DPD using Deep Learning

Contents

1. INTRODUCTION

1.1 MOTIVATION

Digital Pre Distortion (DPD) is one of the most effective techniques of PA linearization. In this technique, a digital non-linear block, known as a Predistorter, is used in front of the Analog to Digital (A/D) component. The role of the Predistorter is to distort the signal in a way that will be, in turn, compensated by the PA. Ideally, the total response of the Predistorter and PAs will be linear up to some saturation voltage.

1.2 OBJECTIVE

The aim of this project is to reduce am/am, am/pm distortions and OOB emission caused by the nonlinearity of the power amplifier. Using both classical and deep learning methods, we will design a digital pre distorter that preprocess a signal before entering a nonlinear power amplifier.

2. LITERATURE REVIEW

2.1 PA MODEL

Nonlinear effects – AM/AM, AM/PM, OOB emission

As we learn from previous literature [1], Power amplifiers (PAs), which are inherently nonlinear systems, are essential components in communication systems. Moreover, PAs also exhibit memory effects. This means that, the current output of the PA depends not only on the current input, but also on past input values, and makes the power amplifier a nonlinear system with memory. Different models, such as the Wiener model and the Hammerstein model, represent this kind of system as a serial two-block model, as described in figure 3.1.a. The first block is a linear FIR filter that deals with linear memory effects, and the second block accounts for the nonlinear effects caused by the PA.

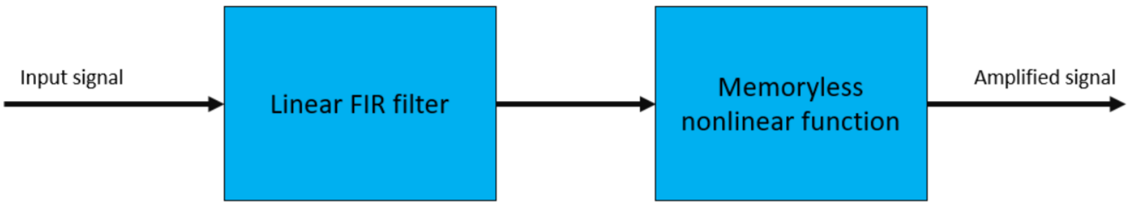

Figure 3.1.a – Block diagram of a two-block PA model

The nonlinearity causes in-band distortion and a spectral regrowth, which in turn leads to interference and violations of the out-of-band emission requirements. The use of different transmission formats, such as wideband Code Division Multiple Access (CDMA) or Orthogonal Frequency Division Multiplexing (OFDM), which are known to have high peak to average power ratios, increases the risk of using voltages that are close to the PAs saturation point, as this will lead to a severe distortion, as mentioned above. For this reason, PA linearization methods have gained popularity and increasing interest in recent years [2].

The PA nonlinearity may be characterized in many ways. In this work, we will concentrate on three types of distortions of the output signal: amplitude distortion (also referred to as Amplitude Modulation/ Amplitude Modulation (AM/AM) distortion), phase distortion (also referred to as Amplitude Modulation/ Phase Modulation (AM/PM)) and out of band emission (OOB emission).

AM/AM is the relation between the amplitude of the input signal and the amplitude of the output signal, which ideally should be linear, but due to non-linear components in the PA, is usually nonlinear. An illustration, of the AM/AM distortion, measured on the PA we used throughout this work, is given in figure-3.1.b, where the output amplitude is shown vs input signal amplitude.

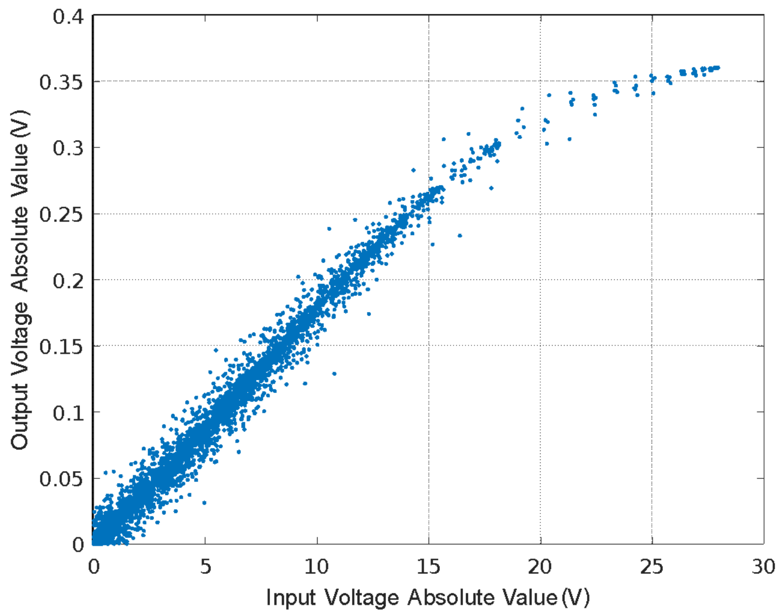

Figure 3.1.b – illustration of AM/AM distortion caused by our lab's nonlinear PA

AM/PM is the relation between the amplitude of the input signal and the phase difference between the input signal and the output signal. Ideally, the phase difference should not change while changing input amplitude. Due to nonlinear affects, the deviation usually happens in the lower range of input signal amplitudes. This distortion, has been measured for in the PA at hand, and illustrated in figure-3.1.c, where the phase difference between input and output is shown vs input signal amplitude. A zoomed in version of figure-3.1.c is shown in figure-3.1.d.

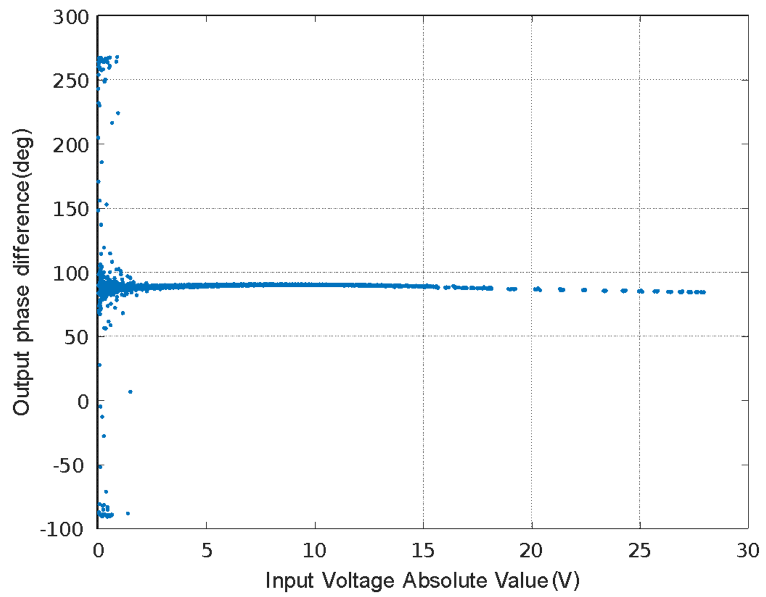

Figure 3.1.c – illustration of AM/PM distortion caused by our lab's nonlinear PA

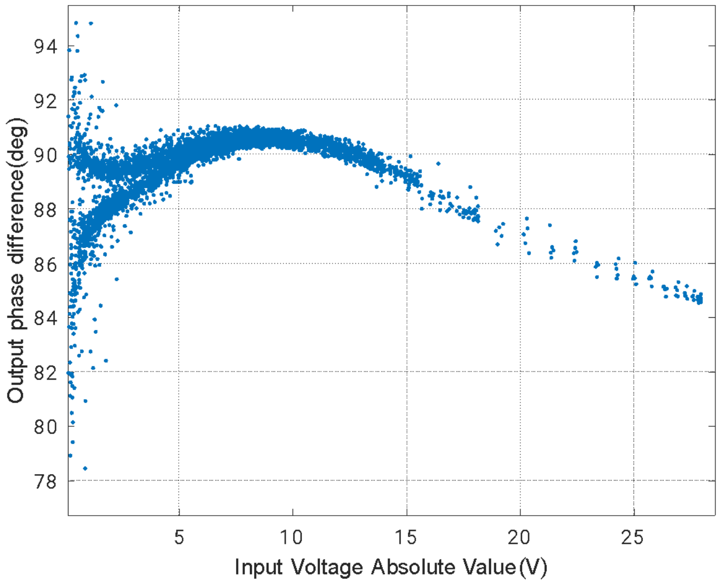

Figure 3.1.d – A zoomed in version of figure-3.1.b

OOB emission is defined as the difference between the bandwidth of the signal after the PA and the bandwidth of the original signal. It is a direct result of the nonlinearity of the PA. For example, if a signal with two fundamental frequencies f_1, f_2, is passed through a nonlinear PA, the output signal will contain new frequency components, which will be in the form of $nf_1 \pm mf_2$, when n and m are integers. Out-of-band frequencies can be suppressed by filters, but in-band frequencies cannot. This distortion, has been measured, and illustrated in figure-3.1.e.

