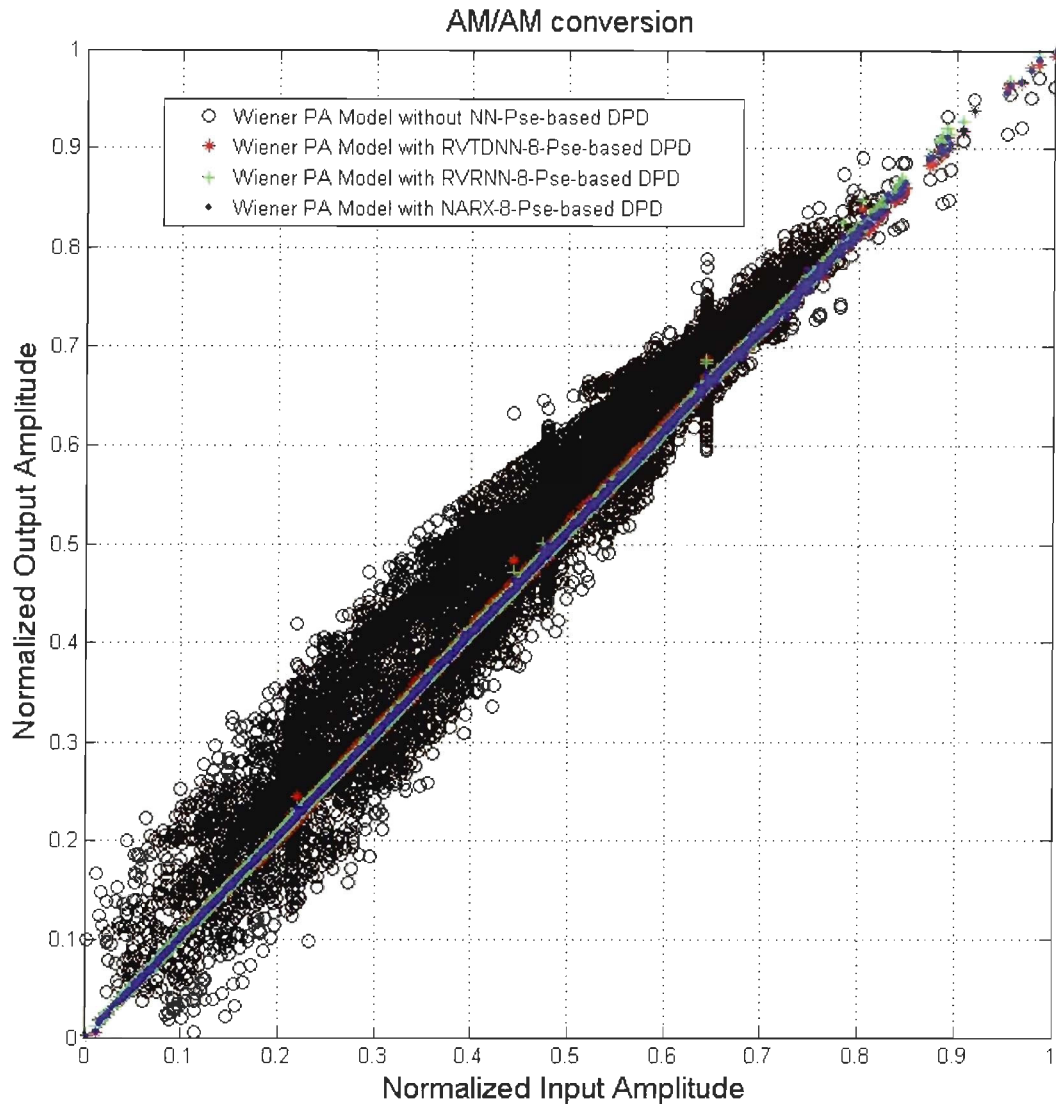


Figure 95: Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.

L'évaluation quantitative des caractéristiques de conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence, représentées sur la figure 96, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVDNN-8-pse, RVRNN-8-pse et NARX-8-pse ont respectivement comme paramètre MSE 0.061, 0.046 et 0.032 par rapport à l'entrée.

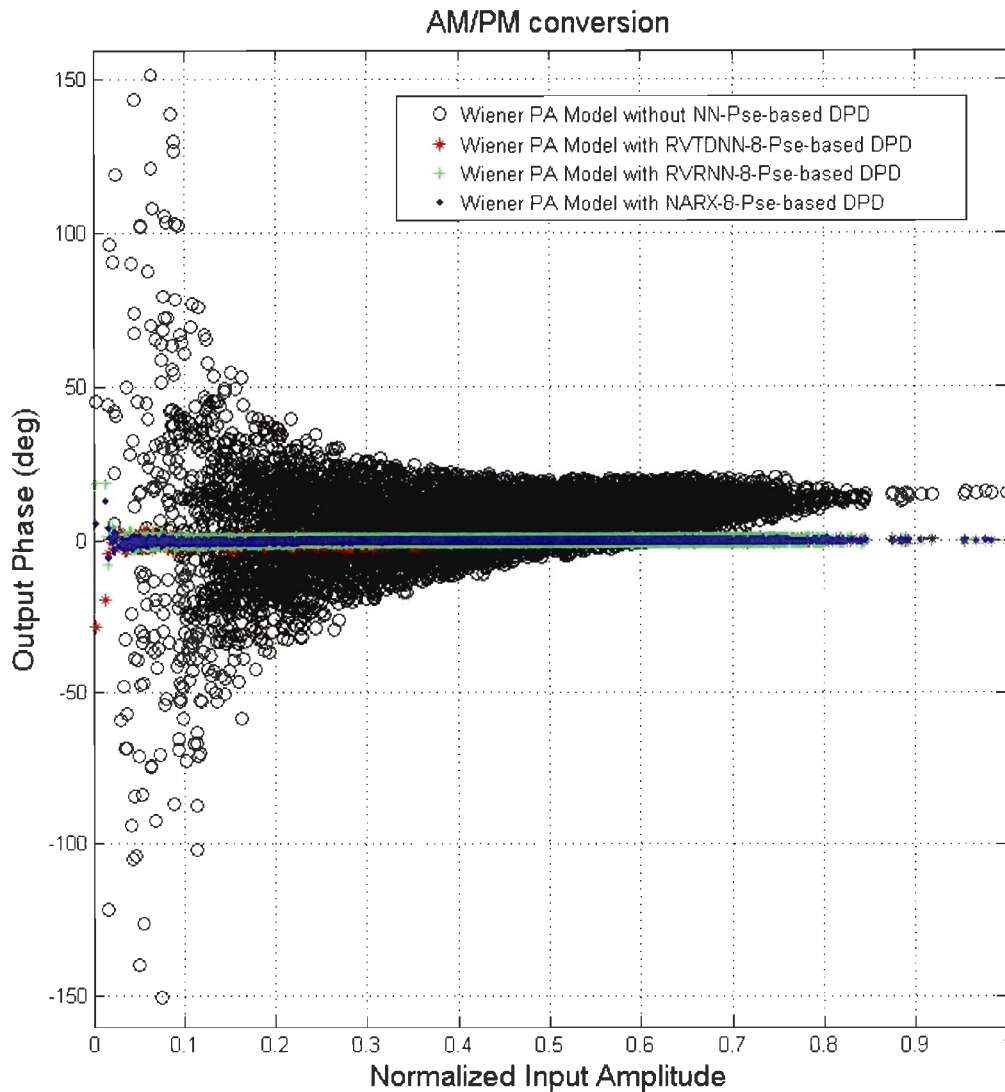


Figure 96: Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVT-DNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.

La caractérisation quantitative des composants cartésiens I et Q de l'entrée et des architectures de prédistorsion RVT-DNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener, représentées sur la figure 97, est aussi faite par calcul du paramètre MSE. Les architectures de prédistorsion RVT-DNN-8-pse, RVRNN-8-pse et NARX-8-pse ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 0.459×10^{-5} , 0.160×10^{-5} et 0.175×10^{-5} et en quadrature (Q) 0.779×10^{-5} , 0.294×10^{-5} et 0.207×10^{-5} par

rapport au signal d'entrée.

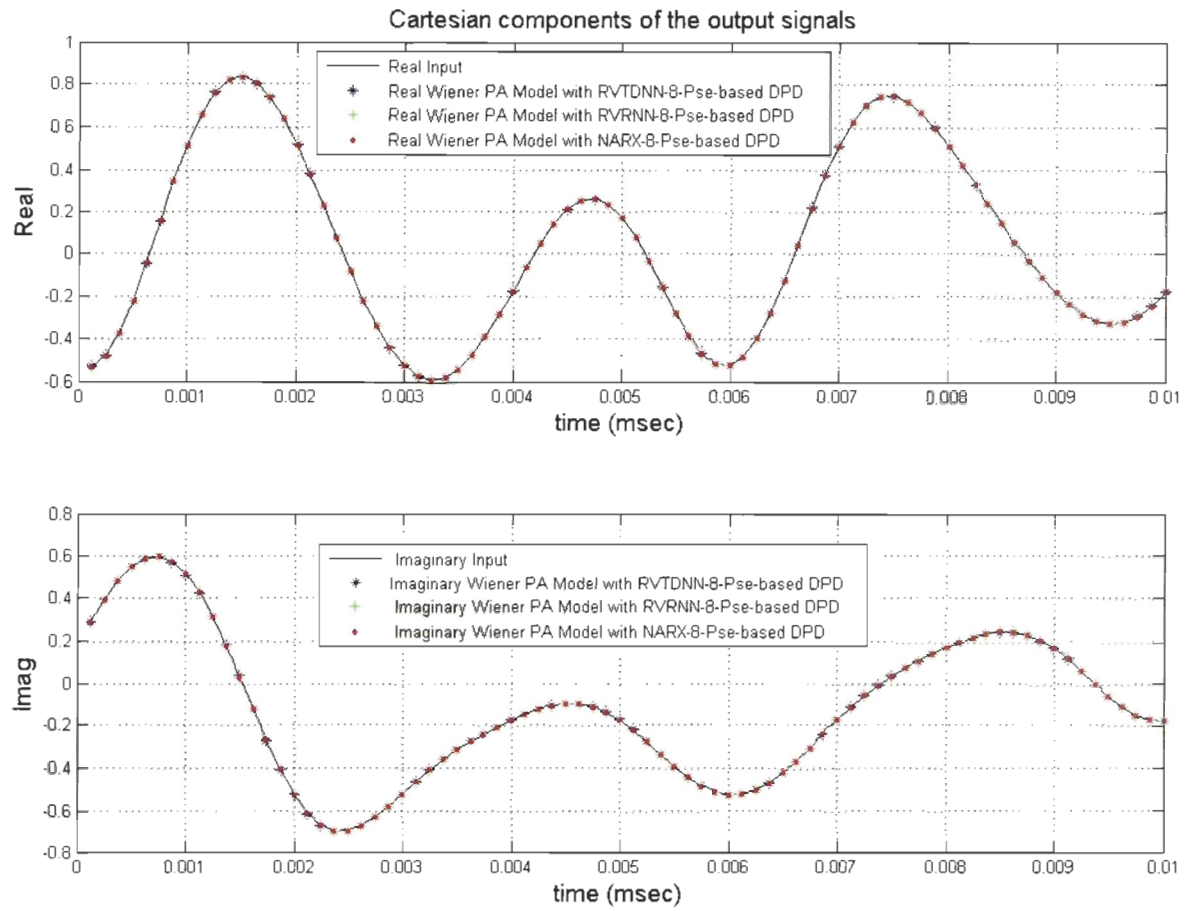


Figure 97: Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDNN-8-pse, RVRNN-8-pse et NARX-8-pse en série avec le modèle de référence de Wiener.

Les spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse en série avec le modèle de référence de Wiener sont représentés sur la figure 98. Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, les diminutions des paramètres ACPR gauche et droit du signal à la sortie du réseau NARX-8-pse, du réseau RVRNN-8-pse et du réseau RVTDNN-8-pse par rapport au signal d'entrée sont respectivement de (18.440 dB et

18.223 dB), (18.786 dB et 17.774) et (15.617 dB et 15.343).

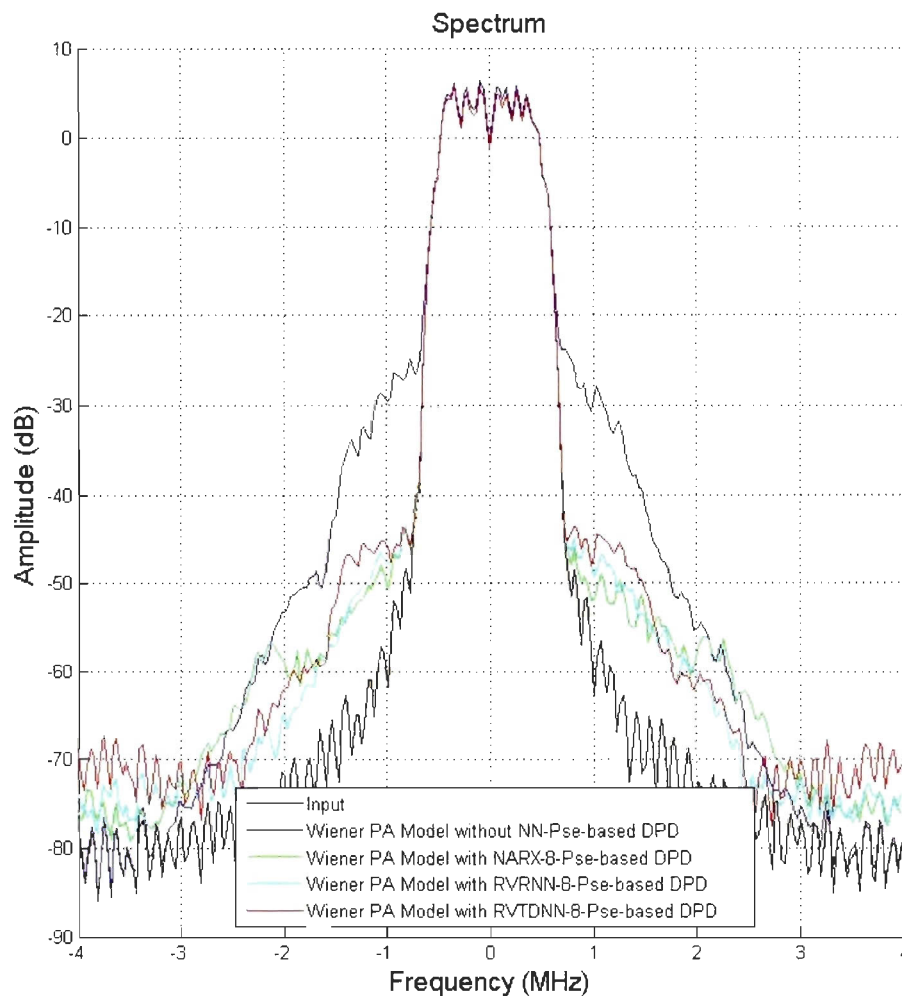


Figure 98: Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion du réseau NARX-8-pse, du réseau RVRNN-8-pse et du réseau RVTDNN-8-pse en série avec le modèle de référence.

Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse en série avec le modèle de référence sont représentés sur la figure 99. Les architectures de prédistorsion NARX-8-pse, RVRNN-8-pse et RVTDNN-8-pse en série avec le modèle de référence ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0111, 0.0102 et 0.0132.

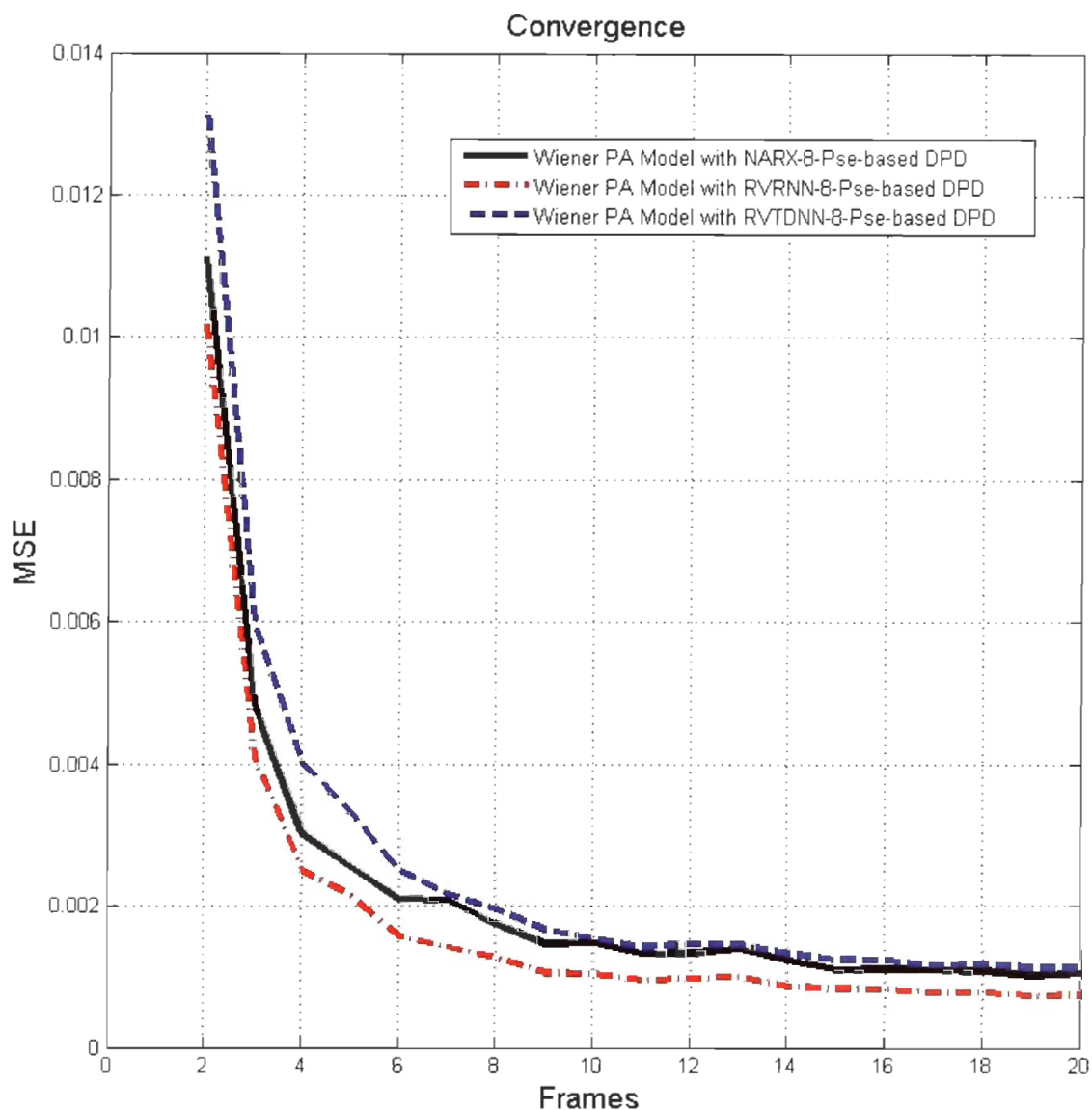


Figure 99: Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pse, RVRNN-8-pse et RVDNN-8-pse en série avec le modèle de référence.

De l'analyse des résultats de simulation des différentes courbes découle que l'architecture de prédistorsion RVRNN-8-pse en série avec le modèle de référence de Wiener corrige mieux les distorsions dues au modèle de référence de Wiener que celle du réseau NARX-8-pse qui elle-même corrige mieux les distorsions que le réseau RVDNN-8-pse. L'architecture de prédistorsion NARX-8-pse a un pourcentage de paramètre EVM (0.505 %) qui est inférieur à

celui du réseau RVRNN-8-pse (0.509 %) qui est lui-même inférieur à celui du réseau RVTDNN-8-pse (0.934 %).

Pour les conversions AM/AM, le paramètre MSE de l'architecture de prédistorsion RVRNN-8-pse est de 0.115×10^{-5} qui est plus petit que celui du réseau NARX-8-pse (0.170×10^{-5}) qui est lui-même plus petit que celui du réseau RVTDNN-8-pse (0.775×10^{-5}). Pour les conversions AM/PM, le paramètre MSE de l'architecture de prédistorsion NARX-8-pse est 0.032 qui est plus petit que celui du réseau RVRNN-8-pse (0.046) qui est lui-même plus petit que celui du réseau RVTDNN-8-pse (0.061).

Le calcul du paramètre MSE pour la caractérisation du composant cartésien I montre que le prédistorteur RVRNN-8-pse a un paramètre MSE de 0.160×10^{-5} , inférieur à celui du réseau NARX-8-pse qui est de 0.175×10^{-5} et du réseau RVTDNN-8-pse d'une valeur de 0.459×10^{-5} . Le calcul du paramètre MSE pour la caractérisation du composant cartésien Q montre que le réseau NARX-8-pse a un paramètre MSE de 0.207×10^{-5} inférieur à celui du réseau RVRNN-8-pse qui est de 0.294×10^{-5} et du réseau RVTDNN-8-pse d'une valeur de 0.779×10^{-5} .

Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, le réseau NARX-8-pse (18.440 dB et 18.223 dB) a une meilleure diminution des paramètres ACPR gauche et droit que le réseau RVRNN-8-pse (18.786 dB et 17.774) qui est meilleur que le réseau RVTDNN-8-pse (15.617 dB et 15.343) par rapport au modèle de référence de Wiener.

Cependant, à la 2^{ème} trame, le réseau RVRNN-8-pse converge plus vite que le réseau NARX-8-pse qui lui-même converge plus vite que le réseau RVTDNN-8-pse, car ils ont respectivement comme paramètre MSE 0.0102, 0.0111 et 0.0132 dans la figure 99.

3.5.5 Performances des architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip

La caractérisation quantitative des diagrammes de constellation de l'entrée et des sorties des architectures de linéarisation à base du réseau RVTDNN-6-pip, du réseau RVRNN-6-pip et du réseau NARX-6-pip en série avec le modèle de référence de Wiener, représentés sur la figure 100, est faite par calcul du paramètre EVM en pourcent. Les architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip présentent respectivement 2.673 %, 3.256 % et 5.203 % de paramètre EVM par rapport à l'entrée.

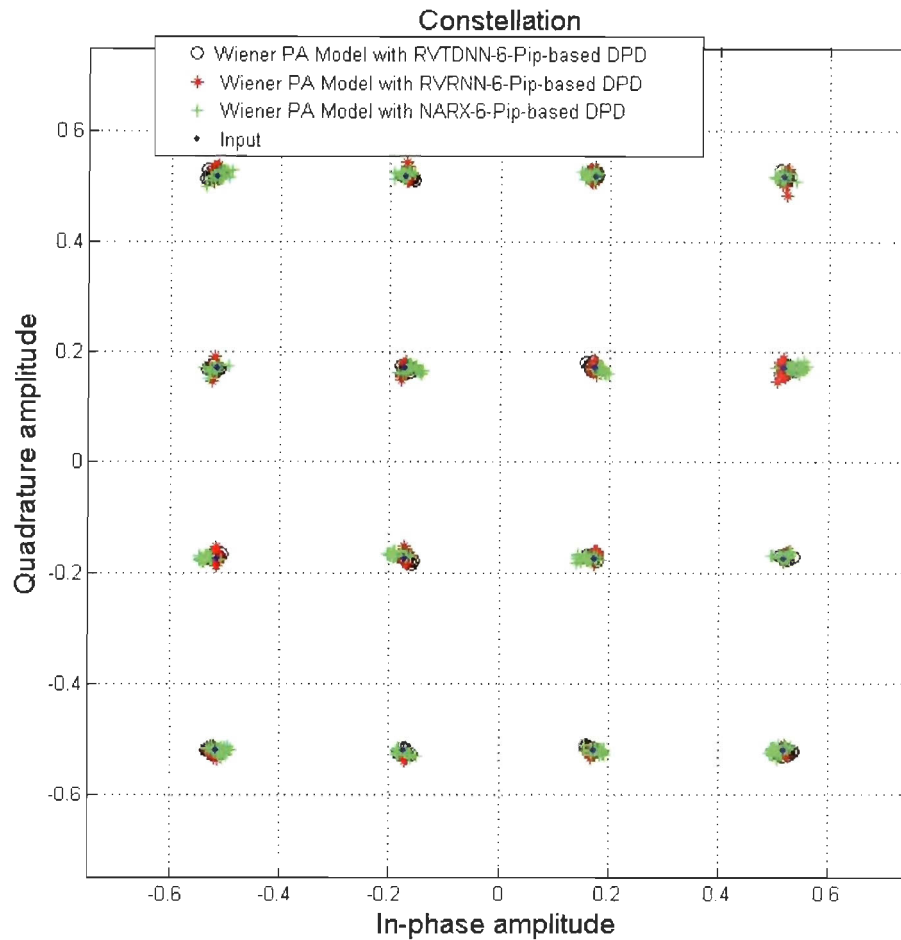


Figure 100: Diagramme de constellation de l'entrée et des sorties des architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.

L'évaluation quantitative des caractéristiques de conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener, représentées sur la figure 101, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip ont respectivement comme paramètre MSE 8.475×10^{-5} , 7172.141×10^{-5} et 7465.193×10^{-5} par rapport à l'entrée.

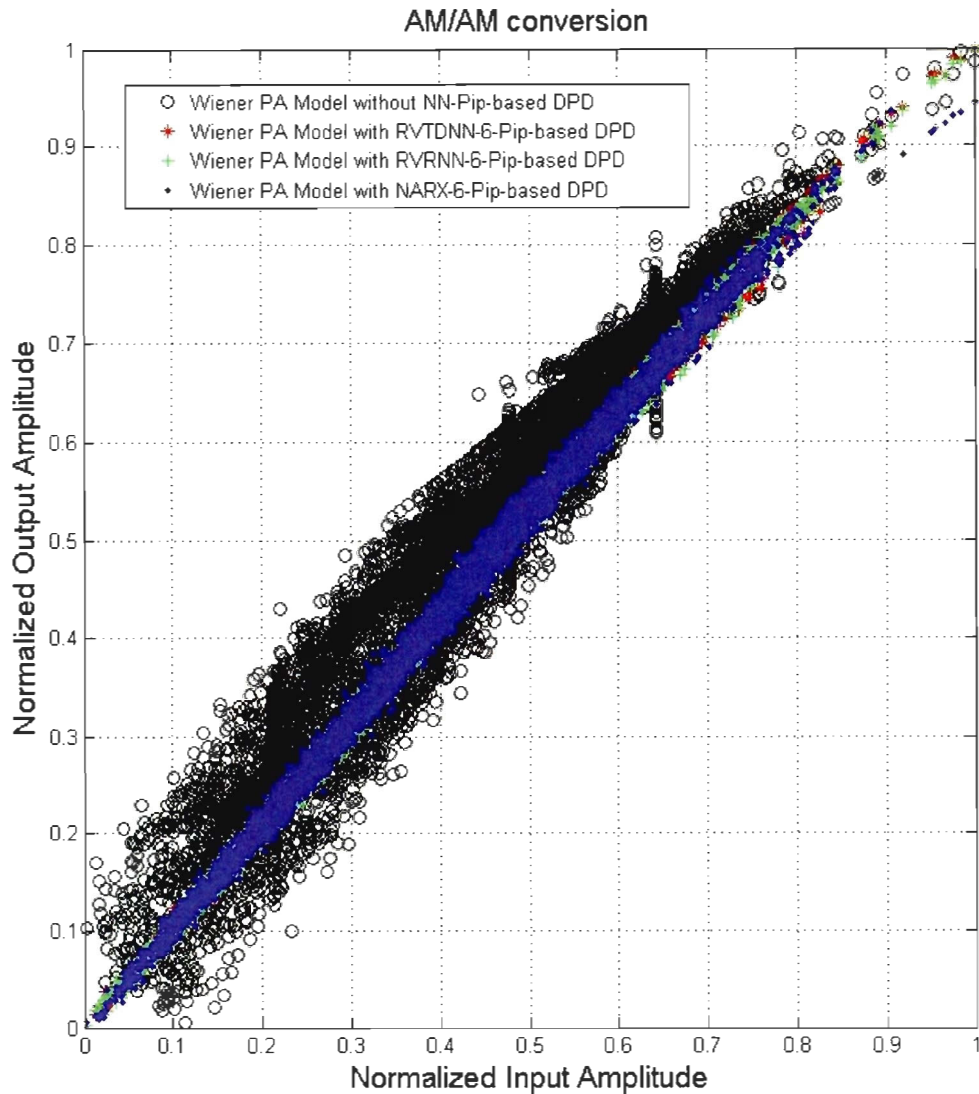


Figure 101: Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.

L'évaluation quantitative des caractéristiques de conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-6-pip RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence, représentées sur la figure 102, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDDN-6-pip RVRNN-6-pip et NARX-6-pip ont respectivement comme paramètre MSE 0.773, 0.585 et 3.686 par rapport à l'entrée.

