EMITTER International Journal of Engineering Technology
Vol. 11, No. 1, June 2023, pp. 35~59
DOI : 10.24003/emitter.v11i1.791

Performance Analysis of MIMO-OFDM System Using Predistortion Neural Network with Convolutional Coding Addition to Reduce SDR-Based HPA Nonlinearity

Melki Mario Gulo[1], I Gede Puja Astawa[2], Amang Sudarsono[1]

[1]Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia
[2]Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia
Corresponding Author: melkimariogulo@pasca.student.pens.ac.id

Received February 12, 2023; Revised March 14, 2023; Accepted April 15, 2023

Abstract

In recent years, the development of communication technology has advanced at an accelerated rate. Communication technologies such as 4G, 5G, Wi-Fi 5 (802.11ac), and Wi-Fi 6 (802.11ax) are extensively used today due to their excellent system quality and extremely high data transfer rates. Some of these technologies incorporate MIMO-OFDM into their protocol. MIMO-OFDM is widely used in modern communication systems due to its benefits, which include high data rates, spectral efficiency, and fading resistance. Despite these benefits, MIMO-OFDM has disadvantages, with the use of a nonlinear HPA being one of them. Nonlinear HPA causes in-band and out-of-band distortions in MIMO-OFDM signals. Utilizing predistortion (PD) is one way of solving this issue. PD is a technique that uses the inverse distortion of the HPA to compensate for the nonlinear characteristics of the HPA. To enhance the quality of MIMO-OFDM systems that the use of HPA has degraded, the convolutional coding (CC) method can be combined with the help of PD. Convolutional coding is a type of channel coding that can be used for error detection and correction. This study will evaluate a combined technique of PD neural networks (PDNN) and CC on the MIMO-OFDM system using Software Defined Radio (SDR) devices. The evaluation of this system led to the use of a technique that combines PDNN and CC to improve SNR and minimise BER on MIMO-OFDM systems that HPA on SDR devices has degraded. In addition, at code rates 1/2, 2/3, and 3/4, using PDNN reduces the SNR value required to achieve BER equal to 0 by 12.037%, 37.8%, and 4.10% when compared to Digital Predistortion (DPD).

Keywords: MIMO-OFDM, SDR, Predistortion Neural Networks.

1. INTRODUCTION

In recent years, the advancement of communication technology has occurred at an accelerated rate. According to research conducted by the GSMA, 2.29 billion individuals used mobile internet technology in 2014. Due to this trend, mobile internet consumers will reach 4.32 billion, or 55 percent of the world's population, by 2021 [1]. The development of more advanced and resilient wireless communication technology influences the global increase in mobile internet users. In recent years, technologies like Wi-Fi 5 (802.11ac), 4G LTE, 5G, and Wi-Fi 6 (802.11ax) have been widely adopted and implemented in mobile devices. According to data from the Ericsson Mobility Report, 5G subscribers increased by 70 million from the previous quarter to the first quarter of 2021. In 2026, it is expected that Southeast Asia and Oceania will experience the greatest traffic growth [2]. The need for quick, dependable, and effective communication supports the growth of modern global communication.

As part of their communication protocols, 4G, 5G, Wi-Fi 5 (802.11ac), and Wi-Fi 6 (802.11ax) employ MIMO-OFDM technology [3][4][5][6]. MIMO-OFDM is utilised in modern communication technologies due to its speed and reliability in data transmission. In addition to its speed and reliability, MIMO-OFDM technology is utilised due to its enhanced spectral efficiency. The spectral efficiency of MIMO-OFDM is used in multi-user MIMO (MU-MIMO) technology on Wi-Fi 6, allowing users to interact at higher data rates. Resilience to channel conditions is also why MIMO-OFDM is widely used in today's communication technology developments.