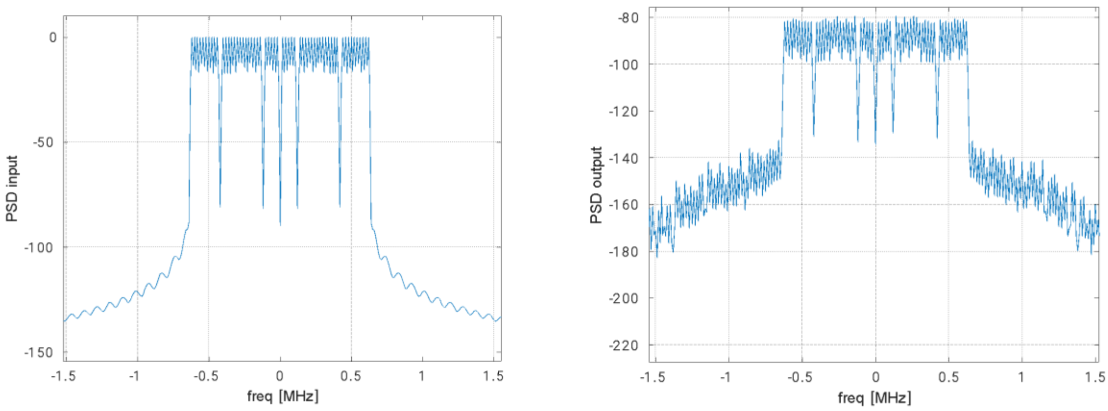

Figure 3.1.e – illustration of OOB emission caused by our lab's nonlinear PA

Volterra series as a PA model (MP)

Volterra series are used in order to model systems that are both nonlinear and have memory. As such, the Volterra series are often chosen to represent PAs response. The general form of the discrete Volterra series is given by [3]:

$$y(n) = \sum_{k=1}^{K} \sum_{m_1}^{M-1} \dots \sum_{m_k}^{M-1} h_k(m_1, \dots, m_k) \prod_{l=1}^{k} x(n - m_l)$$

where $y(n)$ is the output sample, $x(n)$ is the input sample, K is the order of non-linearity of the system, M is the order of memory of the system, and $h_k(m_1, \dots, m_k)$ is a coefficient with set values as a function of k and $m_1 \dots m_k$ [3].

The extensive number of coefficients corresponding to every combination of input sample and delayed input sample combinations within the bounds specified makes it possible to model large scale of nonlinear systems with memory effects. By raising the nonlinear and memory orders, K and M, the model can become even more accurate while dealing with systems which are extremely non-linear and have very long memory. However, the complexity in calculating the coefficients increases dramatically as either K or M is increased. As a result, many simplifications of the full Volterra model, with less coefficients, have been devised. One of these is the Memory Polynomial (MP) model, which is given by [3]:

$$y_{MP}(n) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} h_{km} x(n-m) |x(n-m)|^k$$

where $y_{MP}(n)$ is the output sample, $x(n)$ is the input sample, K is the order of non-linearity of the system, M is the order of memory of the system, and h_{km} is a coefficient with set values as a function of k and m [3].

This model can be represented efficiently in matrix form as [3]:

$$y = X\theta_{\mathrm{MP}}$$

$$\begin{bmatrix} y(n) \\ \vdots \\ y(n+N) \end{bmatrix} = \begin{bmatrix} x(n) & \cdots & x(n-m)|x(n-m)|^k \\ \vdots & \ddots & \vdots \\ x(n+N) & \cdots & x(n+N-m)|x(n+N-m)|^k \end{bmatrix} \begin{bmatrix} h_{00} \\ \vdots \\ h_{km} \end{bmatrix}$$

where $\boldsymbol{y}$ is the outputs array of the MP model, θ_{MP} is the coefficients array, and $\boldsymbol{X}$ is a matrix containing the signal, delayed values of the signal, and their powers, which required for calculating the output. This means that the matrix can be generated given only the input sample array and the order of non-linearity and memory order of the MP model to be used. The coefficient array θ_{MP} contains the unknown coefficients h_{km} that represents the power amplifier [3].

2.2 DPD

[3] After defining a realistic model of a PA, it is also desired to define a parametric model of an associated predistortion function that can be used to create an overall linearly behaving system. The model of the DPD should be an inverse of the model of the PA, hence the DPD will also need to account for both non-linearities and memory effects. That means that the DPD can also be successfully modeled using the MP model, given by [3]:

$$x = Y\theta_{PD}$$

where x is the DPD output signal array, Y is the delay matrix formed by the DPD input signal, and θ_{PD} is an array containing all the MP model coefficients of the DPD. Order of nonlinearity and order of memory for the DPD should be found as well [3].

2.3 MSE

<u>PA modeling</u>

We model the PA output signal as [3]:

$$y_{MP}(n) = \sum_{k=0}^{K-1}\sum_{m=0}^{M-1} h_{km} x(n-m)|x(n-m)|^k$$

This model can, too, be represented efficiently in matrix form as [3]:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta}_{\mathbf{MP}}$$

$$\begin{bmatrix} y(n) \\ \vdots \\ y(n+N) \end{bmatrix} = \begin{bmatrix} x(n) & \cdots & x(n-m)|x(n-m)|^k \\ \vdots & \ddots & \vdots \\ x(n+N) & \cdots & x(n+N-m)|x(n+N-m)|^k \end{bmatrix} \begin{bmatrix} h_{00} \\ \vdots \\ h_{km} \end{bmatrix}$$

where $\boldsymbol{y}$ is the outputs array of the MP model, θ_{MP} is the coefficients array , and $\boldsymbol{X}$ is an matrix containing the signal, delayed values of the signal, and their powers, which required for calculating the output [3].

In order to estimate this coefficients, one requires to have measurements of the output of the PA excited by a well-known input signal. The input signal should be chosen, such that it ranges over the whole bandwidth and input amplitudes that the modeled amplifier should handle. We put the measurements of the output signal that we got from our lab's nonlinear PA in a vector y. By using this vector, the coefficients vector θ_{MP} can be found using two different methods [3].

The first method is the well-known method of minimizing the least squares error between the calculated output and measured output. The coefficients vector is given by [3]:

$$\theta_{MP} = (X^H X)^{-1}\, Xy$$

Where $\boldsymbol{y}$ is the measured output signal, $\boldsymbol{X}$ is the matrix detailed above, which formed by the input signal and the superscript $()^H$ is the conjugate transpose operator.

The second method is Stochastic Gradient Descent (SGD). We used this algorithm in order to find the parameters that minimize the least squares error between the calculated output and measured output.

We set the step size μ using the following formula:

$$\mu = \alpha \cdot (X^H X + \epsilon)^{-1}$$

While ϵ was set to a very small number - 10^{-14}, and alpha was set to $5 \cdot 10^{-3}$. According to this definition, step size μ is a matrix. We did it because the gradient has radical differences between its components, and in order to handle this with SGD, the above definition for step size is required. The full algorithm is given as:

$$\theta^{(n+1)} = \theta^{(n)} - 2\mu X^H \left(X\theta^{(n)} - y \right)$$

In order to find the optimal coefficients for the MP model, it is essential to choose values of m and k which model PA optimally. In order to do that, one should find values of m and k that minimize the error between calculated output generated by coefficients calculated with those specific m and k, and measured output [3]. This result of this optimization process is described in figure-3.3.a. The error is minimized at $k = 9, m = 2$.

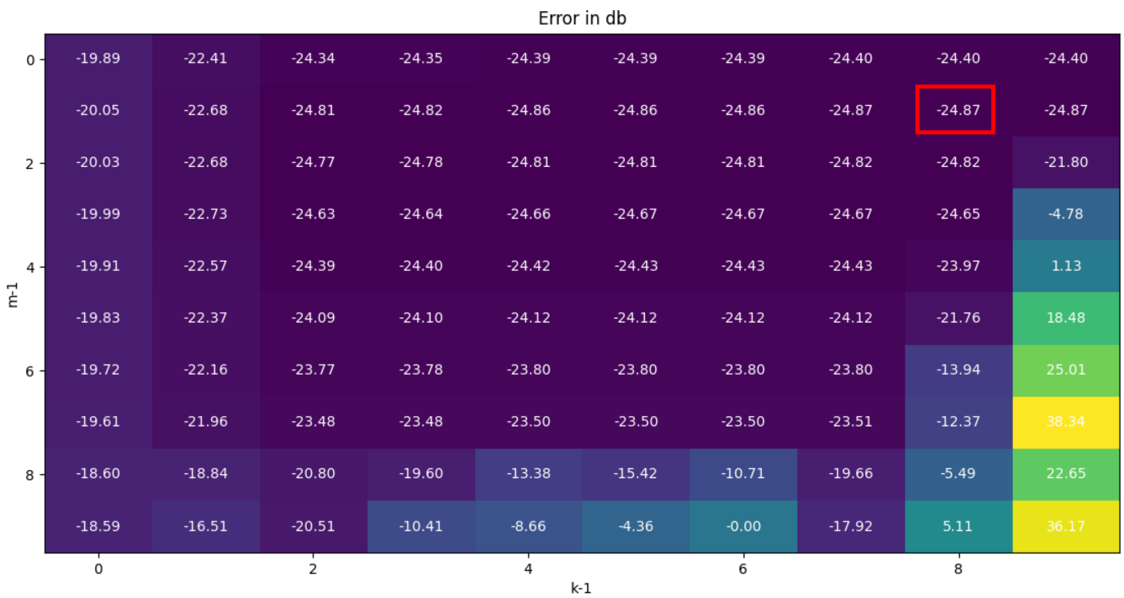

Figure 3.3.a - error between calculated and measured output in db as function of parameters m,k

After an adequate order of non-linearity and order of memory were selected, and if the range of input amplitudes and frequencies present in a general input signal fall within those represented by the calibration data input signal, then the realistic output of the HPA can be simulated (according to MP model) [3].

DPD

As detailed before, we modeled DPD output signal such that [3]:

$$x = Y\theta_{PD}$$

where x is the DPD output signal array, Y is the delay matrix formed by the DPD input signal, and θ_{PD} is an array containing all the MP model coefficients of the DPD.