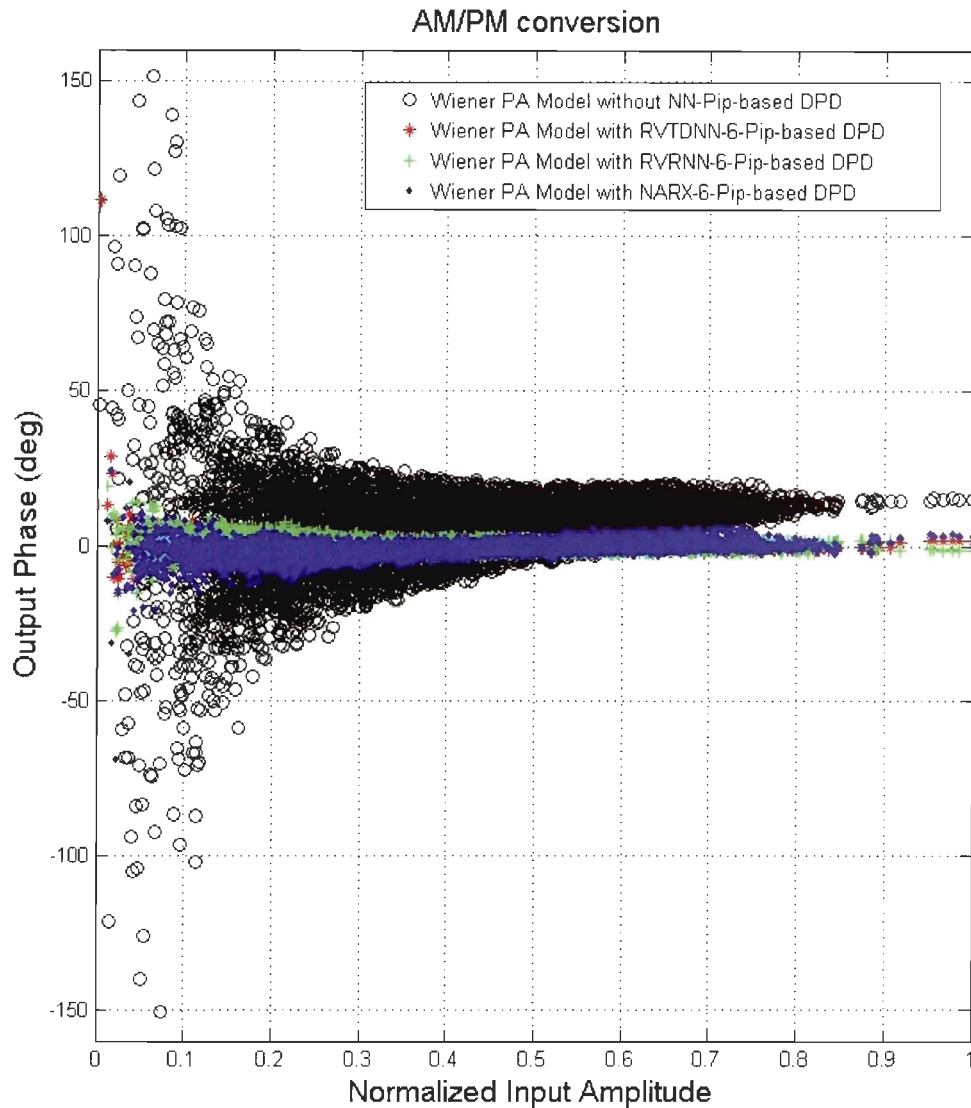


Figure 102: Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVDNN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.

La caractérisation quantitative des composants cartésiens I et Q de l'entrée et des architectures de prédistorsion RVDNN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener, représentées sur la figure 103, est aussi faite par calcul du paramètre MSE. Les architectures de prédistorsion RVDNN-6-pip, RVRNN-6-pip et NARX-6-pip ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 11.512×10^{-5} , 49235.784×10^{-5} et 50086.441×10^{-5} et en quadrature (Q) 1.879×10^{-5} , 21605.679×10^{-5} et

21364.731×10^{-5} par rapport au signal d'entrée.

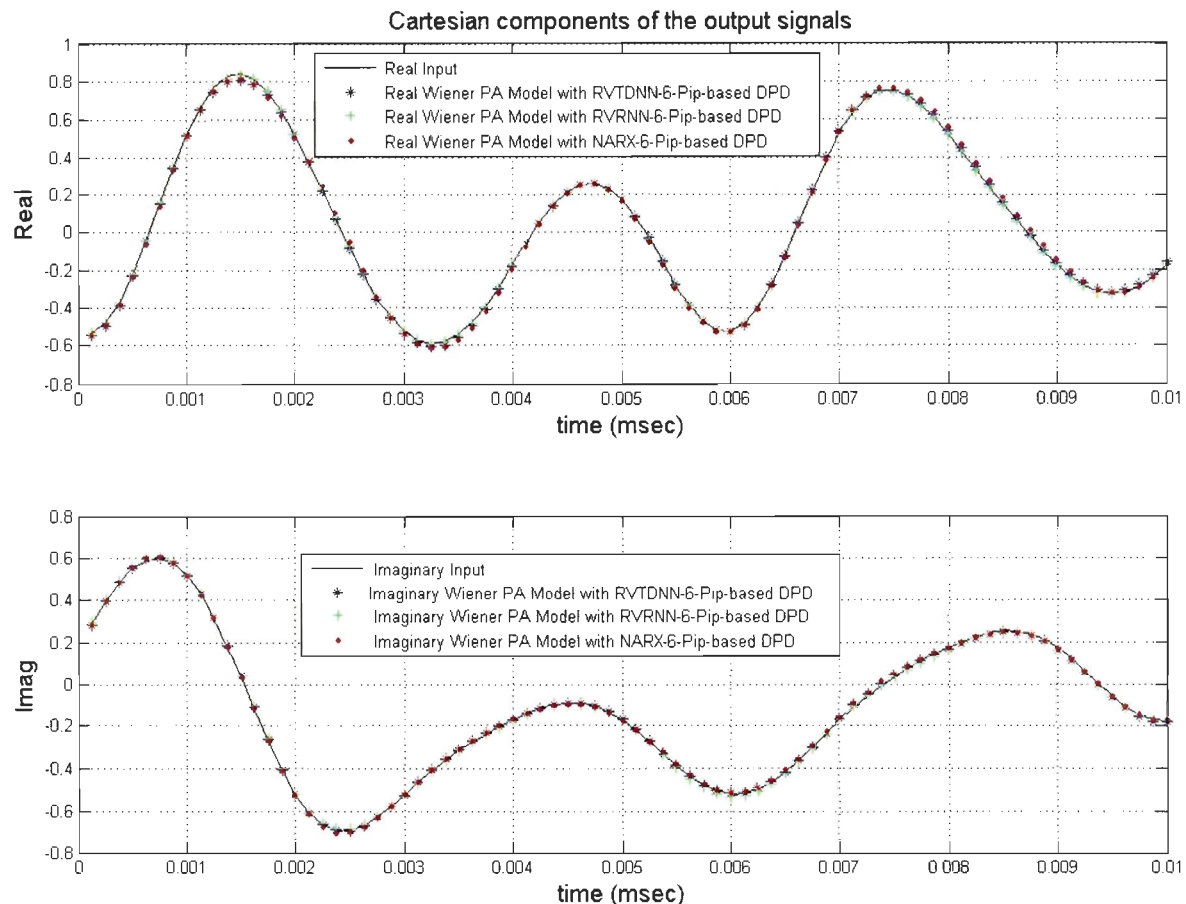


Figure 103: Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDNN-6-pip, RVRNN-6-pip et NARX-6-pip en série avec le modèle de référence de Wiener.

Les spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDNN-6-pip en série avec le modèle de référence de Wiener sont représentés sur la figure 104. Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, les diminutions des paramètres ACPR gauche et droit du signal à la sortie du réseau NARX-6-pip, du réseau RVRNN-6-pip et du réseau RVTDNN-6-pip par rapport au signal d'entrée sont respectivement de (5.472 dB et 5.239 dB), (11.928 dB et 11.928) et (12.685 dB et 12.759).

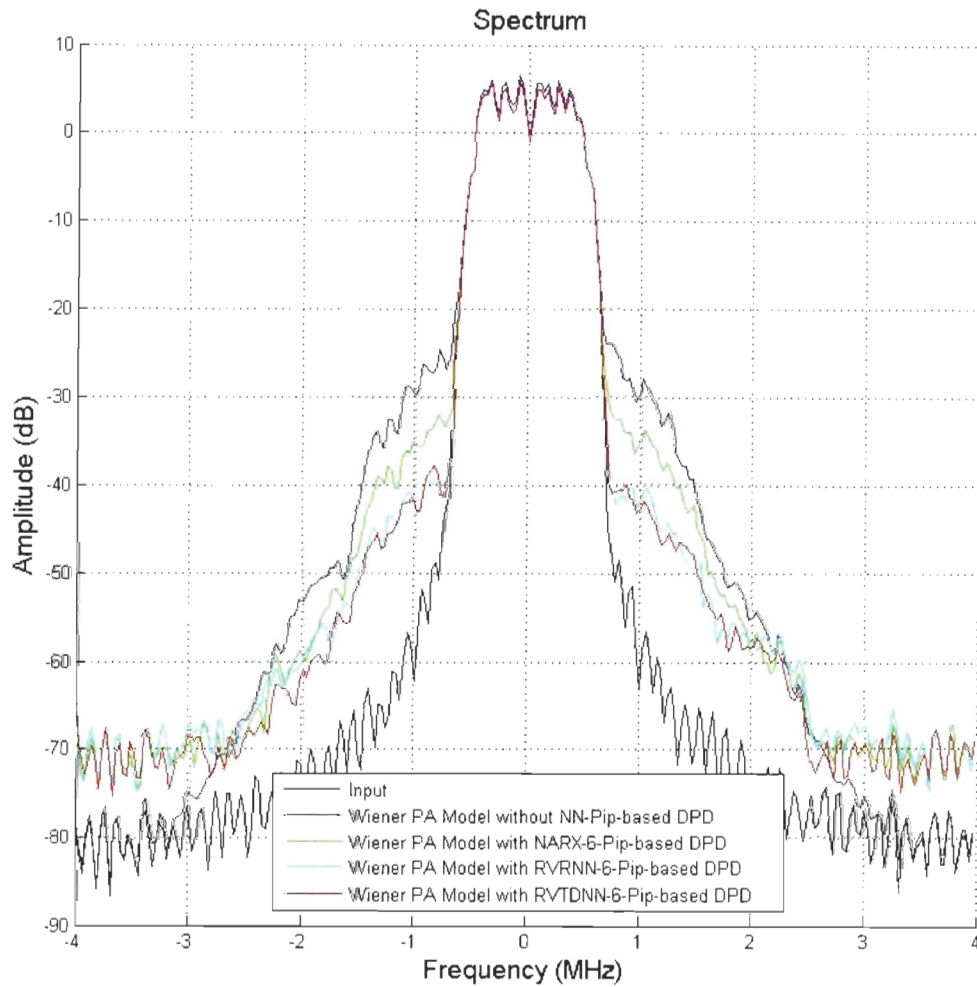


Figure 104: Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip en série avec le modèle de référence.

Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip en série avec le modèle de référence sont représentés sur la figure 105. Les architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip en série avec le modèle de référence ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0684, 0.0519 et 0.0234.

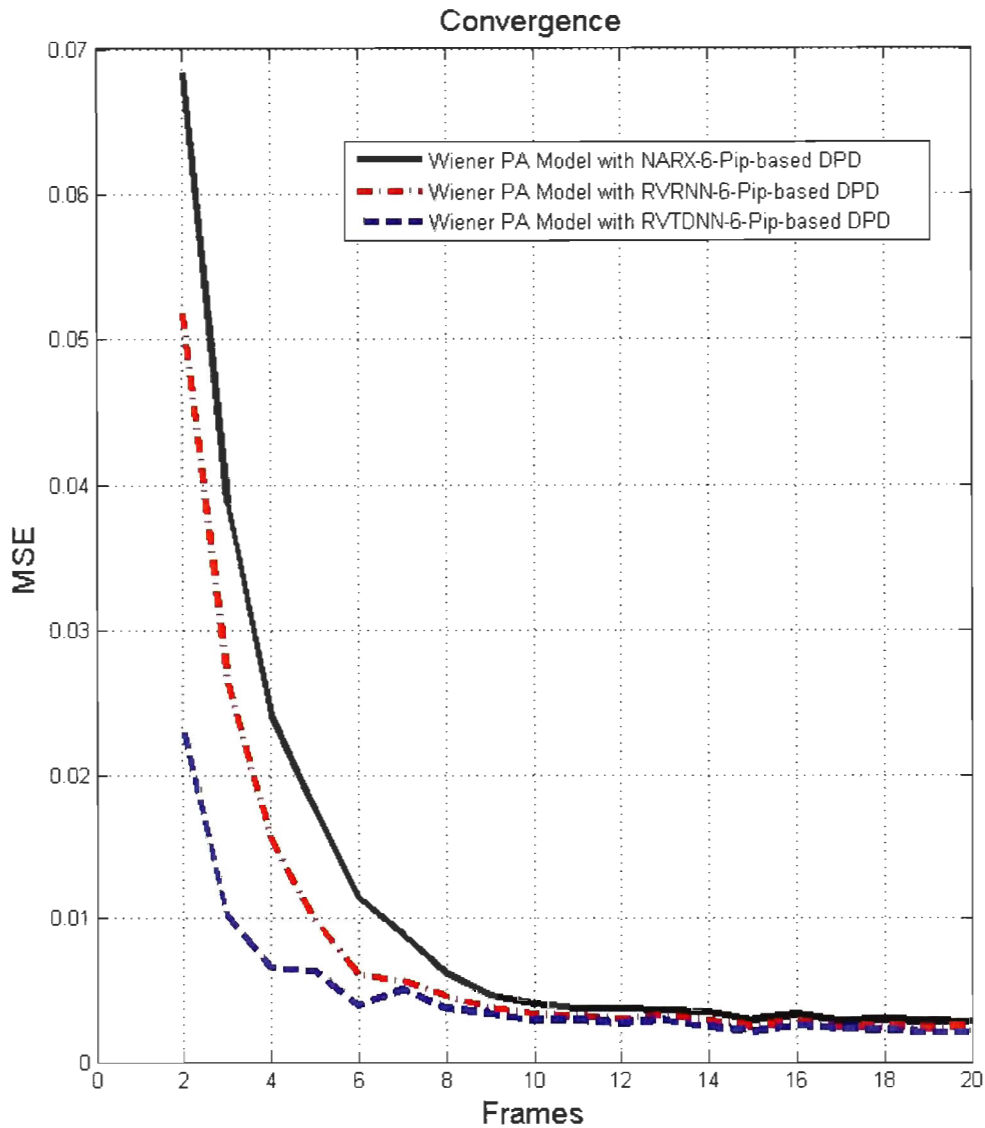


Figure 105: Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-6-pip, RVRNN-6-pip et RVTDDN-6-pip en série avec le modèle de référence.