The use of power amplifiers in communication devices such as smartphones, wireless routers, and base stations aims to improve communication systems' reliability further. A power amplifier will amplify the RF signal on the communication device before transmission. This amplification is intended to increase the SNR in the receiver to improve the quality of the received signal. Despite their beneficial uses, power amplifiers have nonlinear characteristics. The nonlinear part of the power amplifier will cause interference in the transmitted signal. Interference in the transmitted signal will cause signal degradation and distortion at the receiver. MIMO-OFDM systems use multiple data stream transmission schemes simultaneously that can cause high peak-to-average power ratio (PAPR) values in the transmitted signal. Especially in the MIMO-OFDM systems employed in the modern communication system, which have a large data rate because they use a high modulation rate and a larger number of subcarriers, the PAPR value of the signal will be higher [7]. The high PAPR value in the transmission signal can distort if the signal is processed on a nonlinear HPA. Therefore, we need a method that can compensate for the nonlinear characteristic of the HPA in MIMO-OFDM transmitters.

One method that can be used to compensate for the nonlinear characteristic of HPA is the predistortion (PD) method. PD is an inverse distortion method, which functions to linearise nonlinear signals due to the influence of the HPA. PD has the opposite characteristics of HPA. Several PD methods are widely used to compensate for the nonlinear nature of HPA, namely digital distortion (DPD) [8] and neural networks [9]. Neural networks began to be used as an alternative to PD to compensate for nonlinear HPA due to their ability to model PD in complex HPA characteristic model equations. In addition, neural networks are flexible because the same model can be used to model PDs with various HPA characteristics. Thus, using a neural network as a PD can provide more flexible and accurate properties compared to the PD method in general.

Channel coding is another method to improve signal quality and make communication more reliable. Channel coding is a mechanism used to protect data from damage during transmission. Many things, like the environment, fading, interference, and errors caused by nonlinear HPA conditions, can cause data to become corrupted while it is being sent. The communication system will have a minimal BER value by using channel coding. Some of the channel coding techniques used in MIMO-OFDM communication systems are turbo coding [10], LDPC [11], and convolutional coding [12]. Several things must be considered when selecting the type of channel coding for a communication system. Due to the cost of computing, complexity is one of the things that is considered when choosing the type of channel coding to use. Convolutional coding is channel coding with less complexity than LDPC and turbo coding. However, convolutional coding has higher error rates than LDPC and turbo coding. However, the use of convolutional coding in a communication system is enough to help improve the reliability of the communication system.

In this research, a combination of predistortion neural networks and convolutional coding techniques will be implemented on a MIMO-OFDM system based on Software Defined Radio (SDR). This study aims to improve the performance of MIMO-OFDM systems that are distorted due to the influence of non-linear HPA. The use of predistortion neural networks is expected to produce better system performance than the use of conventional digital predistortion (DPD) in compensating for HPA. Then, the addition of convolutional coding is expected to be able to reduce bit errors at the receiver caused by HPA and channels. Convolutional coding was also chosen because it has a low computational cost and will not reduce system performance, which will be evaluated on SDR devices with limited memory. The use of SDR devices in this study is intended for the implementation and evaluation of the proposed system under real-channel conditions.

2. RELATED WORKS

Research on the MIMO-OFDM communication system has been carried out in several previous studies. In some of these studies, there is research on the implementation of PD in MIMO-OFDM, as in the research conducted by Parag Aggarwal [13]. This study performed an end-to-end evaluation using a simulation on a nonlinear MIMO-OFDM system with PD. The joint nonlinear polynomial model is a characteristic modelling technique for the cascaded digital predistortion (DPD). In contrast, the typical high power amplifier (HPA) modelling uses the Saleh model in this research. Cascaded DPD can reduce the effects of signal distortion caused by nonlinear HPA. From this research, the symbol error rate (SER) graph on a system using DPD has the same quality as the graph on a linear system with AWGN channel conditions.

Besides that, there is also research conducted by Tingyu Huang [14]. This study used the Recursive Leas Square Digital Predistortion (RLS-DPD) algorithm to compensate for nonlinear HPA. From this study, it is known that the use of RLS-DPD can reduce the system bit error rate (BER) by 27.778%.

Meanwhile, research on the effect of the forward error correction code (FEC) on MIMO-OFDM was investigated by Arun Agarwal [15]. In this study, an evaluation of the MIMO-OFDM system was carried out with the channel coding types of Convolutional Coding (CC), Reed Solomon Coding (RSC) + CC, Low-Density Parity Check (LDPC) + CC, and Turbo Coding + CC. Four different modulation types were used for the evaluation: BPSK, QPSK, 16-QAM, and 64-QAM. In this research, simulations were done with three channels: AWGN, Rayleigh, and Rician. In addition, the MIMO-OFDM communication system uses a combination of 2x2, 2x4, and 4x4 antennas. The results obtained from this research show that turbo coding and convolutional coding produce a better BER value than other channel coding schemes.