In order to calculate the coefficient array Θ_{PD}, the same set of measurements is used, but it is scaled and used in reverse order. The input signal array x remains unscaled, but the measured output signal y is rescaled in a way that its maximum magnitude equals the maximum magnitude of the input array x [3]:

$$y_{scaled} = y \cdot \frac{\max(|X_{in}|)}{\max(|y|)}$$

When using the well-known method of minimizing the least squares error through the Moore-Penrose Pseudoinverse between the calculated DPD output and measured DPD output (which is the input array x). The coefficients vector is given by [3]:

$$\theta_{PD} = (Y^H Y)^{-1} Y^H x$$

where x is the unscaled input signal array of the calibration data, Y is the delay matrix formed by the rescaled calibration data output signal y, and the superscript $()^H$ is the conjugate transpose operator.

In order to find the optimal coefficients for the MP model, it is essential to choose values of m and k which model the DPD optimally. In order to do that, one should find values of m and k that minimize the error between calculated DPD output generated by coefficients calculated with those specific m and k, and the real input signal [3]. Similar optimization to the one on figure 3.3.a was done for the DPD, shown on figure-3.3.b. The error is minimized at k=9, m=2, similarly to the parameters of the PA model.

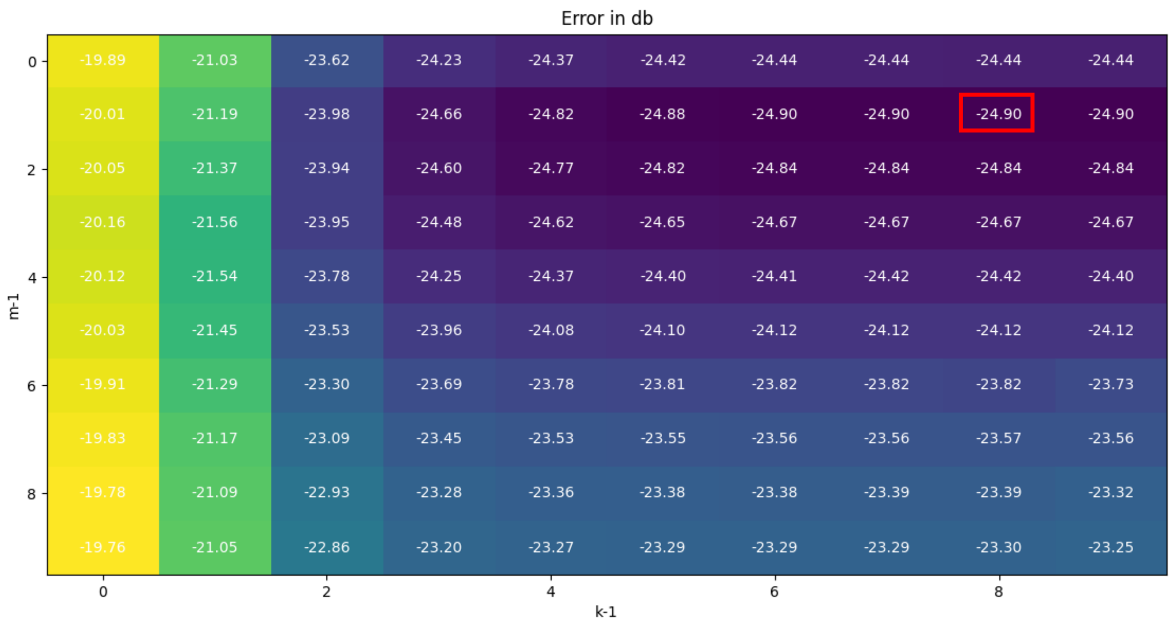

Figure 3.3.b - error between calculated and measured input in db as function of parameters m,k

3.4 NEURAL NETWORK

Neural Network explained

NNs usually consist of an input layer, hidden layers, and an output layer. The number of hidden layers is a hyperparameter which changes according to the specific problem, and many times is determined during training of the network or by using empirical methods. The output of each neuron is equal to the bias plus the sum of the products of the input signals and corresponding weights, which is expressed as [4]:

$$c_i^l = f\left(\sum_{j=1}^{p} w_{ij}^l x_j^{l-1}(n) + b_i^l\right)$$

Were l is the index of the layer, i is the index of the neuron in the layer, j is the index of the neuron in the previous layer, p is the number of neurons in the previous layer, w_{ij}^l is the weight connecting the i-th neuron of the l-th layer to the j-th neuron of the (l-1)-th layer, x_j^{l-1} is the input from the j-th neuron of the (l-1)-th layer to the i-th neuron of the l-th layer, b_i^l is the bias of the i-th neuron at the l-th layer and f is the activation function[4] . Commonly used activation functions are the log sigmoid (Logsig) and Relu.

After the cost function between the NN output of the first iteration and desired output is calculated, the values of the weights and biases are updated using a well-known type of the back-propagation algorithm that has been shown in literature to perform exceptionally well. The cost function is then calculated again using the updated weights and biases during the next iteration. Learning stops when the number of iterations reaches the maximum iterations allowed, or when the NN satisfies the desired performance in terms of MSE [4].

Modeling a PA and a DPD with a NN

Due to their strong adaptive nature and approximation capability, NNs are very attractive for the behavioral modeling of PAs. One optional architecture for such NN is a single-input single-output feedforward model, in which the network gets a complex signal as input and extracts complex PA output. However, this architecture requires the use of complex-valued weights and activation functions, which results in complicated calculations. Another proposed architecture is a polar feed forward NN, in which the network consist of two different NNs. The first NN extracts the amplitude response of the PA output, and the second extracts the phase response. However, the two NN branches in this design usually cannot converge at the same time, resulting in overtraining or undertraining [4].

In order to solve this, another architecture has been proposed, as shown in figure 3.4.a. This model uses the I and Q signals as inputs, and it resolves the simultaneous convergence issue mention above. However, this architecture does not take into account memory effects. These effects become dominant in wideband signals and need more consideration [4].

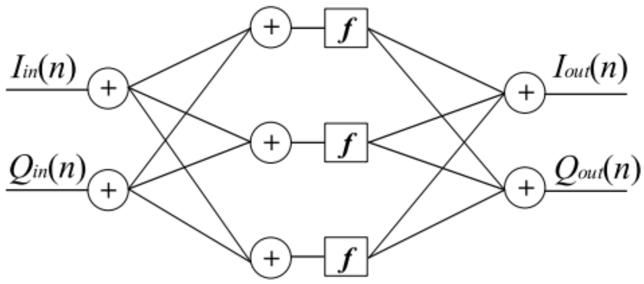

Figure 3.4.a [4] – NN architecture for modeling a PA

The proposed model in the article [4] is a combination of the above architectures, and it presented in figure 3.4.b. We used this model as a baseline for our proposed solution.

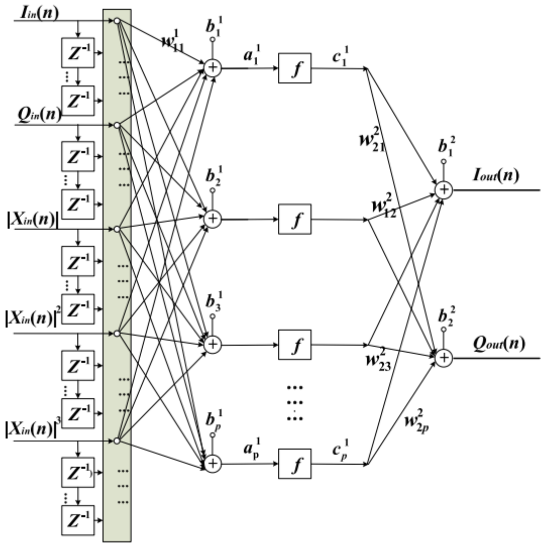

Figure 3.4.b [4] – proposed NN architecture for modeling a PA. We used this as a baseline for our proposed solution.

It considers input's in-phase (I) and quadrature (Q) components, the amplitudes of the input signal and nonlinear orders of it. In addition, this model includes past samples, and, therefore, quite suitable for the modeling of memory effects. The input feature vector for this model is [4]:

$$[I_{in}(n), I_{in}(n-1), \ldots, I_{in}(n-(M-1)), Q_{in}(n), Q_{in}(n-1), \ldots, Q_{in}(n-(M-1)),$$

$$|X_{in}(n)|, |X_{in}(n-1)|, \ldots, |X_{in}(n-(M-1))|, |X_{in}(n)|^2, |X_{in}(n-1)|^2, \ldots, |X_{in}(n-(M-1))|^2, \ldots,$$

$$\ldots |X_{in}(n)|^K, |X_{in}(n-1)|^K, \ldots, |X_{in}(n-(M-1))|^K]$$

And therefore, the input vector size is: $(2+K)*M$, and output vector size is 2 [4].

The memory order M and the order of nonlinearity K are determined by an optimization process and may vary depending on the PA.

The cost function is the MSE between the measured PA output and the NN output, and it defined as [4]:

$$E = \frac{1}{N}\sum_{n=1}^{N}(|y_{meas}(n) - y_{NN}(n)|)^2 = \frac{1}{N}\sum_{n=1}^{N}\left(\left(I_{meas}(n) - I_{NN}(n)\right)^2 + \left(Q_{meas}(n) - Q_{NN}(n)\right)^2\right)$$

As mentioned before, we found the ideal memory order ($M = 2$) and nonlinearity order ($K = 9$), therefore input vector size is 22, and output vector size is 2. In the training process, we have tested different architectures (while using the architecture in figure 3.4.b as a baseline) in order to find the optimal architecture which can model the PA and the DPD. We tried to change the number of the hidden layers, and the number of neurons at each layer. After tests and comparisons, the architecture that was chosen for the PA is a feedforward NN consist of three hidden layers, which contain 22,40,2 neurons, respectively. For the DPD, the architecture that was chosen is a feedforward NN consist of seven hidden layers, which contain 22-30-40-30-20-10-2 neurons. The chosen architectures are shown on figure 3.4.c-d respectively.