De l'analyse des résultats de simulation des différentes courbes, découle que l'architecture de prédistorsion RVTDDN-6-pip corrige en général mieux les distorsions dues au modèle de référence de Wiener que celle du réseau RVRNN-6-pip qui elle-même corrige mieux les distorsions que le réseau NARX-6-pip. L'architecture de prédistorsion RVTDDN-6-pip a un pourcentage de paramètre EVM (2.673 %) qui est inférieur à celui du réseau RVRNN-6-pip (3.256 %) de Wiener qui est lui-même inférieur à celui du réseau NARX-6-pip (5.203 %). Pour

les conversions AM/AM, le paramètre MSE de l'architecture de prédistorsion RVTDDN-6-pip en série avec le modèle de référence de Wiener est de 8.475×10^{-5} qui est plus petit que celui du réseau RVRNN-6-pip (7172.141×10^{-5}) qui est lui-même plus petit que celui du réseau NARX-6-pip (7465.193×10^{-5}). Pour les conversions AM/PM, le paramètre MSE de l'architecture de prédistorsion RVRNN-6-pip a 0.585 qui est plus petit que celui du réseau RVTDDN-6-pip (0.773) qui est lui-même plus petit que celui du réseau NARX-6-pip (3.686). Le calcul du paramètre MSE pour la caractérisation du composant cartésien I montre que le réseau RVTDDN-6-pip a un paramètre MSE (11.512×10^{-5}) inférieur à celui du réseau RVRNN-6-pip qui est de 49235.784×10^{-5} et du réseau NARX-6-pip d'une valeur de 50086.441×10^{-5} . Le calcul du paramètre MSE pour la caractérisation du composant cartésien Q montre que le réseau RVTDDN-6-pip a un paramètre MSE (1.879×10^{-5}) inférieur à celui du réseau NARX-6-pip qui est de 21364.731×10^{-5} et du réseau RVRNN-6-pip d'une valeur de 21605.679×10^{-5} . Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, le réseau RVTDDN-6-pip (12.685 dB et 12.759) a une meilleure diminution des paramètres ACPR gauche et droit que le réseau RVRNN-6-pip (11.928 dB et 11.928) qui est meilleure que le réseau NARX-6-pip (5.472 dB et 5.239 dB) par rapport au modèle de référence de Wiener.

Cependant, à la 2^{ème} trame de la figure 105, le réseau RVTDDN-6-pip converge plus vite que le réseau RVRNN-6-pip qui lui-même converge plus vite que le réseau NARX-6-pip, car ils ont respectivement comme paramètre MSE 0.0234, 0.0519 et 0.0684. Par contre, à la 4^{ème} trame, le réseau RVTDDN-6-pip et le réseau RVRNN-6-pip en série avec le modèle de référence de Wiener ont la même vitesse de convergence tandis que le réseau RVRNN-6-pip converge plus vite.

3.5.6 Performances des architectures de prédistorsion RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip

La caractérisation quantitative des diagrammes de constellation de l'entrée et des sorties des architectures de linéarisation à base du réseau RVTDDN-8-pip, du réseau RVRNN-8-pip et du réseau NARX-8-pip en série avec le modèle de référence de Wiener, représentés sur la figure 106, est faite par calcul du paramètre EVM en pourcent. Les architectures de prédistorsion RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip présentent respectivement 2.488 %, 2.414 % et 2.193 % de paramètre EVM par rapport à l'entrée.

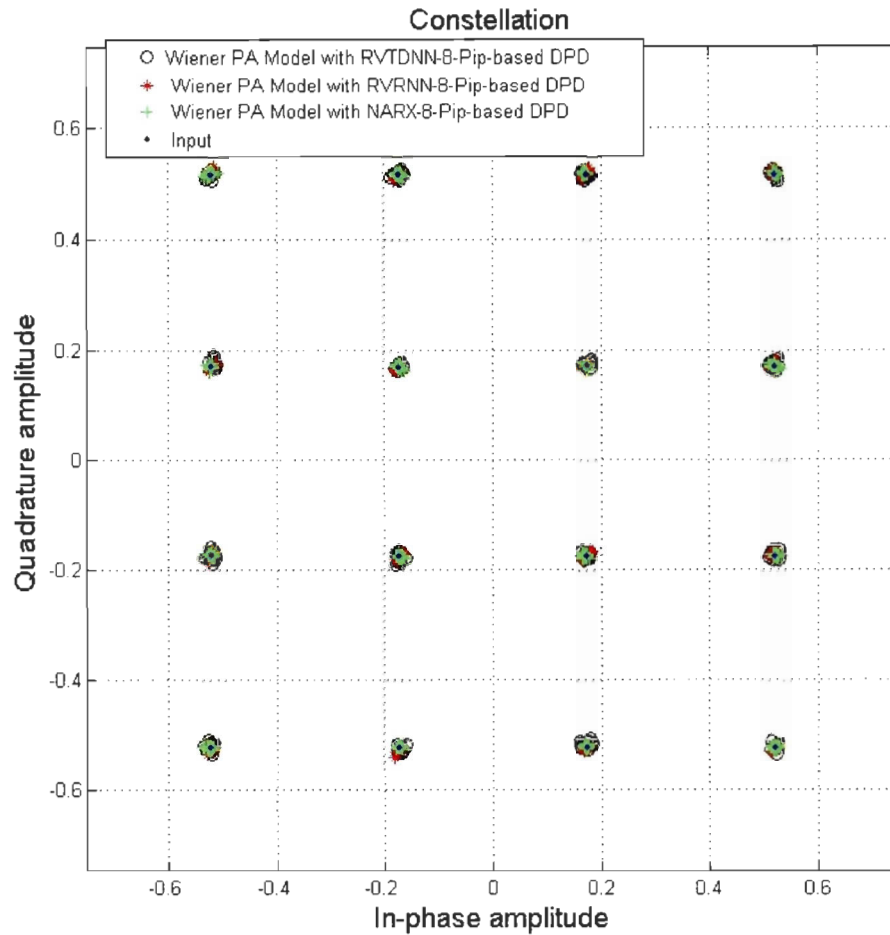


Figure 106: Diagramme de constellation des architectures de prédistorsion RVTDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener et de l'entrée.

L'évaluation quantitative des caractéristiques de conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener, représentées sur la figure 107, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDNN-8-pip, RVRNN-8-pip et NARX-8-pip ont respectivement comme paramètre MSE 3.005×10^{-5} , 5.581×10^{-5} et 5.576×10^{-5} par rapport à l'entrée.

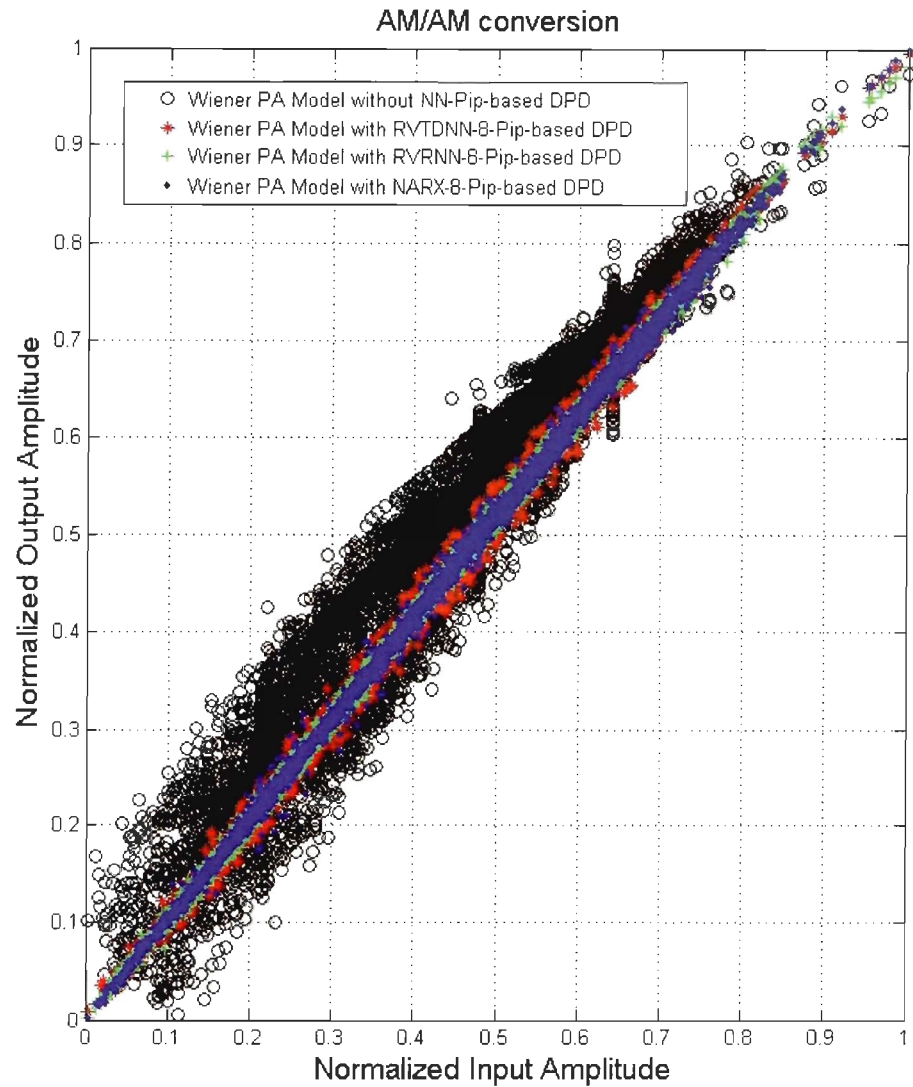


Figure 107: Conversion AM/AM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener.

L'évaluation quantitative des caractéristiques de conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence, représentées sur la figure 108, est faite par calcul du paramètre MSE. Les architectures de prédistorsion RVTDDN-8-pip RVRNN-8-pip et NARX-8-pip ont respectivement comme paramètre MSE 0.491, 1.192 et 0.438 par rapport à l'entrée.

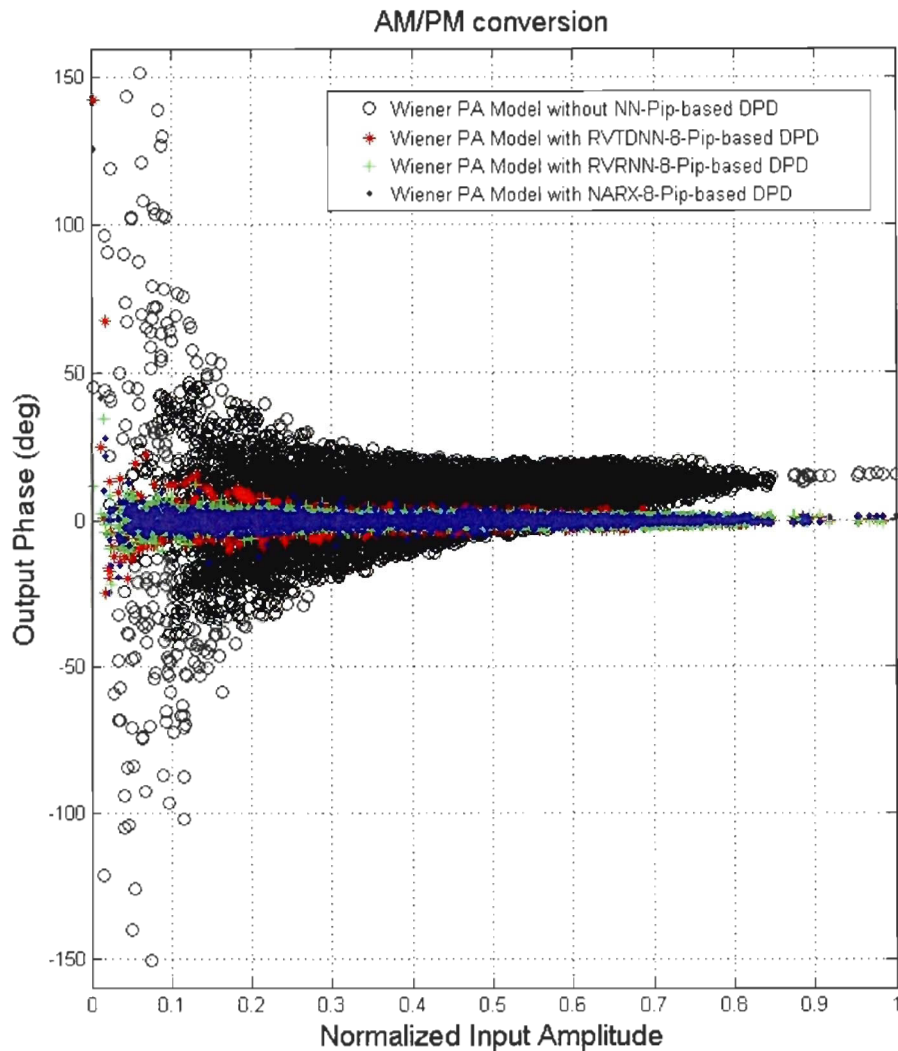


Figure 108: Conversion AM/PM du modèle de référence de Wiener et des architectures de prédistorsion RVT-DNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener.

La caractérisation quantitative des composants cartésiens I et Q de l'entrée et des architectures de prédistorsion RVT-DNN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener prédistorsion, représentées sur la figure 109, est aussi faite par calcul du paramètre MSE. Les architectures de prédistorsion RVT-DNN-8-pip, RVRNN-8-pip et NARX-8-pip ont respectivement comme paramètre MSE des composants cartésiens en phase (I) 2.371×10^{-5} , 6.010×10^{-5} et 8.191×10^{-5} et en quadrature (Q) 3.594×10^{-5} , 4.510×10^{-5} et 1.220×10^{-5} par rapport au signal d'entrée.