Furthermore, there is also research on convolutional coding in SDR-based OFDM systems conducted by Ashish Thakur [16]. This study simulated the SDR system without hardware using GNU Radio software. This simulation uses AWGN, Rayleigh, and Rician channel types to evaluate the system's performance. This study found that convolutional coding resulted in a smaller BER value than the OFDM system without convolutional coding for all measurement scenarios.

Hendy Briantoro also conducted research on convolutional codes to reduce errors in SDR-based OFDM systems [17]. In this study, the modified convolutional code was used, which has better performance than the convolutional code. In this study, the use of a modified convolutional code was able to reduce the bit error at the receiver by up to 0%.

Research on computational complexity in FEC on SDR devices was researched by Zhenzhi Wu [18]. This research compares the computational complexity of several types of FEC, such as LDPC Layered Decoding, Turbo BCJR Decoding, and Convolutional Code Viterbi Decoding. Performance comparisons were made on several SDR devices. This research results show that convolutional coding with parameter CC (2, 1, 5) has the lowest computational complexity compared to other methods. At the same time, 802.16e LDPC has the highest computational complexity. This can be seen by comparing the number of decoded bits with the total number of operations.

Research on the implementation of MIMO-OFDM on SDR devices has been researched by Ridlo Qomarrullah [19]. In this study, USRP N210 and GNU Radio were used to evaluate the performance of the MIMO-OFDM system compared to SISO-OFDM. In this study, it was found that the use of MIMO-OFDM in real-world conditions with SDR was able to improve system quality. This can be seen by comparing the BER graphs between the MIMO-OFDM system and the SISO-OFDM system.

Krishna Kumar Sreedharan Pillai also studied the implementation of digital PD on SDR [20]. In this study, the author evaluated the effect of DPD on SDR devices using HPA hardware. From this study, it was found that the use of DPD can compensate for HPA hardware. This can be seen from the Error Vector Magnitude (EVM) value of the transmission signal with DPD, which is smaller than the EVM value of the transmission signal with HPA without DPD.

Then there is also research on the effect of PD on MIMO-OFDM conducted by M. Wisnu Gunawan [21]. In this research, we evaluate the impact of a PD neural network on an SDR-based MIMO-OFDM system. This research results in a PD neural network capable of producing a better signal constellation on MIMO-OFDM receivers in real channel conditions.

In this study, the performance of digital predistortion (DPD) and predistortion neural networks (PDNN) will be compared. Research on the comparison of DPD and PDNN has previously been carried out by Chance Tarver [22]. The result of this study is that the use of a neural networks-based predictor can outperform the use of memory polynomials to compensate for PA type CGH40006-TB in FPGA devices.

3. ORIGINALITY

The contribution made to this research is the creation and evaluation of PD neural networks combined with convolutional code can produce better system performance than the combination of conventional digital PD and convolutional code in MIMO-OFDM systems, which are degraded due to non-linear HPA. In addition, the difference between this study and previous research is that the evaluation of the proposed system is carried out using SDR devices with real-world channel conditions. Evaluation of the proposed system will be carried out by comparing BER graphs and SNR values in several measurement scenarios. The use of different coding rates in convolutional coding will also be used in this study. Some of the coding rates that will be used in this study are 1/2, 2/3, and 3/4. The purpose of using different coding rates in this study is to find the best combination between the PD technique with either a neural network or digital PD and the value of the coding rate under real-world conditions during measurement.

4. SYSTEM DESIGN

In building the system, several stages will be discussed. Some of these stages include the design of the MIMO-OFDM system, the use of convolutional coding on the system, the creation of PD neural networks, and the implementation of the system on SDR.

MIMO-OFDM

The creation of the MIMO-OFDM system in this study is based on the block diagrams in Figures 1 and 2.