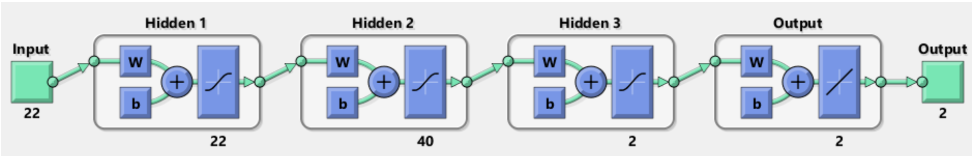

Figure 3.4.c – The chosen NN architecture we used to model PA

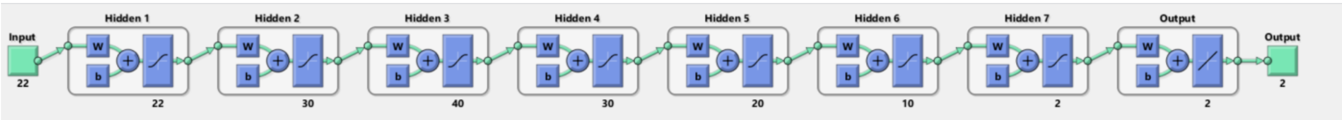

Figure 3.4.d – The chosen NN architecture we used to model DPD

3. METHODOLOGY

Data (lab & simulation)

In order to model a real PA, we used data recorded in our lab's PA. We took sampling rate of 10MHz, of about 60 tones, from a bandwidth of about 1MHz.

Road Map

ID	Action Item		Analysis tools	Literature review	Final report	Notes
1	Signal generation		Plot spectrum Plot time variation Calculate PAPR Power histogram		Plots and PAPR	Type: multi-tone
2	General PA Model		Pin Vs. Pout AM/AM AM/PM	Volterra sums AM/AM AM/PM Intermodulation	Volterra sums AM/AM AM/PM Intermodulation	Input to block: Parameter vector Time domain signal Output: Time domain signal
3	Parameter selection for the PA (LS)		MSE	GD and LS	GD and LS	Based on lab samples
4	Parameter selection for the PA (NN)		MSE/Loss function	NN in general	NN (General) Selected model: description and explanation Results	
5	DPD LS			The different methods to perform DPD	Solution description and explanation Results	
6	DPD NN				Selected model: description and explanation Results	

4. RESULTS

4.1 PA MODELLING

Classical

In order to test the parameter estimation process several sanity checks were made. The first is visualizing the AM/AM, AM/PM and OOB emission and to check if they are similar to the same graphs of the measured output as shown in figures 3.1.a-d. These graphs are shown in figures 5.1.a-d respectively.

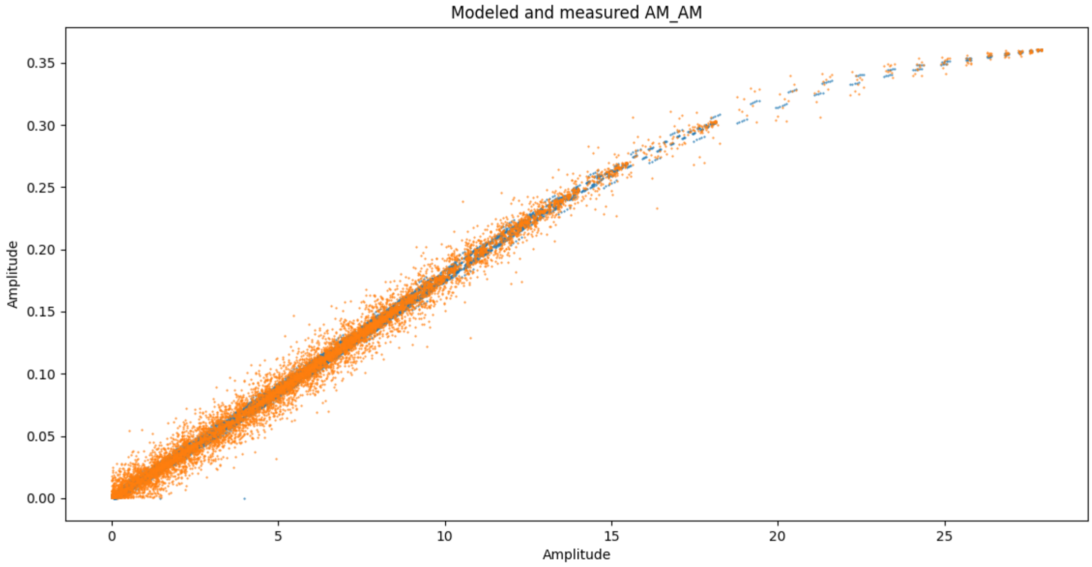

figure 5.1.a - illustration of AM/AM distortion of input signal, caused by classically modeled PA in blue, and real lab's PA in orange

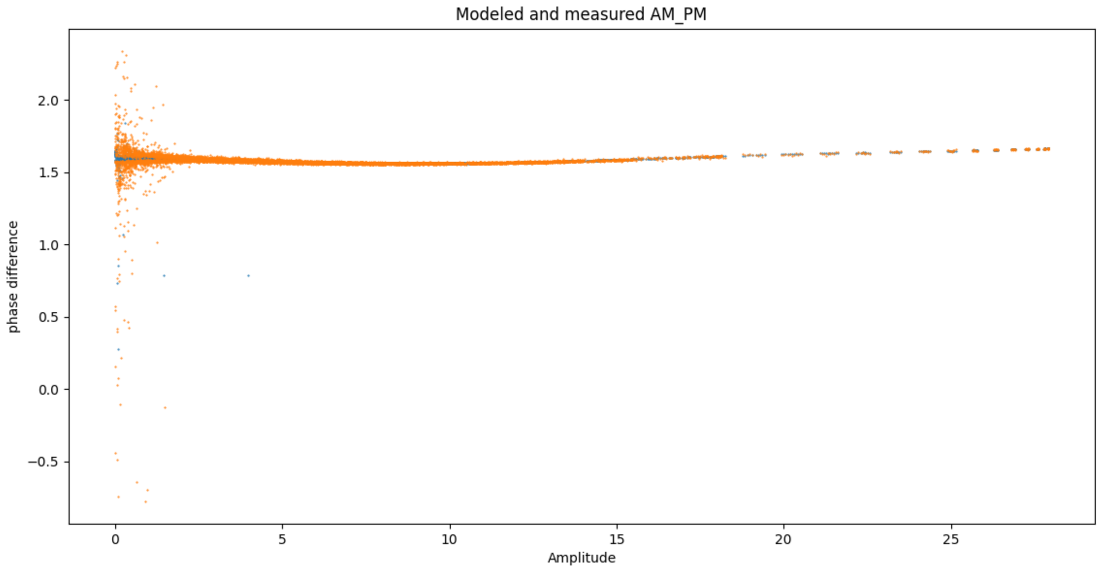

figure 5.1.b - illustration of AM/PM distortion of input signal, caused by classically modeled PA in blue, and real lab's PA in orange