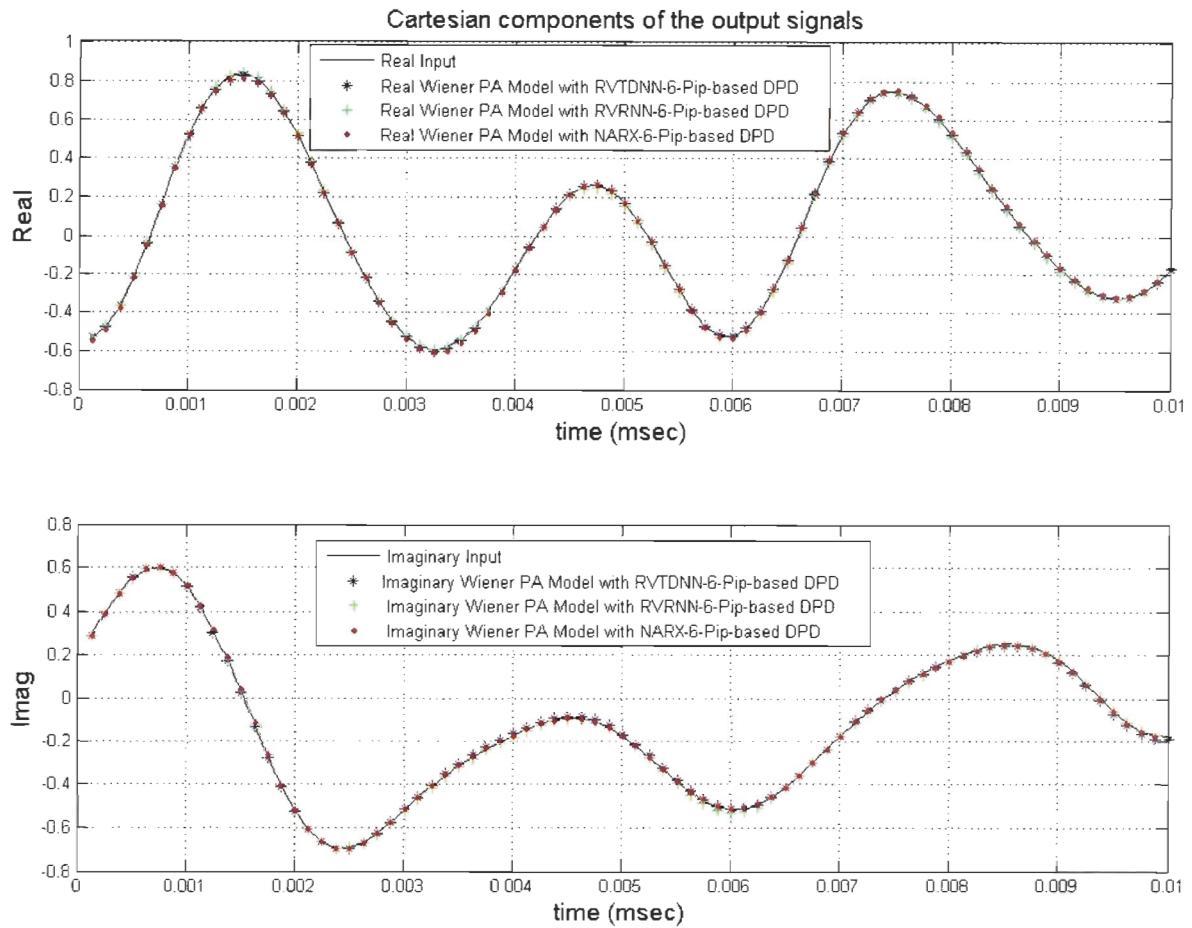


Figure 109: Composants cartésiens I et Q de l'entrée et des sorties des architectures de prédistorsion RVTDDN-8-pip, RVRNN-8-pip et NARX-8-pip en série avec le modèle de référence de Wiener et composant cartésien Q du modèle de référence de Wiener.

Les spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVTDDN-8-pip en série avec le modèle de référence de Wiener sont représentés sur la figure 110. Au décalage -1 MHz du canal adjacent gauche et au décalage 1 MHz du canal adjacent droit, la diminution des paramètres ACPR gauche et droit du signal à la sortie du réseau NARX-8-pip, du réseau RVRNN-8-pip et du réseau RVTDDN-8-pip par rapport au signal d'entrée est respectivement de (14.412 dB et 14.562 dB), (14.093 dB et 14.420) et (13.583 dB et 12.735).

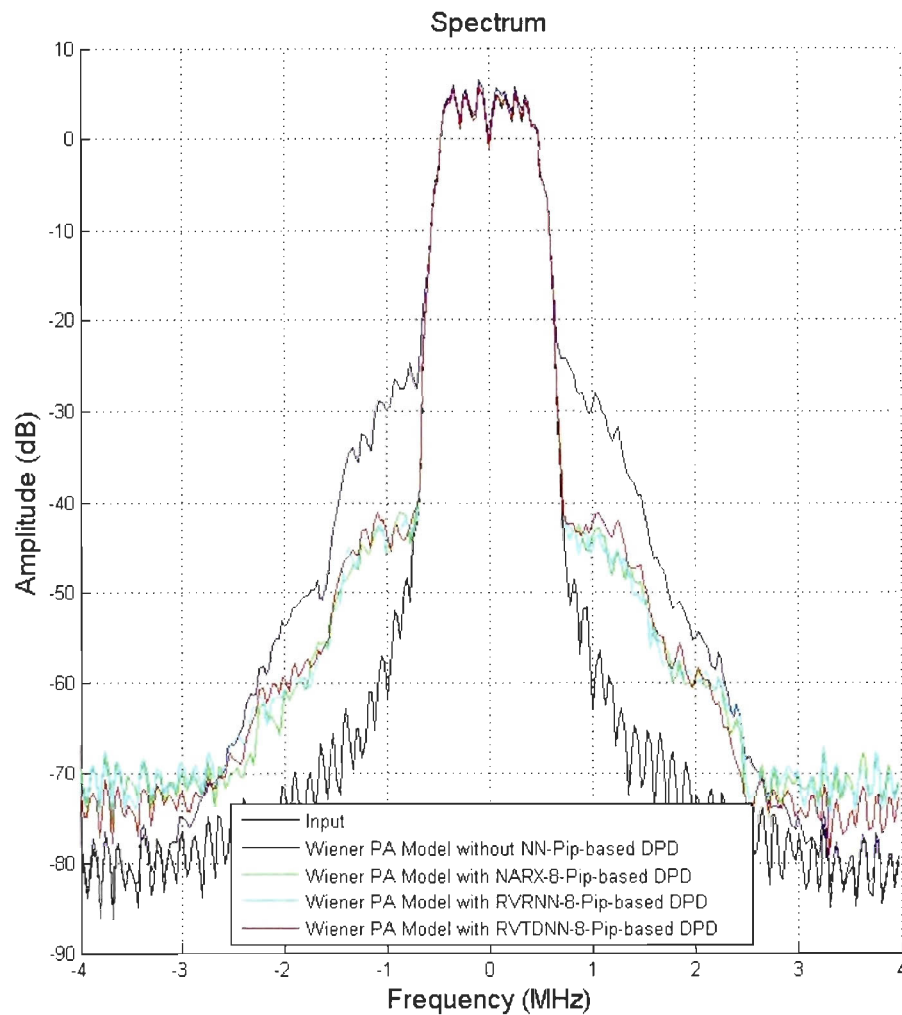


Figure 110: Spectres de l'entrée, des sorties du modèle de référence de Wiener et des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVTDDNN-8-pip en série avec le modèle de référence.

Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVTDDNN-8-pip en série avec le modèle de référence sont représentés sur la figure 111. Les architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVTDDNN-8-pip en série avec le modèle de référence ont respectivement comme paramètre MSE à la 2^{ème} trame environ 0.0607, 0.0566 et 0.0531.

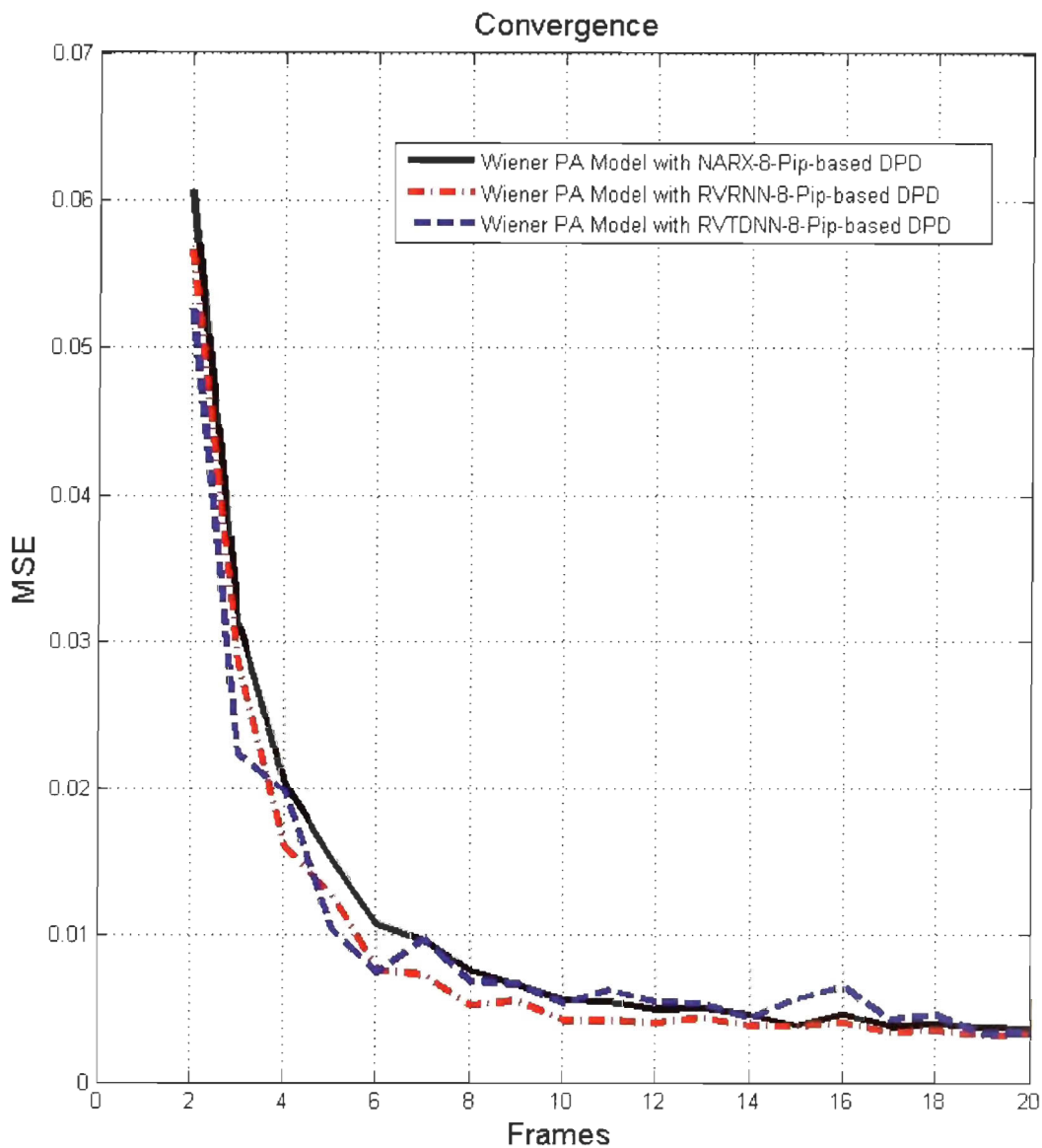


Figure 111: Les paramètres MSE relatifs à la convergence des architectures de prédistorsion NARX-8-pip, RVRNN-8-pip et RVTDNN-8-pip en série avec le modèle de référence.

De l'analyse des résultats de simulation des différentes courbes, découle que l'architecture de prédistorsion NARX-8-pip corrige en général mieux les distorsions dues au modèle de référence de Wiener que celle du réseau RVTDNN-8-pip qui elle-même corrige mieux les distorsions dues au modèle de référence de Wiener que le réseau RVRNN-8-pip. L'architecture de prédistorsion NARX-8-pip a un pourcentage de paramètre EVM (2.193 %) qui est

inférieur à celui du réseau RVRNN-8-pip (2.414 %) qui est lui-même inférieur à celui du réseau RVTDNN-8-pip (2.488 %). Pour les conversions AM/AM, le paramètre MSE de l'architecture de prédistorsion RVTDNN-8-pip a 3.005×10^{-5} qui est plus petit que celui du réseau NARX-8-pip (5.576×10^{-5}) qui est lui-même plus petit que celui du réseau RVRNN-8-pip (5.581×10^{-5}). Pour les conversions AM/PM, le paramètre MSE de l'architecture de prédistorsion NARX-8-pip a 0.438 qui est plus petit que celui du réseau RVTDNN-8-pip (0.491) qui est lui-même plus petit que celui du réseau RVRNN-8-pip (1.192). Le calcul du paramètre MSE pour la caractérisation du composant cartésien I montre que le réseau RVTDNN-8-pip a un paramètre MSE de 2.371×10^{-5} inférieur à celui du réseau RVRNN-8-pip qui est de 6.010×10^{-5} et du réseau NARX-8-pip d'une valeur de 8.191×10^{-5} . Le calcul du paramètre MSE pour la caractérisation du composant cartésien Q montre que le réseau NARX-8-pip a un paramètre MSE de 1.220×10^{-5} inférieur à celui du réseau RVTDNN-8-pip qui est de 3.594×10^{-5} et du réseau RVRNN-8-pip d'une valeur de 4.510×10^{-5} .

Au décalage -1 MHz du canal adjacent gauche et décalage 1 MHz du canal adjacent droit, le réseau NARX-8-pip (14.412 dB et 14.562 dB) a une meilleure diminution des paramètres ACPR gauche et droit que le réseau RVRNN-8-pip (14.093 dB et 14.420) qui est meilleure que le réseau RVTDNN-8-pip (13.583 dB et 12.735) par rapport au modèle de référence de Wiener. Cependant, à la 2^{ème} trame le réseau RVTDNN-8-pip converge plus vite que le réseau RVRNN-8-pip qui, lui-même, converge plus vite que le réseau NARX-8-pip, car ils ont respectivement comme paramètre MSE 0.0531, 0.0566 et 0.0607. L'analyse des résultats de linéarisation des architectures de prédistorsion NARX- N_1 -pse, RVRNN- N_1 -pse, RVTDNN- N_1 -pse, RVTDNN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip montrent qu'elles corrigent efficacement les déformations introduites par le modèle de référence de Wiener.

Tableau 6: Résultats de simulation de la constellation, des caractéristiques AM/AM et AM/PM, des composants cartésiens I et Q et spectre des différentes architectures de prédistorsion.