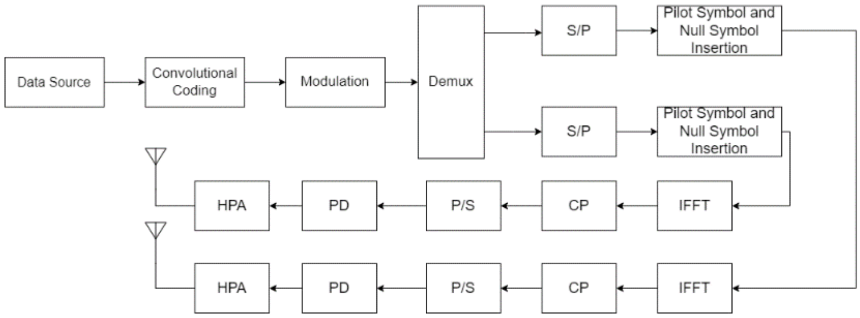

Figure 1. MIMO-OFDM Transmitter System Design.

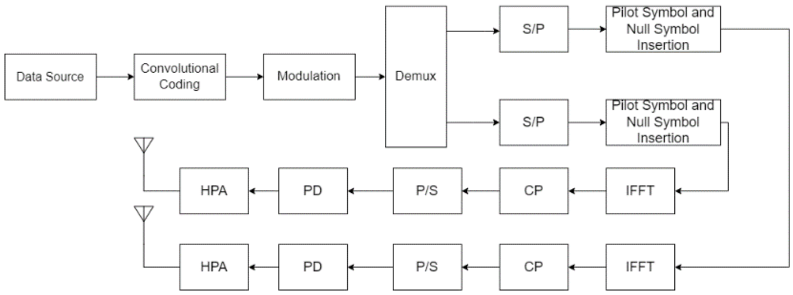

Figure 2. MIMO-OFDM Receiver System Design.

Figure 1 is an image of the MIMO-OFDM transmitter system used in this system. The data source will first be turned into a binary array at the MIMO-OFDM system transmitter. Then the binary collection resulting from the data source conversion will be coded using the convolutional coding method. Convolutional coding in this system aims to protect the data source from bit errors due to channel conditions during transmission. Further explanation regarding convolutional coding will be discussed in the next section. The binary array that has become a codeword will then be modulated into a complex signal. The modulation uses the Quadrature Amplitude Modulation (QAM) method at this stage. The modulation signal, commonly called the symbol, will then be demultiplexed into two signal streams. The demultiplexing process aims to divide the signal stream on each MIMO-OFDM transmitting device. In this process, the spatial-multiplexing method is used in demultiplexing. Spatial multiplexing [23] is a MIMO-OFDM scheme used to increase data rates in communication systems. With this method, the data streams on transmitter 1 and transmitter 2 are two different pieces of data. Thus, spatial multiplexing will increase the channel capacity according to the number of transmitters used. After the demultiplexing process, the serial-to-parallel conversion process is carried out. Next, a null symbol will be added as a guard interval and a pilot symbol for the channel estimation process. Furthermore, the symbol will be processed by the IFFT block diagram to convert the symbol into the time domain and OFDM modulation. The IFFT process can be formulated as equation 1.

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{\frac{j2\pi nk}{N}} \quad (1)$$

$x(n)$ is the n-th symbol in the OFDM signal in the time domain, while N is the number of subcarriers in the OFDM. $x(k)$ is the k-th symbol in the OFDM signal in the frequency domain, while $e^{\frac{j2\pi nk}{N}}$ is the twiddle factor used to orthogonalise each subcarrier in the OFDM signal. Then the OFDM signal will be added with a cyclic prefix (CP) to protect the signal against intercarrier interference (ICI). Furthermore, the OFDM signal on each stream will be converted into a serial form from parallel form. The PD will process signals already in serial form to compensate for the signal before being amplified by the HPA. A further discussion of PD will be explained in the next section. Finally, the PD-compensated signal will be processed by the HPA before the signal is transmitted.