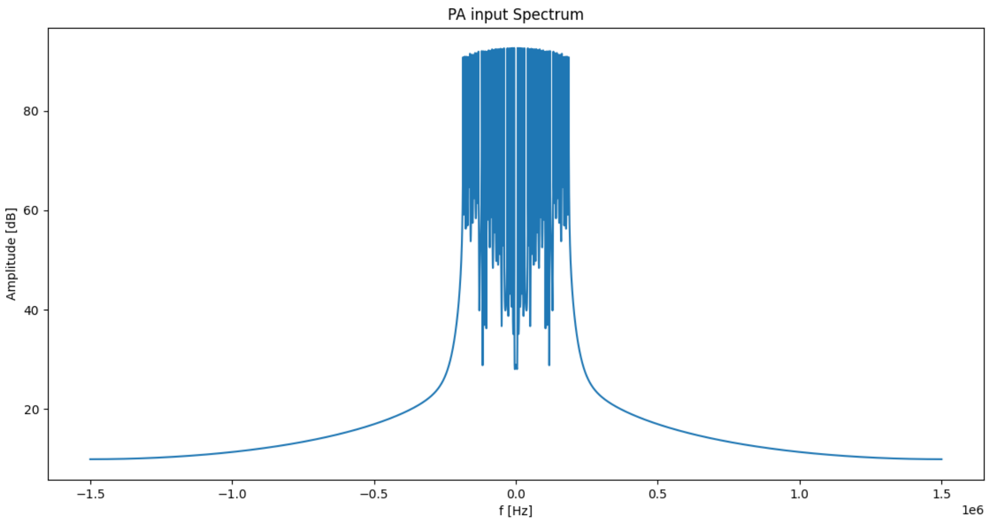

Figure 5.1.c – illustration of input signal spectrum

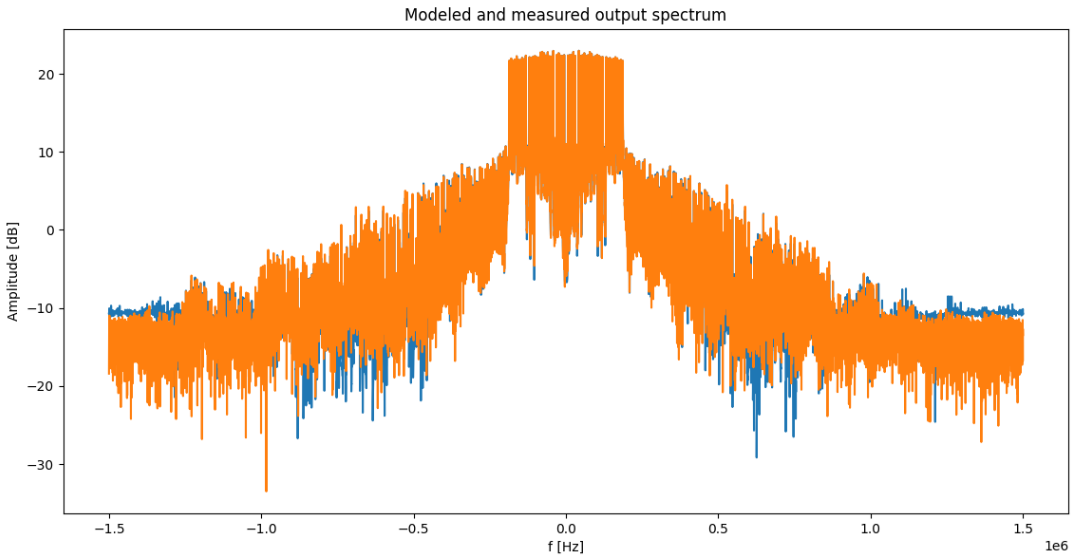

Figure 5.1.d – illustration of OOB emission caused by classically modeled PA in blue, and real lab's PA in orange

The results of the SGD process were satisfying as well, which are shown in figure 5.1.e.

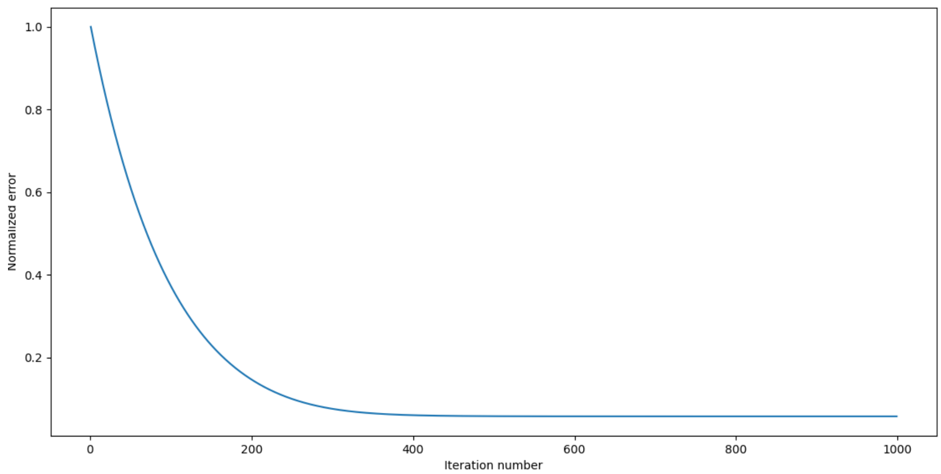

figure 5.1.e - Normalized error between SGD modeled PA output and real Lab's PA output

Although the above problem has an analytic solution, we also used SGD to model the PA. After we got the final parameter vector θ_{SGD}, we made a comparison to it with θ_{LS}, by calculating:

$$\frac{|\theta_{SGD} - \theta_{LS}|}{|\theta_{LS}|}$$

And the result was $0.00008 \approx -82dB$, which indicates a good convergence to solution.

NN

As described before, we tried different NN architectures for modeling the PA. One of the architectures we tried was a feedforward NN that consist of three hidden layers, at the sizes of $22,40,2$ neurons, respectively. AM/AM, AM/PM and OOB emission graphs of this NN output are shown in figures 5.1.f-h respectively.

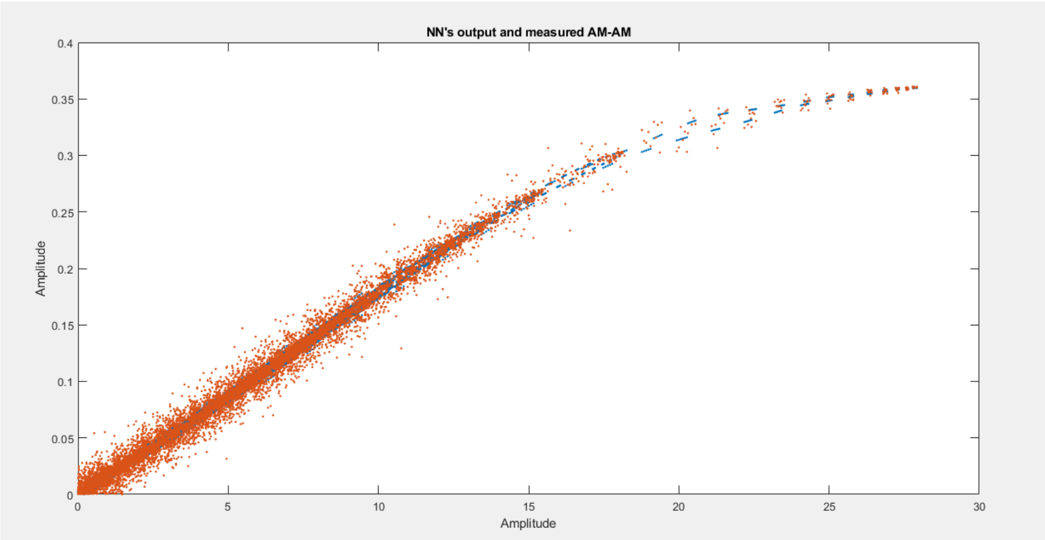

figure 5.1.f - illustration of AM/AM distortion of input signal, caused by feedforward (22-40-2) NN modeled PA in blue, and real lab's PA in orange

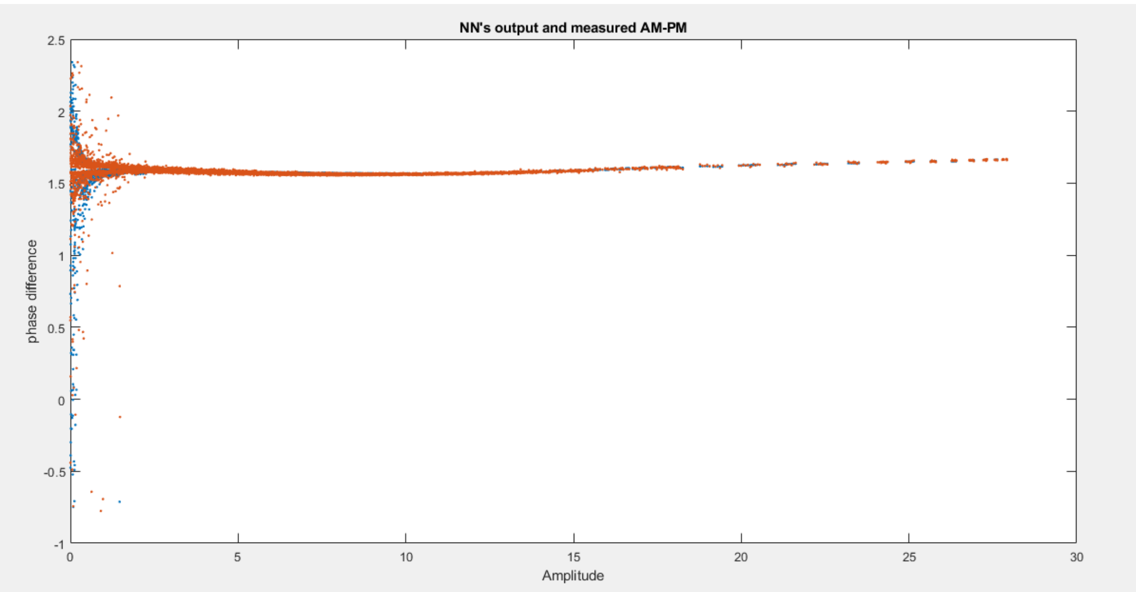

figure 5.1.g - illustration of AM/PM distortion of input signal, caused by feedforward (22-40-2) NN modeled PA in blue, and real lab's PA in orange

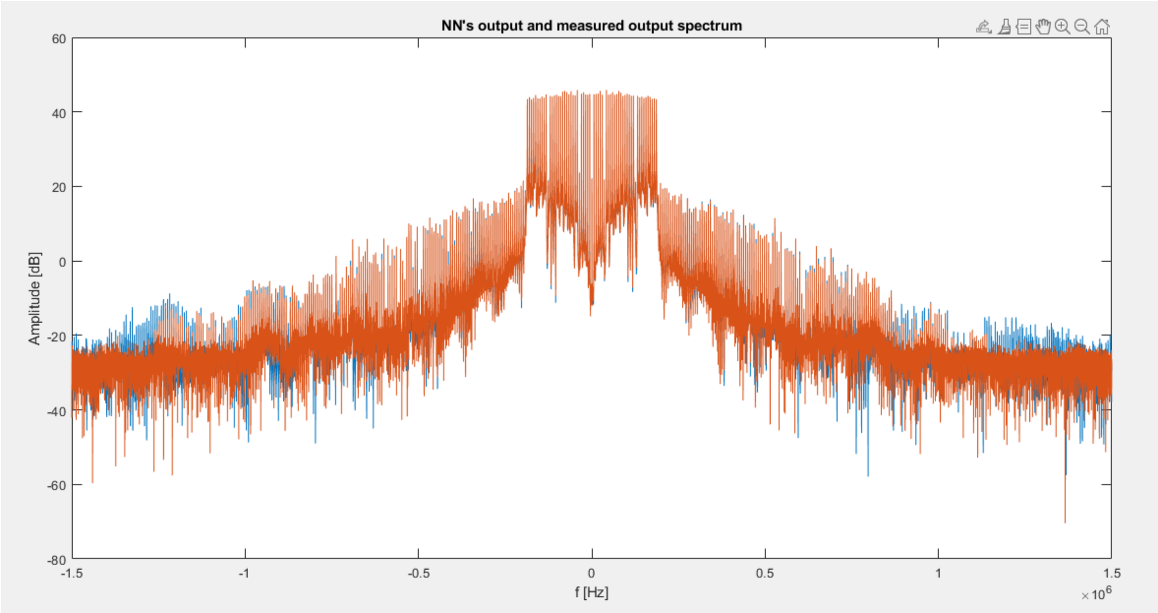

Figure 5.1.h – illustration of OOB emission caused by feedforward (22-40-2) NN modeled PA in blue, and real lab's PA in orange

4.2 DPD

Classical

In order to check the effectiveness of the proposed classical DPD, we will show AM/AM, AM/PM and OOB emission graphs of the output, with and without using DPD, to illustrate the ability of the classical DPD to overcome the PA nonlinearity. These are shown in figures 5.2.a-c respectively.