Architecture de prédistorsion	Constellations	Caracteristiques		Composants cartésiens		Spectre	
	EVM (%)	MSE d'AM/AM ($\times 10^{-5}$)	MSE d'AM/PM	MSE de Phase (I) ($\times 10^{-5}$)	MSE de Quadrature (Q) ($\times 10^{-5}$)	ACPR Gauche Réduction (dB)	ACPR Droit Réduction (dB)
Modèle de référence	33.205	-	-	-	-	-	-
RVTDNN-6-pse	4.050	13.153	4.781	10.057	37.973	6.127	5.335
RVRNN-6-pse	1.170	0.790	0.095	0.763	0.795	14.670	14.848
NARX-6-pse	0.913	0.727	0.076	0.488	1.174	14.570	14.376
RVTDNN-8-pse	0.934	0.775	0.061	0.459	0.779	15.617	15.343
RVRNN-8-pse	0.509	0.115	0.046	0.160	0.294	18.786	17.774
NARX-8-pse	0.505	0.170	0.032	0.175	0.207	18.440	18.223
RVTDNN-6-pip	2.673	8.475	0.773	11.512	1.879	12.685	12.759
RVRNN-6-pip	3.256	7172.141	0.585	49235.784	21605.619	11.928	11.928
NARX-6-pip	5.203	7465.193	3.686	50086.441	21364.731	5.472	5.239
RVTDNN-8-pip	2.488	3.005	0.491	2.371	3.594	13.583	12.735
RVRNN-8-pip	2.414	5.581	1.192	6.010	4.510	14.093	14.420
NARX-8-pip	2.193	5.576	0.438	8.191	1.220	14.412	14.562

Exception faite pour le réseau NARX-6-pip et le réseau RVTDDN-6-pse qui, en série avec le modèle de PA de Wiener, permettent une diminution des paramètres ACPR gauche et droit respectivement de 6.127 dB et 5.335 dB et 5.472 dB et 5.239 dB de réduction par rapport au modèle de référence. Dans le tableau 6, les architectures NARX-8-pse et RVRNN-8-pse corrigent mieux les distorsions à l'intérieur de la bande EVM dues au modèle PA de Wiener que toutes les autres architectures de prédistorsion. En effet, elles ont respectivement 0.505 % et 0.509 % de paramètre EVM, qui sont une bonne correction des paramètres EVM par rapport au modèle de référence de Wiener (33.205 %). Elles sont suivies par le réseau NARX-6-pse qui a 0.913 % de paramètre EVM. Pour les caractéristiques AM/AM, le réseau RVRNN-8-pse a le paramètre MSE le plus bas, soit 0.115×10^{-5} suivi du réseau NARX-8-pse (0.170×10^{-5}) et du réseau NARX-6-pse (0.727×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau NARX-8-pse a le paramètre MSE le plus bas, soit 0.032 suivi du réseau RVRNN-8-pse (0.046) et du réseau RVTDDN-8-pse (0.061). Pour la prédistorsion numérique du composant cartésien en phase I, le réseau RVRNN-8-pse (0.160×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-8-pse (0.175×10^{-5}) qui est lui-même meilleur que le réseau RVTDDN-8-pse (0.459×10^{-5}). Aussi, pour la prédistorsion numérique du composant cartésien en quadrature Q, le réseau NARX-8-pse a un paramètre MSE de 0.207×10^{-5} qui est respectivement meilleur que ceux du réseau RVRNN-8-pse (0.294×10^{-5}) et du réseau RVTDDN-8-pse (0.779×10^{-5}). Aux décalages -1 MHz du canal adjacent gauche et 1 MHz du canal adjacent droit, l'architecture de prédistorsion NARX-8-pse a une meilleure réduction des paramètres ACPR gauche et droit par rapport au modèle de référence de Wiener qui sont respectivement de 18.440 dB et 18.223 dB que le réseau RVRNN-8-pse (18.786 dB et 17.774 dB) et le réseau RVTDDN-8-pse (15.617 dB et 15.343 dB).

Également, dans le tableau 6, l'architecture NARX-8-pip, corrige mieux les distorsions à l'intérieur de la bande EVM dues au modèle PA de Wiener que toutes les autres architectures pipelinées de prédistorsion, car elle est 2.193 % de paramètre EVM, qui est une bonne correction de paramètre EVM par rapport au modèle PA de Wiener (33.205 %). Pour les caractéristiques AM/AM, le réseau RVTDDN-8-pip a le paramètre MSE le plus bas (3.005×10^{-5}) suivi du réseau NARX-8-pip (5.576×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau NARX-8-pip a le paramètre MSE le plus bas (0.438) suivi du réseau RVTDDN-8-pip (0.491). Pour la prédistorsion numérique du composant cartésien en phase I, le réseau RVTDDN-8-pip a un meilleur paramètre MSE (2.371×10^{-5}) que le réseau RVRNN-8-pip

(6.010×10^{-5}). Aussi, pour la prédistorsion numérique du composant cartésien en quadrature Q, le réseau NARX-8-pip a un paramètre MSE de 1.220×10^{-5} qui est meilleur que celui du réseau RVTDDN-8-pip (3.594×10^{-5}). Aux décalages -1 MHz du canal adjacent gauche et 1 MHz du canal adjacent droit, l'architecture de prédistorsion NARX-8-pip a une meilleure réduction des paramètres ACPR gauche et droit, par rapport au modèle de référence de Wiener, qui sont respectivement de 14.412 dB et 14.562 dB que le réseau RVRNN-8-pip qui sont respectivement de 14.093 dB et 14.420 dB.

Les architectures de prédistorsion pseudo-conventionnelles donnent de meilleurs résultats de simulation que celles avec pipeline, mais ne sont malheureusement pas implantables sur la puce FPGA Virtex-6 de la carte ML605 malgré le fait qu'elles ont une quantité de ressources estimées inférieure à celle disponible sur cette puce FPGA. Leur défaut est qu'elles ont une fréquence maximale d'opération d'environ 42.2 MHz qui est inférieure à la fréquence minimale de traitement d'un échantillon dans cette puce FPGA. Les architectures avec pipeline sont implantables sur la puce FPGA Virtex-6 de la carte ML605, car elles ont une fréquence maximale d'opération de 156.128 MHz et une quantité de ressources estimées inférieure à celle disponible sur cette puce FPGA, mais sont légèrement moins performantes en prédistorsion que celles pseudo-conventionnelles à cause des retards insérés dans la chaîne pour le pipeline.

CHAPITRE 4

CONCLUSION GÉNÉRALE ET PERSPECTIVES

4.1 Conclusion générale

Le travail développé dans ce mémoire est basé sur l'implantation des réseaux de neurones sur la puce FPGA pour la modélisation et la linéarisation des amplificateurs de puissance. En effet, les distorsions dues aux non-linéarités et aux effets mémoire des amplificateurs de puissance créent des remontées spectrales dans les canaux adjacents (ACPR) ainsi que la déformation de la constellation du signal modulé (EVM) dans la bande passante des systèmes de communication sans fil modernes à large bande. Plusieurs approches ont été développées dans le but de corriger ces distorsions par linéarisation des PAs, parmi lesquelles la prédistorsion numérique (DPD). Celle-ci est en voie de devenir l'une des techniques de linéarisation les plus importantes en raison de ses capacités de traitement rapide du signal numérique avec des outils tels que les puces FPGA en remplacement de la technique *feedforward* qui est actuellement la technique de linéarisation la plus utilisée dans les stations de base en RF. Les puces FPGA ont une capacité de traitement supérieure à celle des puces DSP et peuvent augmenter leur vitesse de calcul dans les applications caractérisées par un profond pipeline et un haut parallélisme.

L'objectif du projet a donc été de proposer un système à base de réseaux de neurones implantable sur des puces FPGA, pour la correction dynamique en temps-réel des distorsions causées par les non-linéarités et les effets mémoire des PA des stations de base en RF, à l'intérieur (EVM) et à l'extérieur (ACPR) de la bande de fréquence du signal. Deux types d'algorithmes ont été proposés, un pour la modélisation, car il faut s'assurer que le modèle choisi est capable de reproduire fidèlement le comportement du PA, et l'autre pour la linéarisation du PA par prédistorsion adaptative, soit le modèle de correction de ces distorsions.

Les architectures DPD qui ont été développées sont à base de réseaux de neurones RVTDDN, RVRNN et NARX. Ces réseaux sont fondés sur le perceptron multicouche (MLP), auquel sont ajoutés des couches de retard (TDL) aux entrées, pour qu'elles ne soient pas seulement constituées de valeurs courantes, mais aussi des valeurs précédentes. Les réseaux récurrents RVRNN et NARX tiennent aussi compte des valeurs précédentes de leurs sorties. Ces ar-

chitectures développées ont fait l'objet d'une comparaison quantitative du paramètre EVM et des courbes respectives du spectre de puissance, de constellation, du paramètre MSE, d'évolution temporelle des composants cartésiens en phase I et en quadrature de phase Q et des caractéristiques AM/AM et AM/PM. Elles ont également été implantées sur la puce FPGA Virtex-6 de la carte ML605, en utilisant Xilinx System Generator comme outil de programmation sous l'environnement Matlab/Simulink. Le pipeline a aussi été réalisé dans le but d'optimiser le chemin critique et, par conséquent, d'augmenter la fréquence maximale d'opération (MOF) de chacune de ces architectures. Ainsi, cela a permis de faire fonctionner le système à des fréquences capables de faire une prédistorsion numérique en temps-réel. Les architectures de prédistorsion proposées sont toutes à apprentissage direct. La majorité de ces architectures sont capables de réduire considérablement les distorsions en dehors et dans la bande de fréquence des signaux 3G et 4G à larges bandes. Par ailleurs, le signal de test utilisé pour la validation des résultats est un signal à bande de base modulé 16-QAM de largeur de bande 1.35 MHz, car la technologie LTE supporte aussi 16-QAM comme type de modulation en bande de base.

C'est dans l'optique de s'assurer que les architectures de modélisation proposées reflètent fidèlement le comportement du PA qu'est développé le chapitre 2. Dans ce chapitre, il découle de l'analyse de l'estimation des ressources que la diminution du nombre de bits de données entraîne une diminution du nombre de DSP48E1 et de RAMB36E1, et par conséquent une augmentation de la fréquence maximale d'opération MOF. Le nombre de *Bonded IOB* ne varie pas, quelle que soit la variation du nombre d'entrées et du nombre de neurones à la couche cachée, mais suivant le nombre de bits de données. Les *Flips-Flops* et les LUTs augmentent avec l'ajout du nombre d'entrées et du nombre de neurones à la couche cachée. La fréquence MOF a augmenté de plus de 4 fois pour toutes les architectures de modélisation pipelinées RVTDNN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip, soit 156.128 MHz par rapport aux architectures de modélisation pseudo-conventionnelles RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse qui ont chacune une fréquence MOF d'environ 37.4 MHz.

Cette augmentation est due à la réduction du chemin critique. L'uniformité de la fréquence MOF pour les architectures avec pipeline justifie la bonne optimisation et la bonne synchronisation des retards. L'analyse des résultats des architectures de modélisation pseudo-conventionnelles RVTDNN- N_1 -pse, RVRNN- N_1 -pse, NARX- N_1 -pse, NARX- N_1 -pip, RVRNN- N_1 -pip et RVTDNN- N_1 -pip montrent que ces architectures modélisent fidèlement le modèle

de référence de Wiener. L'évaluation des ressources montrent également que les architectures de modélisation pipelinées NARX- N_1 -pip, RVRNN- N_1 -pip et RVDNN- N_1 -pip sont implémentables dans la puce FPGA Virtex-6 XC6VLX240T-1FFG1156 de la carte ML605.

Pour les architectures de modélisation pseudo-conventionnelles, le réseau RVRNN-6-pse et le réseau NARX-8-pse modélisent mieux les distorsions à l'intérieur de la bande EVM dues au modèle de référence de Wiener que toutes les autres architectures de modélisation pseudo-conventionnelles, car elles ont toutes les deux 33.199 % de paramètre EVM, qui est la valeur la plus proche du paramètre EVM du modèle de référence de Wiener (33.205 %). De même, pour les caractéristiques AM/AM, le réseau NARX-8-pse a le paramètre MSE le plus bas (0.581×10^{-5}) suivi du réseau RVRNN-6-pse (0.670×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau RVDNN-8-pse a le paramètre MSE le plus bas (0.029) suivi du réseau RVRNN-8-pse 0.080. Pour la modélisation du composant cartésien en phase I, le réseau RVRNN-6-pse (0.459×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-8-pse (0.485×10^{-5}). Pour la modélisation du composant cartésien en quadrature Q, le réseau RVRNN-6-pse a un paramètre MSE de 0.186×10^{-5} qui est meilleur que celui du réseau RVDNN-8-pse (0.501×10^{-5}). Par conséquent, le réseau RVRNN-6-pse avec un paramètre MSE de (0.322×10^{-5}) modélise mieux le spectre du modèle de référence de Wiener que le réseau RVDNN-8-pse (0.536×10^{-5}). En définitive, parmi toutes les architectures de modélisation pseudo-conventionnelles, le réseau RVRNN-6-pse modélise mieux les distorsions du modèle de référence de Wiener.

Il permet aussi d'utiliser moins de ressources dans la puce FPGA Virtex-6 de la carte ML605 que le réseau RVDNN-8-pse. Par conséquent, le fait de passer du réseau RVDNN-8-pse au réseau RVRNN-6-pse a permis de réduire le nombre de neurones à la couche cachée et de diminuer considérablement les ressources en ayant de meilleures performances, grâce au fait que le réseau RVRNN-6-pse tient compte de l'historique des sorties. En terme de ressources, le réseau RVRNN-6-pse utilise 31.73 % de RAMB36E1 et 37.50 % de DSP48E1, alors que le réseau RVDNN-8-pse utilise 42.30 % de RAMB36E1 et 41.66 % de DSP48E1.

Pour les architectures de modélisation avec pipeline le réseau RVRNN-8-pip modélise mieux les distorsions à l'intérieur de la bande EVM dues au modèle de référence de Wiener que toutes les autres architectures de modélisation avec pipeline, car elle a 33.198 % de paramètre EVM, qui est la valeur la plus proche du paramètre EVM du modèle de référence de Wiener (33.205 %). De même, pour les caractéristiques AM/AM, le réseau RVRNN-8-pip

a le paramètre MSE le plus bas (0.926×10^{-5}) suivi du réseau RVTDDN-8-pip (2.849×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau RVRNN-8-pip a le paramètre MSE le plus bas (0.068) suivi du réseau RVTDDN-8-pip (0.328). Pour la modélisation du composant cartésien en phase I, le réseau RVRNN-8-pip (1.169×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-6-pip (2.983×10^{-5}). Aussi, pour la modélisation du composant cartésien en quadrature Q, le réseau RVRNN-8-pip a un paramètre MSE de 0.544×10^{-5} qui est meilleur que celui du réseau RVTDDN-8-pip (2.413×10^{-5}). Par conséquent, le réseau RVRNN-8-pip avec un paramètre MSE de (0.857×10^{-5}) modélise mieux le spectre du modèle de référence de Wiener que le réseau RVTDDN-8-pip (2.969×10^{-5}). En fin de compte, le réseau RVRNN-8-pip modélise mieux les distorsions dues au modèle de référence de Wiener que toutes les autres architectures de modélisation avec pipeline. Les architectures de modélisation pseudo-conventionnelles donnent de meilleurs résultats de simulation que celles avec pipeline, mais ne sont malheureusement pas implantables sur la puce FPGA Virtex-6 de la carte ML605, malgré le fait qu'elles ont une quantité de ressources estimée inférieure à celle disponible sur cette puce FPGA. Leur défaut est qu'elles ont une fréquence maximale d'opération qui est d'environ 37.4 MHz et est inférieure à la fréquence minimale de traitement d'un échantillon dans cette puce FPGA. Celles avec pipeline sont implantables sur la puce FPGA Virtex-6 de la carte ML605, car elles ont une fréquence maximale d'opération de 156.128 MHz et une quantité de ressources estimée inférieure à celle disponible sur la cette puce FPGA, mais sont moins performantes que celles pseudo-conventionnelles à cause des retards insérés dans la chaîne pour le pipeline.