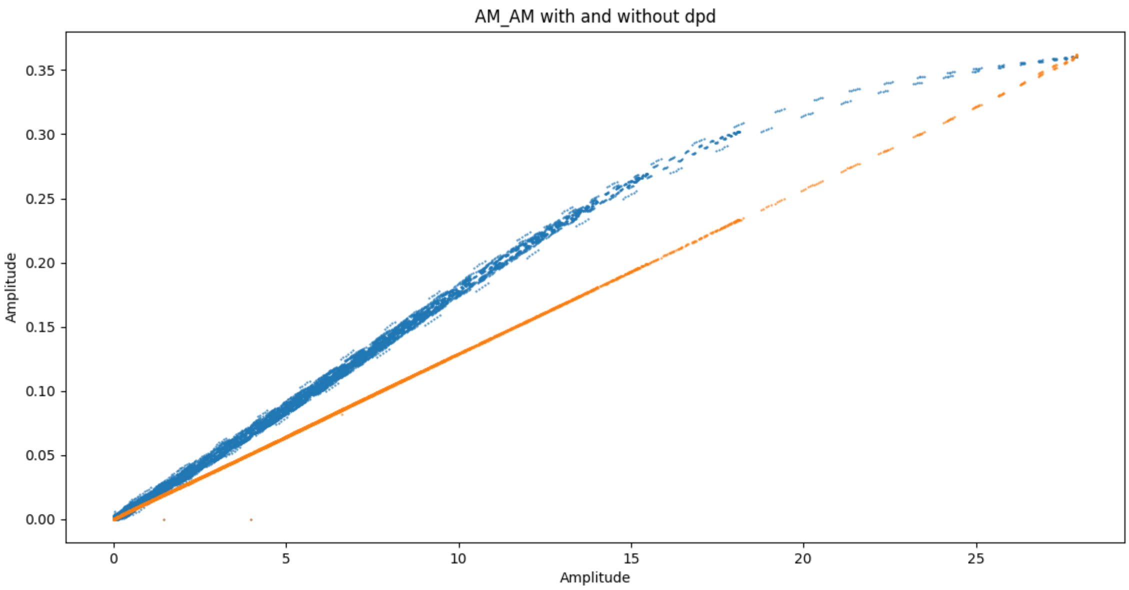

figure 5.2.a - illustration of AM/AM distortion of input signal, with (orange) and without (blue) classically modeled DPD

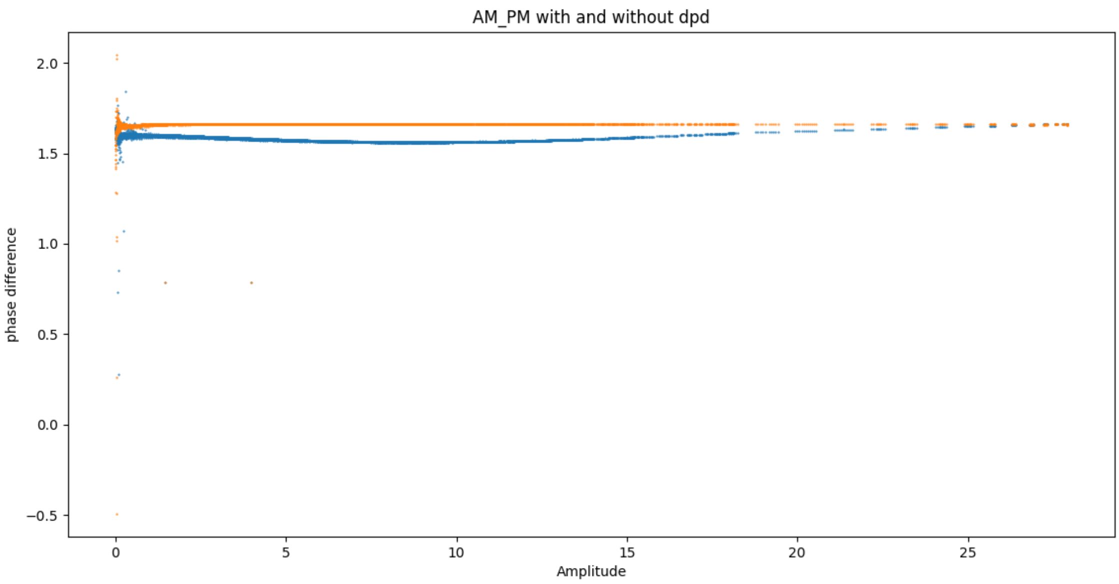

figure 5.2.b - illustration of AM/PM distortion of input signal, with (orange) and without (blue) classically modeled DPD

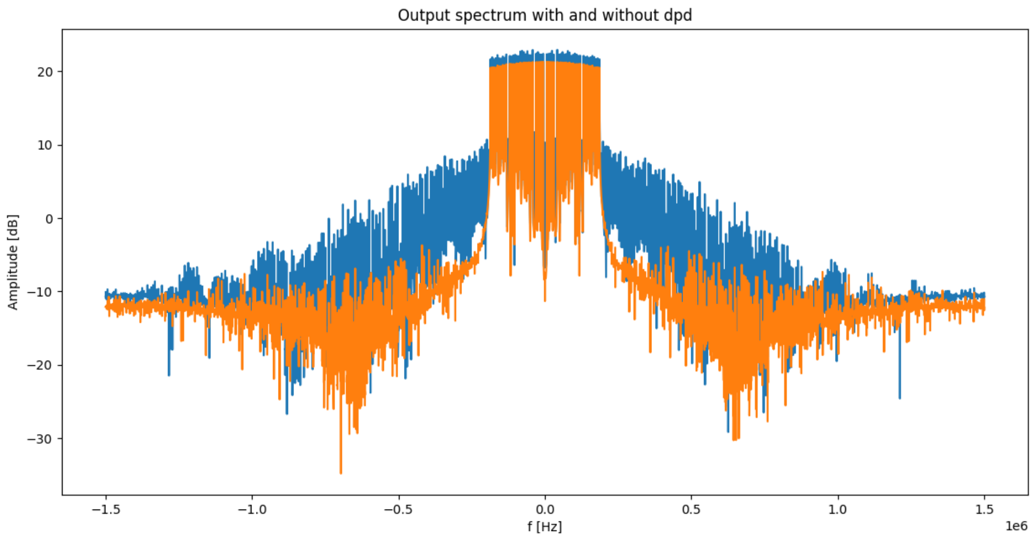

figure 5.2.c - illustration of OOB emission caused by modeled PA, with (orange) and without (blue) classically modeled DPD

<u>NN</u>

We will show AM/AM, AM/PM and OOB emission graphs of the output, with and without using DPD, to illustrate that the DPD somewhat overcomes the PA nonlinearity. These are shown in figures 5.2.d-f respectively.

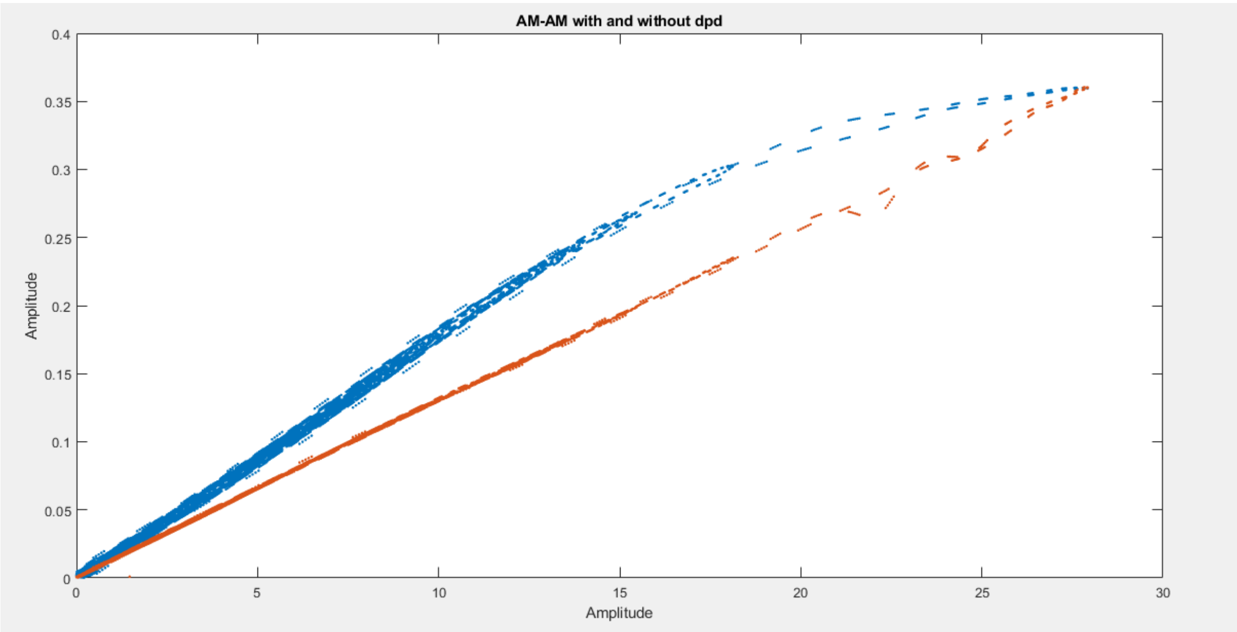

figure 5.2.d - illustration of AM/AM distortion of input signal, with (orange) and without (blue) feedforward (22-30-40-30-20-10-2) NN modelled DPD

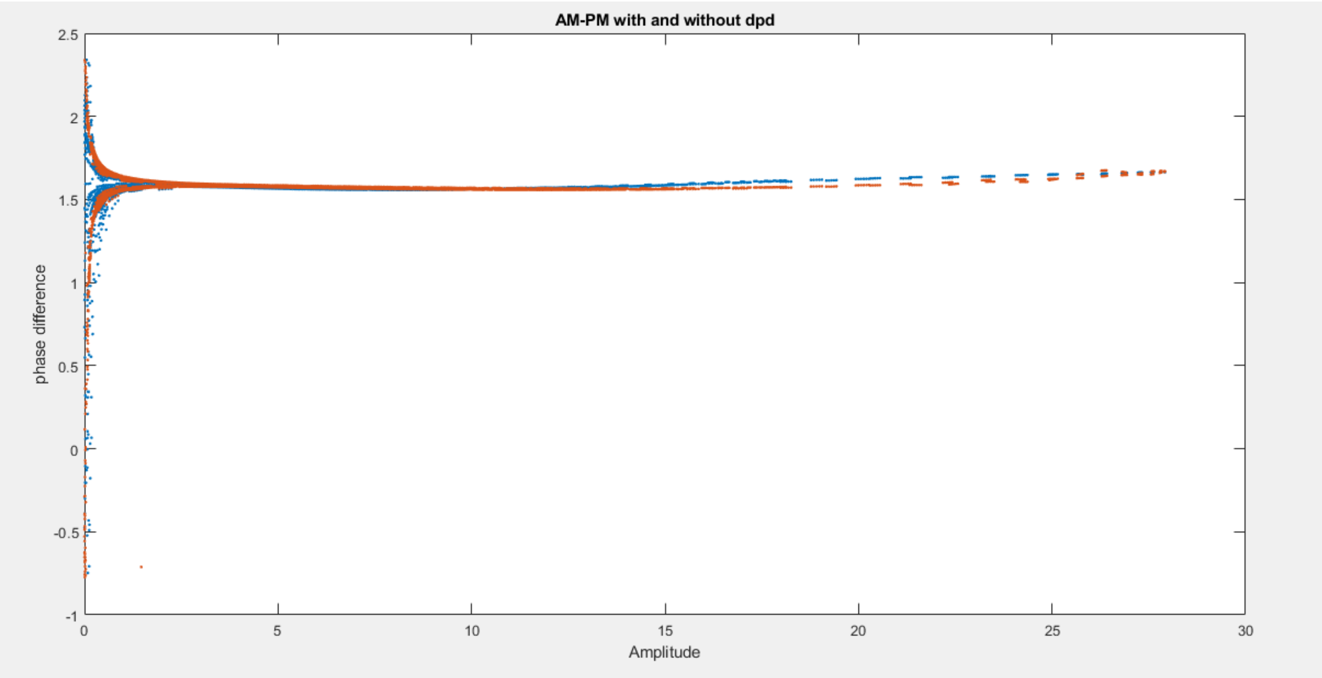

figure 5.2.e - illustration of AM/PM distortion of input signal, with (orange) and without (blue) feedforward (22-30-40-30-20-10-2) NN modeled DPD

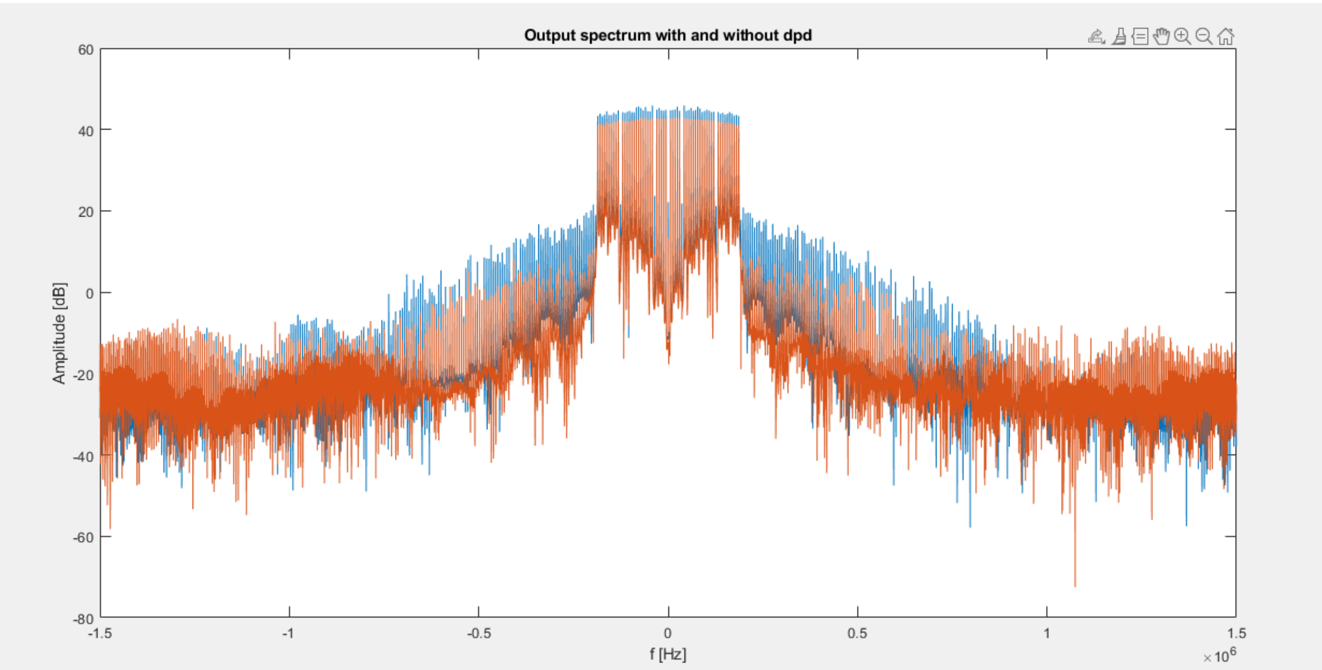

figure 5.2.f - illustration of OOB emission, with (orange) and without (blue) feedforward (22-30-40-30-20-10-2) NN modeled DPD

In order to justify why we chose this architecture, we will show examples of the AM/AM, AM/PM and OOB emission performance of different architectures that differ by hidden layers number and neurons number.

For example, the performance of a three hidden layers feedforward NN, with different number of neurons in the middle hidden layer, is shown in figure 5.2.g. It is shown that both AM/AM and OOB emission performance worse than proposed NN, as will be shown in the discussion chapter.