Après optimisation de ces architectures pour la modélisation des PA, le chapitre 3 est développé pour la linéarisation des PA par prédistorsion numérique. Les architectures de prédistorsion pseudo-conventionnelles ont le même nombre de bascules, de *Bonded IOB*, de RAMB36E1 et de DSP48E1 que celles de modélisation pseudo-conventionnelles. Seuls les tables LUT ont augmenté. La fréquence MOF de l'architecture de prédistorsion RVTDDN-6-pse est de 43.148 MHz, alors que la fréquence MOF de l'architecture de modélisation RVTDDN-6-pse est de 37.440 MHz.

Les architectures de prédistorsion RVTDDN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip ont le même nombre de ressources bascules, IOB, RAMB36E1 et DSP48E1 que leurs correspondantes de modélisation. La fréquence MOF a augmenté de plus de 3.5 fois pour toutes les architectures de prédistorsion RVTDDN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip, soit 156.128

MHz par rapport aux architectures DPD RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse qui ont chacune une fréquence MOF d'environ 42.2 MHz. Cette augmentation est due à la diminution du chemin critique. L'uniformité de la fréquence MOF pour les architectures DPD avec pipeline justifie la bonne optimisation et la bonne synchronisation des retards. Aussi, la fréquence MOF des architectures de modélisation et prédistorsion RVTDNN- N_1 -pip, RVRNN- N_1 -pip et NARX- N_1 -pip reste la même soit 156.128 MHz. Par contre, la fréquence MOF des architectures de modélisation et prédistorsion RVTDNN- N_1 -pse, RVRNN- N_1 -pse et NARX- N_1 -pse est d'environ 37.4 MHz pour celles de modélisation et d'environ 42.2 MHz pour celles de prédistorsion numérique.

Pour les architectures de prédistorsion pseudo-conventionnelles, le réseau NARX-8-pse et le réseau RVRNN-8-pse corrigent mieux les distorsions à l'intérieur de la bande EVM dues au modèle de référence de Wiener que toutes les autres architectures de prédistorsion, car elles ont respectivement 0.505 % et 0.509 % de paramètre EVM, qui constituent une bonne correction des paramètres EVM par rapport au modèle de référence de Wiener (33.205 %). Pour les caractéristiques AM/AM, le réseau RVRNN-8-pse a le paramètre MSE le plus bas, soit 0.115×10^{-5} suivi du réseau NARX-8-pse (0.170×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau NARX-8-pse a le paramètre MSE le plus bas, soit 0.032 suivi du réseau RVRNN-8-pse 0.046. Pour la prédistorsion du composant cartésien en phase I, le réseau RVRNN-8-pse (0.160×10^{-5}) a un meilleur paramètre MSE que le réseau NARX-8-pse (0.175×10^{-5}). Aussi, pour la prédistorsion du composant cartésien en quadrature Q, le réseau NARX-8-pse a un paramètre MSE de 0.207×10^{-5} qui est meilleur que celui du réseau RVRNN-8-pse (0.294×10^{-5}). Aux décalages -1.6 MHz du canal adjacent gauche et 1.6 MHz du canal adjacent droit, l'architecture de prédistorsion NARX-8-pse a une meilleure réduction des paramètres ACPR gauche et droit par rapport au modèle de référence de Wiener, qui sont respectivement de 18.440 dB et 18.223 dB, que le réseau RVRNN-8-pse qui sont respectivement de 18.786 dB et 17.774 dB. Tout bien considéré, parmi toutes les architectures de prédistorsion pseudo-conventionnelles, le réseau NARX-8-pse corrige mieux les distorsions du modèle de référence de Wiener. Par conséquent, le fait qu'il tienne compte de l'historique des sorties avec une profondeur de la mémoire aux sorties égale à celle des entrées, le réseau NARX-8-pse a de meilleures performances de prédistorsion numérique que toutes les autres architectures de prédistorsion pseudo-conventionnelles.

Pour les architectures de prédistorsion avec pipeline, le réseau NARX-8-pip corrige mieux

les distorsions à l'intérieur de la bande EVM dues au modèle de référence de Wiener que toutes les autres architectures de prédistorsion, car elle est de 2.193 % de paramètre EVM, qui constitue une bonne correction des paramètres EVM par rapport au modèle de référence de Wiener (33.205 %). Pour les caractéristiques AM/AM, le réseau RVTDDN-8-pip a le paramètre MSE le plus bas (3.005×10^{-5}) suivi du réseau NARX-8-pip (5.576×10^{-5}). Par contre, pour les caractéristiques AM/PM, le réseau NARX-8-pip a le paramètre MSE le plus bas (0.438) suivi du réseau RVTDDN-8-pip 0.491. Pour la prédistorsion du composant cartésien en phase I, le réseau RVTDDN-8-pip (2.371×10^{-5}) a un meilleur paramètre MSE que le réseau RVRNN-8-pip (6.010×10^{-5}). Aussi, pour la prédistorsion du composant cartésien en quadrature Q, le réseau NARX-8-pip a un paramètre MSE de 1.220×10^{-5} qui est meilleur que celui du réseau RVTDDN-8-pip (3.594×10^{-5}). Aux décalages -1.6 MHz du canal adjacent gauche et 1.6 MHz du canal adjacent droit, l'architecture de prédistorsion du réseau NARX-8-pip a une meilleure réduction des paramètres ACPR gauche et droit par rapport au modèle de référence de Wiener, qui sont respectivement de 14.412 dB et 14.562 dB, que le réseau RVRNN-8-pip qui sont respectivement de 14.093 dB et 14.420 dB. Tout bien pesé, le réseau NARX-8-pip pré-distord mieux les distorsions dues au modèle de référence de Wiener que toutes les autres architectures DPD avec pipeline.

Les architectures de prédistorsion pseudo-conventionnelles donnent de meilleurs résultats de simulation que celles avec pipeline, mais ne sont malheureusement pas implantables sur la puce FPGA Virtex-6 de la carte ML605, malgré le fait qu'elles ont une quantité de ressources estimée inférieure à celle disponible sur cette puce FPGA. Leur défaut est qu'elles ont une fréquence maximale d'opération d'environ 42.2 MHz qui est inférieure à la fréquence minimale de traitement d'un échantillon dans cette puce FPGA. Celles avec pipeline sont implantables sur la puce FPGA, car elles ont une fréquence maximale d'opération de 156.128 MHz et quantité de ressources estimée inférieure à celle disponible sur cette puce FPGA, mais sont moins performantes que celles pseudo-conventionnelles à cause des retards insérés dans la chaîne pour le pipeline.

Pour la co-simulation, la fréquence maximale d'opération (MOF) est passée d'environ 37.4 MHz pour toutes les architectures de modélisation pseudo-conventionnelles à 156.128 MHz pour celles avec pipeline. Aussi, la fréquence MOF est passée d'environ 42.2 MHz pour toutes les architectures de prédistorsion pseudo-conventionnelles à 156.128 MHz pour celles avec pipeline. Cette optimisation a permis l'implantation par l'interface *JTAG Hardware co-simulation*

des architectures de modélisation et de prédistorsion avec pipeline à 100 MHz qui est la fréquence maximale de co-simulation disponible sur la puce FPGA Virtex-6 XC6VLX240T-1FFG1156 de la carte d'évaluation ML605.

4.2 Perspectives

Comme perspectives, il est possible d'améliorer la fréquence maximale d'opération en trouvant le moyen de réduire davantage le nombre de bits de données de telle sorte que chaque multiplication n'utilise qu'un DSP48E1 au lieu de 2 comme développé dans ce projet.

Il serait intéressant d'explorer la possibilité d'implanter les tangentes hyperboliques à l'aide de fonctions d'approximation qui les modélise fidèlement au lieu de larges tables LUT.

Au lieu d'utiliser l'algorithme du gradient, l'algorithme de Levenberg-Marquardt pourrait être utilisé dans le but d'avoir un apprentissage un peu plus stable avec moins de fluctuations.

Il est envisageable que les différents algorithmes de ce projet soit testés dans un environnement réel de télécommunication sans fil des stations de base 3G et 4G. Ils pourraient par exemple être utilisés dans la carte MSDPD (*Mixed Signal Digital Prédistorsion System Board*) qui est une plateforme d'évaluation de prédistorsion numérique. Elle pourrait éventuellement être connectée à la carte ML605 dans lequel seront implantés les algorithmes de modélisation et de prédistorsion développés. La carte MSDPD elle-même possède un convertisseur numérique-analogique (DAC) et analogique-numérique (ADC). Les entrées de la carte MSDPD comprennent des données numériques à bande de base et cette carte est conçue pour fonctionner avec un PA externe et un réseau RF.

BIBLIOGRAPHIE

- Abusaid, O. M. (2010). *Network Planning and Dimensioning Process in WCDMA*. Master Thesis, Michigan State University, Michigan, USA.
- Aladren, L., Garcia-Ducar, P., de Mingo, J., Carro, P. L., et Sanchez-Perez, C. (2012). Digital Predistortion Using Non-stationary Windowed Sequences for LTE Power Amplifier Linearization. In *2012 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–3, Montréal, Canada.
- Alghanim, A., Lees, J., Williams, T., Benedikt, J., et Tasker, P. (2007). Investigation of Electrical Base-band Memory Effects in High-power 20W LDMOS Power Amplifiers. In *2007 EuMC European Microwave Conf.*, pages 48–51, Munich, Germany.
- Azam, S., Jonsson, R., et Wahab, Q. (2008). Designing, Fabrication and Characterization of Power Amplifiers Based on 10-Watt SiC MESFET & GaN HEMT at Microwave Frequencies. In *2008 EuMC European Microwave Conf.*, pages 444–447, Amsterdam, The Netherlands.
- Ba, S. N. (2009). *Efficient Digital Baseband Predistortion for Modern Wireless Handsets*. Ph.D. Thesis, Georgia Institute of Technology, Georgia, USA.
- Bahoura, M. et Ezzaidi, H. (2011). FPGA-Implementation of Parallel and Sequential Architectures for Adaptive Noise Cancelation. *Circuits, Systems & Signal Processing*, 30(6) :1521–1548.
- Bahoura, M. et Park, C.-W. (2011a). FPGA-Implementation of an Adaptive Neural Network for RF Power Amplifier Modeling. In *2011 9th IEEE Int. NEWCAS Conf.*, pages 29–32, Bordeaux, France.
- Bahoura, M. et Park, C.-W. (2011b). FPGA-Implementation of High-Speed MLP Neural Network. In *2011 18th IEEE Int. Electronics, Circuits & Systems Conf.*, pages 426–429, Beirut, Lebanon.
- Bahoura, M. et Park, C.-W. (2012). FPGA-implementation of Dynamic Time Delay Neural

- Network for Power Amplifier Behavioral Modeling. *Journal Analog Integrated Circuits and Signal Processing*, 73(3) :819–828.
- Barbieri, A., Colavolpe, G., Foggi, T., Forestieri, E., et Prati, G. (2010). OFDM versus Single-Carrier Transmission for 100 Gbps Optical Communication. *Journal Lightwave Technology*, 28(17) :2537–2551.
- Baudoin, G. et al. (2007). *Radiocommunications Numériques - Tome 1 - Principes, Modélisation et Simulation*. Dunod, Paris, France, 2nd edition.
- Bennadji, A. (2006). *Implémentation de Modèles Comportementaux d'Amplificateurs de Puissance dans des Environnements de Simulation Système et Co-simulation Circuit Système*. Ph.D. Thesis, Université de Limoges, Limoges, France.
- Bensmida, S. (2005). *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, Paris, France.
- Bensmida, S., Mimis, K., Morris, K. A., Beach, M. A., McGeehan, J. P., Lees, J., Benedikt, J., et Tasker, P. J. (2012). Overlapped Segment Piece-wise Polynomial Pre-distortion for the Linearisation of Power Amplifiers in the Presence of High PAPR OFDM Signals. In *2012 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–3, Montréal, Canada.
- Boumaiza, S., Gauthier, J., et Ghannouchi, F. M. (2003). Dynamic Electro-thermal Behavioral Model for RF Power Amplifiers. In *2003 IEEE MTT-S Int. Microwave Symp. Dig.*, volume 1. pages 351–354, Philadelphia, PA, USA.
- Boumaiza, S. et Mkaem, F. (2009). Wideband RF Power Amplifier Predistortion Using Real-Valued Time-Delay Neural Networks. In *2009 EuMC European Microwave Conf.*, pages 1449–1452, Rome, Italy.
- Braithwaite, R. N. (2012). Reducing Estimator Biases due to Equalization Errors in Adaptive Digital Predistortion Systems for RF Power Amplifiers. In *2012 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–3, Montréal, Canada.

- Braithwaite, R. N. et Hunton, M. J. (2010). A Positive Feedback Pilot System for a Wide Bandwidth Feedforward Amplifier. In *2010 EuMC European Microwave Conf.*, pages 1778–1781, Paris, France.
- Cao, Y. et Zhang, Q.-J. (2009). A New Training Approach for Robust Recurrent Neural-Network Modeling of Nonlinear Circuits. *IEEE Trans. Microwave Theory & Tech.*, 57(6) :1539–1553.
- Choi, S., Jeong, E.-R., et Lee, Y. H. (2007). A Direct Learning Structure for Adaptive Polynomial-Based Predistortion for Power Amplifier Linearization. In *2007 65th IEEE Vehicular Technology Conf.*, pages 1791–1795, Dublin, Ireland.
- Chun, S. H., Kim, Y. H., Choi, K. J., Kim, J. Y., Kim, J. H., Kim, D., et Hahn, C. K. (2010). Adaptive Digital Pre-distortions Based on Affine Projection Algorithm for WCDMA Power Amplifier Applications. In *2010 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1094–1097, Anaheim, California, USA.
- Daho, F. B., Neveux, G., Decroze, C., Vaudon, P., Mouhamadou, M., et Carsenat, D. (2011). Performance Measurements of Modified LINC Transmitter for Wireless Applications. In *2011 IEEE INMMIC Integrated Nonlinear Microwave and Millimetre-Wave Circuits Workshop*, pages 1-4, Vienna, Austria.
- Dardenne, S. (2005). *Amélioration de la Linéarité des Amplificateurs de Puissance par Injection de Composantes Basses Fréquences et d'Intermodulation, pour des Applications de Radiocommunications Mobiles*. Ph.D. Thesis, Université de Poitiers, Poitiers, France.
- Ding, L., Yang, Z., et Gandhi, H. (2012). Concurrent Dual-band Digital Predistortion. In *2012 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–3, Montréal, Canada.
- Ding, L., Zhou, G. T., Morgan, D. R., Ma, Z., Kenney, J. S., Kim, J., et Giardina, C. R. (2004). A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials. *IEEE Trans. Microwave Theory & Tech.*, 52(1) :159–165.
- Djamai, M., Bachir, S., et Duvanaud, C. (2008). Kalman Filtering Algorithm for on-line Memory Polynomial Predistortion. In *2008 EuMC European Microwave Conf.*, pages 575–578, Amsterdam, The Netherlands.

- Doufana, M. (2009). *Approche par Réseaux de Neurones pour la Linéarisation par Prédistorion Adaptative des Amplificateurs de Puissance RF*. Master Thesis, Université du Québec à Rimouski, Rimouski, Québec, Canada.
- Doufana, M. et Park, C. W. (2008). Neural Network Based Power Amplifier Dynamic Modeling for Wireless Communications. In *2008 ICMMT Int. Microwave & Millimeter Wave Technology Conf.*, volume 1, pages 150–153, Nanjing, China.
- Doufana, M., Park, C.-W., et Bahoura, M. (2010). A Neural Network based On-Line Adaptive Predistorter for Power Amplifier. In *2010 IEEE WAMICON Wireless & Microwave Technology Conf.*, pages 1–4, Melbourne Beach, Florida, USA.
- Fehri, B. et Boumaiza, S. (2011). Systematic Estimation of Memory Effects Parameters in Power Amplifiers' Behavioral Models. In *2011 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–4, Baltimore, USA.
- Forestier, S., Bouysse, P., Quere, R., Mallet, A., Nebus, J. M., et Lapierre, L. (2003). A Dynamic Bias Control Technique of PHEMT SSPA for Optimised PAE and EVM Applied to MQAM Satellite Communication Systems. In *2003 EuMC European Microwave Conf.*, volume 3, pages 1345–1348, Munich, Germany.
- Gandhi, H. et Abbott, W. (2010). A Digital Signal Processing Solution for PA Linearization and RF Impairment Correction for Multi-Standard Wireless Transceiver Systems. In *2010 13th EuMC European Microwave Conf.*, pages 719–722, Paris, France.
- Gilabert, P. L., Cesari, A., Montoro, G., Bertran, E., et Dilhac, J.-M. (2008). Multi-Lookup Table FPGA Implementation of an Adaptive Digital Predistorter for Linearizing RF Power Amplifiers with Memory Effects. *IEEE Trans. Microwave Theory & Tech.*, 56(2) :372–384.
- Golio, M. et Golio, J. (2008). *RF and Microwave Circuits, Measurements, and Modeling*. CRC, Phoenix, Arizona, USA, 2nd edition.
- Guan, L. et Zhu, A. (2010). Low-Cost FPGA Implementation of Volterra Series-Based Digital Predistorter for RF Power Amplifiers. *IEEE Trans. Microwave Theory & Tech.*, 58(4) :866–872.

- Guan, L. et Zhu, A. (2011a). Dual-loop Model Extraction for Digital Predistortion of Wideband RF Power Amplifiers. *IEEE Microwave & Wireless Components Letters*, 21(9) :501–503.
- Guan, L. et Zhu, A. (2011b). Simplified Dynamic Deviation Reduction-Based Volterra Model for Doherty Power Amplifiers. In *2010 INMMIC int. Integrated Nonlinear Microwave & Millimeter-Wave Circuits workshop*, pages 1–4, Göteborg, Sweden.
- Halder, A. et Chatterjee, A. (2005). Low-cost Alternate EVM Test for Wireless Receiver Systems. In *2005 IEEE VLSI Test Symp.*, pages 255–260, Palm Springs, CA, USA.
- Hammi, O. et Ghannouchi, F. M. (2009). Twin Nonlinear Two-Box Models for Power Amplifiers and Transmitters Exhibiting Memory Effects With Application to Digital Predistortion. *IEEE Microwave & Wireless Components Letters*, 19(8) :530–532.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice-Hall, Upper Saddle River, NJ, USA, 3th edition.
- Helaoui, M. et Ghannouchi, F. M. (2010). Linearization of Power Amplifiers Using the Reverse MM-LINC Technique. *IEEE Trans. Circuits & Systems II : Express Briefs*, 57(1) :6–10.
- Hwangbo, H., Jung, S. C., Yang, Y., Park, C. S., Kim, B. S., et Nah, W. (2006). Power Amplifier Linearization Using an Indirect-Learning-Based Inverse TDNN Model. *Microwave Journal*, 49(11) :76–88.
- Jang, D.-H., Choi, J.-H., et Kim, J.-H. (2010). Asymmetric Doherty Power Amplifier with Optimized Characteristics in Output Power Back-Off Range between 6 dB and 10 dB. In *2010 EuMC European Microwave Conf.*, pages 870–873, Paris, France.
- Jennings, D. J. et McGeehan, J. P. (1998). A High-efficiency RF Transmitter using VCO-derived Synthesis : CALLUM. In *1998 IEEE RAWCON Radio and Wireless Conf.*, pages 137–140, Colorado, USA.
- Jeong, Y.-C., Ahn, D., Kim, C.-D., et Chang, I.-S. (2006). Feedforward Amplifier using Equal Group-Delay Signal Canceller. In *2006 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1530–1533, San Francisco, CA, USA.

- Kolinko, P. V. et Larson, L. E. (2009). Background Estimation of Power Amplifier Nonlinearities for OFDM Signals. In *2009 IEEE RFIC Radio Frequency Integrated Circuits Symp.*, pages 405–408, Boston, MA, USA.
- Ku, H. et Kenney, J. S. (2003). Behavioral Modeling of Nonlinear RF Power Amplifiers Considering Memory Effects. *IEEE Trans. Microwave Theory & Tech.*, 51(12) :2495–2504.
- Kwan, A., Hammi, O., Helaoui, M., et Ghannouchi, F. M. (2010). High Performance Wideband Digital Predistortion Platform for 3G+ Applications with Better than 55dBc over 40 MHz Bandwidth. In *2010 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1082–1085, Anaheim, California, USA.
- Kwon, S., Kim, M., Jung, S., Jeong, J., Lim, K., Van, J., Cho, H., Kim, H., Nah, W., et Yang, Y. (2009). Inverted-load Network for High-power Doherty Amplifier. *2009 IEEE Microwave Magazine*, 10(1) :93–98.
- Liszewski, J., Schubert, B., Keusgen, W., et Kortke, A. (2011). Low-Complexity FPGA Implementation of Volterra Predistorters for Power Amplifiers. In *2011 IEEE PAWR Topical Power Amplifiers for Wireless and Radio Applications Conf.*, pages 41–44, Phoenix, Arizona, USA.
- Liu, T., Boumaiza, S., et Ghannouchi, F. M. (2004). Dynamic Behavioral Modeling of 3G Power Amplifiers Using Real-valued Time-delay Neural Networks. *IEEE Trans. Microwave Theory & Tech.*, 52(3) :1025–1033.
- Liu, T., Ye, Y., Fan, Z., Zeng, X., et Ghannouchi, F. M. (2008a). Linearization of Wideband RF Doherty Power Amplifiers with Complex Dynamic Nonlinearities. In *2008 ChinaCom Int. Communications and Networking Conf.*, pages 974 – 977, Hangzhou, China.
- Liu, T., Ye, Y., Zeng, X., et Ghannouchi, F. M. (2008b). Memory Effect Modeling of Wideband Wireless Transmitters Using Neural Networks. In *2008 IEEE ICCSC Int. Circuits & Systems for Communications Conf.*, pages 703–707, Shanghai, China.
- Liu, Y.-J., Zeng, R., Cao, T., Zhou, B.-H., Zhou, J., et Liu, Y.-N. (2011). Up-Converted Dual-Envelope Injection Enhanced Digital Predistortion for Inverse Class-E Power Amplifier

- Linearization. In *2011 EuMIC European Microwave Integrated Circuits Conf.*, pages 280–283, Manchester, United Kingdom.
- Marsalek, R. (2003). *Contribution à la Linéarisation des Amplificateurs de Puissance par Prédistorsion Numérique Adaptative en Bande de Base*. Ph.D. Thesis, Université de Marne la Vallée, Paris, France.
- Matlab (2012). *Communications System Toolbox*. Reference r2012a (v 5.2), USA.
- Mellein, R. K. H. (2007). *UMTS Introduction and Measurement*. Rohde&Schwarz, Munchen, Germany, 4th edition.
- Minhui, Z. (2006). *Implémentation d'un Système de Réseaux de Neurones sur un FPGA pour le Contrôle d'un Hélicoptère à Quatres Rotors*. Master Thesis, The Royal Military College of Canada, Ottawa, Canada.
- Mkadem, F. (2010). *Behavioural Modeling and Linearization of RF Power Amplifier using Artificial Neural Networks*. Master Thesis, University of Waterloo, Waterloo, Ontario, Canada.
- Mkadem, F., Ayed, M. B., Boumaiza, S., Wood, J., et Aaen, P. (2010). Behavioral Modeling and Digital Predistortion of Power Amplifiers with Memory using Two Hidden Layers Artificial Neural Networks. In *2010 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 656–659, Anaheim, California, USA.
- Mkadem, F., Boumaiza, S., Staudinger, J., et Wood, J. (2011). Systematic Pruning of Volterra Series using Wiener G-functionals for Power Amplifier and Predistorter Modeling . In *2011 EuMIC European Microwave Integrated Circuits Conf.*, pages 482–485, Manchester, United Kingdom.
- Moon, J. et Kim, B. (2010). Wideband Digital Feedback Predistortion Employing Segmented Memory Compensation for Linearization of Doherty Amplifier. In *2010 EuMC European Microwave Conf.*, pages 727–730, Paris, France.
- Nader, C., Landin, P. N., Moer, W. V., Bjorsell, N., Handel, P., et Isaksson, M. (2011). Peak-to-average Power Ratio Reduction Versus Digital Pre-distortion in OFDM Based Systems. In *2011 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–4, Baltimore, Maryland, USA.

- Ntouné Ntouné, R. S., Bahoura, M., et Park, C.-W. (2012a). FPGA-Implementation of Pipelined Neural Network for Power Amplifier Modeling. In *2012 10th IEEE Int. NEWCAS Conf.*, pages 109–112, Montréal, Canada.
- Ntouné Ntouné, R. S., Bahoura, M., et Park, C.-W. (2012b). Power Amplifier Behavioral Modeling by Neural Networks and their Implementation on FPGA. In *2012 76th IEEE Vehicular Technology Conf.*, Québec, Canada.
- Poitau, G., Birafane, A., et Kouki, A. (2004). Experimental Characterization of LINC Outphasing Combiners' Efficiency and Linearity. In *2004 IEEE RAWCON Radio and Wireless Conf.*, pages 87–90, Atlanta, GA, USA.
- Rahman, R. (2011). *CPM-SC-IFDMA A Power Efficient Transmission Scheme for Uplink LTE*. Master of science in electrical engineering and computer science, University of Kansas, Kansas, USA.
- Rawat, M., Rawat, K., et Ghannouchi, F. M. (2010). Adaptive Digital Predistortion of Wireless Power Amplifiers/Transmitters using Dynamic Real-Valued Focused Time-Delay Line Neural Networks. *IEEE Trans. Microwave Theory & Tech.*, 58(1) :95–104.
- Salch, A. A. M. (1981). Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers. *IEEE Trans. Communications*, 29(11) :1715–1720.
- Shan, W. et Sundstrom, L. (2002). Effects of Anti-aliasing Filters in Feedback Path of Adaptive Predistortion. In *2002 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 469–472, Seattle, WA, USA.
- Subramaniam, K. (2005). *Radio Resource Management in UMTS-WCDMA Systems*. Ph.D. Thesis, North Carolina State University, North Carolina, USA.
- Tabatabai, F. et Al-Raweshidy, H. S. (2007). Feedforward Linearization Technique for Reducing Nonlinearity in Semiconductor Optical Amplifier. *Journal Lightwave Technology*, 25(9) :2667–2674.
- Tornblad, O., Wu, B., Dai, W., Blair, C., Ma, G., et Dutton, R. W. (2007). Modeling and Measurements of Electrical and Thermal Memory Effects for RF power LDMOS. In *2007 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 2015–2018, Honolulu, Hawaii.

- Woo, Y. Y., Yi, J., Yang, Y., et Kim, B. (2003). SDR Transmitter Based on LINC Amplifier with Bias Control. In *2003 IEEE MTT-S Int. Microwave Symp. Dig.*, volume 3, pages 1703–1706, Philadelphia, PA, USA.
- Xilinx (2012a). *Virtex-6 Family Overview*. Product specification ds150 (v2.4), USA.
- Xilinx (2012b). *Virtex-6 FPGA ML605 Evaluation Kit*. High-performance, high-speed FPGA design platform, USA.
- Ye, Y., Liu, T., Fan, Z., Zeng, X., He, J., et Ghannouchi, F. M. (2008a). Accurate Modeling of Wideband RF Doherty Power Amplifiers Using Dynamic Nonlinear Models. In *2008 ICMMT Int. Microwave & Millimeter Wave Technology Conf.*, volume 3, Nanjing, China.
- Ye, Y., Liu, T., et Ghannouchi, F. M. (2008b). Linear and Nonlinear Memory Effects of RF Power Amplifiers. In *2008 APMC Asia-Pacific Microwave Conf.*, pages 1–4, Hong Kong, China.
- Yu, C., Guan, L., et Zhu, A. (2011). Band-limited Volterra Series-based Behavioral Modeling of RF Power Amplifiers. In *2012 IEEE MTT-S Int. Microwave Symp. Dig.*, pages 1–3, Montréal, Canada.
- Zhu, A., Pedro, J. C., et Cunha, T. R. (2007). Pruning the Volterra Series for Behavioral Modeling of Power Amplifiers Using Physical Knowledge. *IEEE Trans. Microwave Theory & Tech.*, 55(5) :813–821.
- Zhu, A., Pedro, J. C., et J.Brazil, T. (2006). Dynamic Deviation Reduction-Based Volterra Behavioral Modeling of RF Power Amplifiers. *IEEE Trans. Microwave Theory & Tech.*, 54(12) :4323–4332.