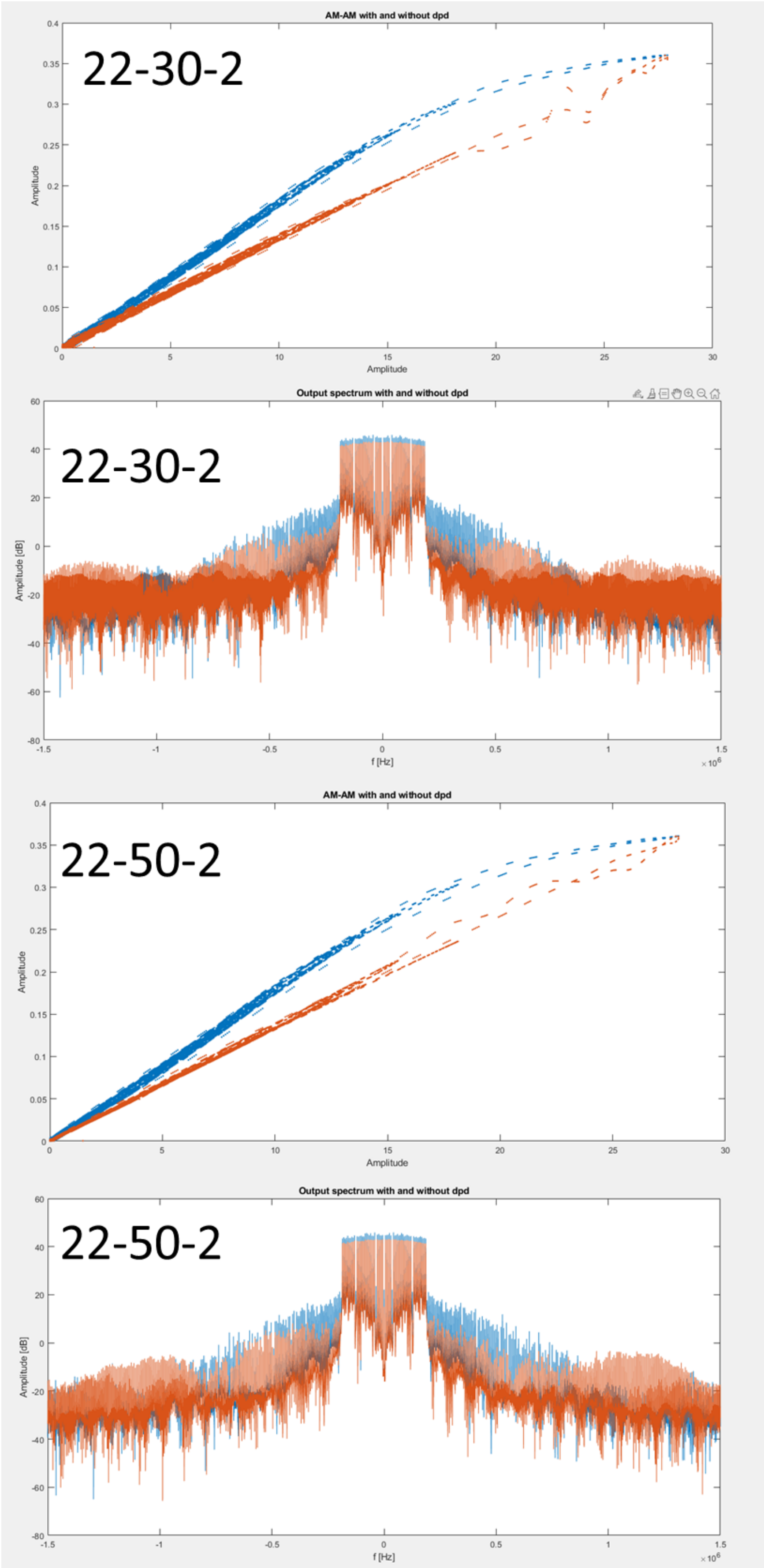
AM-AM with and without dpd
22-30-2
Amplitude
Amplitude
Output spectrum with and without dpd
22-30-2
Amplitude [dB]
f [Hz]
$\times 10^6$
AM-AM with and without dpd
22-50-2
Amplitude
Amplitude
Output spectrum with and without dpd
22-50-2
Amplitude [dB]
f [Hz]
$\times 10^6$

figure 5.2.g - illustration of AM/AM distortion and OOB emission, with (orange) and without (blue) feedforward (22-30-2) NN modeled PA and DPD, and feedforward (22-50-2) NN modeled PA and DPD, respectively

The performance of similar architectures as the proposed architecture, that were accepted by changing the number of neurons in hidden layers are shown in figure 5.2.h. It can be seen that AM/AM linearity damaged, as will be shown in the discussion chapter.

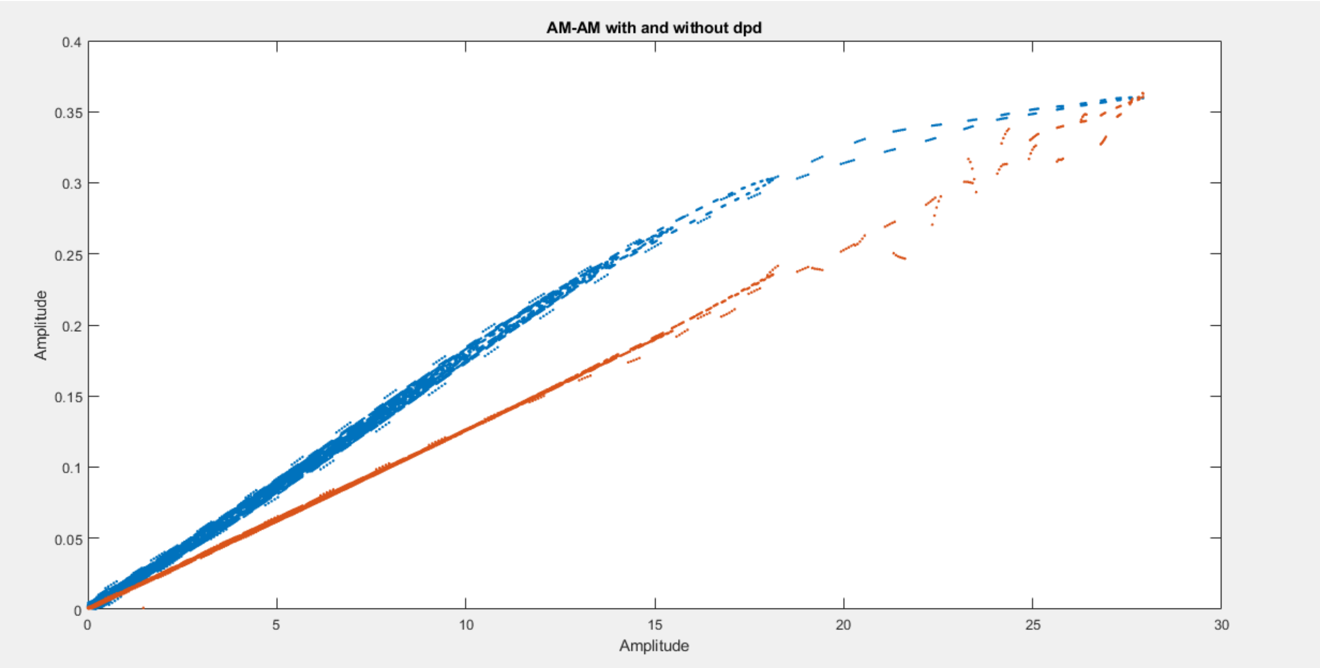

figure 5.2.g - illustration of AM/AM distortion, with (orange) and without (blue) feedforward (22-25-35-30-20-10-2) NN modeled DPD

5. DISCUSSION

In order to compare different NN architectures, together with classical method, and to justify why we chose the proposed model, different comparisons will be shown.

In order to test AM/AM performance, we calculated the normalized MMSE between desired output (Gain*input) and modeled output with DPD. Moreover, we checked what is the R^2 of linear regression of the AM/AM curvature. The results are shown in figure 6.1.a.

DPD Model	MMSE	R^2
22-30-2	-14.4 dB	0.9969
22-40-2	-16.4 dB	0.9985
22-50-2	-16.7 dB	0.9986
22-25-35-30-20-10-2	-17.7 dB	0.9993
22-30-40-30-20-10-2	**--17.4 dB**	**0.9996**
Classical method	**-22 dB**	**0.9998**

figure 6.1.a – AM/AM performance for different DPD models. The proposed model has the most linear AM/AM curvature, but still does not outperform classical method

It is clear that the proposed model does not gives the minimal MMSE, but we chose it due to its linearity performance. It is also does not outperform classical method.

In order to test OOB emission performance, we calculated the energy of the signal out-of-band and divided it by the energy of the signal in-band. We marked this quotient as η_{emission}.

$$\eta_{\text{emission}} = \frac{E_{OOB}}{E_{IB}}$$

We did this process for both output signals, with NN DPD and without NN DPD, and divided the result of the former by the latter. We marked this quotient as η_{DPD}, and it is given by:

$$\eta_{DPD} = \frac{\eta_{\text{emission}}^{with-DPD}}{\eta_{\text{emission}}^{without-DPD}}$$

The results are shown in figure 6.1.b.

DPD Model	η_{DPD}
22-30-2	-7.2 dB
22-40-2	-7.8 dB
22-50-2	-7 dB
22-25-35-30-20-10-2	-8.1 dB
22-30-40-30-20-10-2	**-8 dB**
Classical method	**-19.5 dB**

figure 6.1.b – OOB emission performance for different DPD models.

It is clear that the proposed model does not gives the minimal η_{DPD}, but we chose it due to it's linearity performance. It is also does not outperform classical method.

6. CONCLUSIONS

The methods described in this report can model a real power amplifier and a digital pre distorter that their total response together will be, approximately, linear. For this purpose, we used both classical and based on Neural Network methods. From our results, it can be shown that the proposed classical model outperforms every other model. Therefore, classical methods should be considered and tested seriously before using NN solutions. Moreover, we have shown that although MMSE is a convenient cost function for training, it doesn't necessarily indicates about the linearity of the model. Yet, we used data that contains relatively low content of high power, and was relatively narrowband. Therefore, in the future this methods should be examined in the communication lab with more suitable measurements. The proposed classical method, yielded MMSE of $-21.951dB$, and proposed NN method yielded MMSE of $-17.387\ dB$. Our code is supplied as a supplementary to this report.

7. REFERENCES

[1] Fadhel M.Ghannouchi, Oualid Hammi, Mohamad Helaoui, "Behavioral Modeling And Predistortion Of Wideband Wireless Transmitters", Chichester, United Kingdom, John Wiley & Sons Ltd, 2015, pp. 115-116.

[2] Lei Ding, "Digital Predistortion of Power Amplifiers for Wireless Applications", 2004, ProQuest Information and Learning Company, Ann Arbor, MI, pp. 1-4

[3] Z. Dunn, M. Yeary, C. Fulton, N. Goodman, “Memory Polynomial Model for Digital Predistortion of Broadband Solid-State Radar Amplifiers”, in 2015 IEEE Radar Conference (RadarCon), IEEE, 2015, pp. 1482-1486.

[4] Dongming Wang , Mohsin Aziz, Mohamed Helaoui, and Fadhel M. Ghannouchi, “Augmented Real-Valued Time-Delay Neural Network for Compensation of Distortions and Impairments in Wireless Transmitters”, IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 1, pp. 242-254, 2019.