Figure 2 is a MIMO-OFDM receiver system design. First, the signal received at each MIMO-OFDM receiver will be converted into parallel form. The cyclic prefix will be removed from each signal symbol in parallel form. The FFT process then converts the OFDM signal into the frequency domain. The FFT process can be formulated as equation 2.

$$x(k) = \sum_{n=0}^{N-1} x(n)e^{-\frac{j2\pi nk}{N}} \quad (2)$$

From equation 2, $x(k)$ is the k-th OFDM signal in the frequency domain, while $x(n)$ is the n-th OFDM signal in the time domain. At each receiver, the OFDM signal already in the frequency domain will be estimated for the channel. The channel estimation method used in this study is least squares (LS). The use of least squares in the MIMO-OFDM system has been used in several studies [24][25]. The least-square formula for channel estimation is written in equation 3.

$$\widehat{H} = (XX^H)^{-1}XY^H \quad (3)$$

From equation 3, $\widehat{H}$ is the estimated channel result, X is the $N_t \times N$ matrix from the pilot symbol in the transmitted signal, and Y is the $N_t \times L$ matrix from the pilot symbol in the received signal. N_t is the number of the subcarrier, while N and L is the pilot symbol in the transmitted and received signal.

Furthermore, the channel equalisation process uses the zero-forcing (ZF) method after obtaining the channel estimation results. The use of the ZF method for equalising MIMO-OFDM channels has been studied previously [26]. The ZF formula for channel equalisation is written in equation 4.

$$\hat{x} = \widehat{H}^{-1}y \quad (4)$$

From equation 4, $\hat{x}$ is the equalised signal, $\widehat{H}$ is the estimated channel, and y is the received signal. The pilot symbol and null symbol will be removed from the equalised signal. Furthermore, the signal will be converted into serial form to enter the multiplexing process. In the multiplexing process, the received signals from the two streams are combined into one stream. After demultiplexing, the demodulation process will be carried out using QAM. The demodulated data bits will then be processed in the convolutional coding decoder using the Viterbi algorithm. Further explanation about Viterbi will be provided in the next section. The resulting Viterbi bits will then be returned to their original data form.

Convolutional Coding

Convolutional coding (CC) is used in this paper to improve the signal quality of the system. This method works by adding redundant bits to the data bits. The use of redundant bits in CC is used to detect and correct bit errors in the decoder. CC works by using the concept of a linear feedback shift register (LSFR) in the encoding and decoding processes.

In this research, 3 types of CC will be used with different code rates, namely code rates 1/2, 2/3, and 3/4. Different code rate values aim to get the best rate when measuring the system. Of the several code rates, a code rate of 1/2 is the one that will produce the most redundant bits among the other code rates. This is because, for each input bit, two bits will be generated, one of which is a redundant bit. Meanwhile, a code rate of 3/4 will produce the fewest redundant bits among the others.

In this paper, each code rate was constructed using the constraint length parameter K = 3. This is intended so that the computational process on SDR is manageable. A larger constraint length will make the computation heavier because the number of shift registers used in coding operations will increase. Thus, to not burden SDR devices with limited memory, K = 3 is recommended for use.

Furthermore, each code rate used has its generator value. This can be seen from the generator representation for each code rate (1/2, 2/3, and 3/4) using the polynomial equation in Table 1.

Table 1. Generator Matrix

Code Rate	Generator
1/2	$G(x) = \begin{bmatrix} 1+x^2 & 1+x+x^2 \end{bmatrix}$
2/3	$G(x) = \begin{bmatrix} 1+x+x^2 & x^2 & 1 \\ x & 1+x^2 & 1+x+x^2 \end{bmatrix}$
3/4	$G(x) = \begin{bmatrix} 1+x & x^2 & 0 & 1+x+x^2 \\ x+x^2 & 1 & x^2 & 1+x \\ x & x+x^2 & 1+x+x^2 & 1 \end{bmatrix}$

The bit decoding process is carried out on the receiver side using the Viterbi method. Viterbi has functions to detect and correct bit errors in MIMO-

OFDM receivers. Viterbi uses the most likely sequence of the hidden state scheme in the Hidden Markov Model (HMM) in performing bit correction and detection. In its implementation, Viterbi will use a trellis diagram to represent the output of each bit transition in the convolutional code generator. In the proposed system, Viterbi is used in hard decision mode. In hard decision mode, the received bits will be compared with the output bits for each possibility in the trellis diagram. After reaching this, a comparison is made between each path on the trellis diagram. The path with the smallest error value will be selected as the decoded output bit. The use of Viterbi hard decision in this study is used because the method has a smaller computational cost than the Viterbi soft decision.

Predistortion Neural Network

PD is a method used to compensate for the nonlinear characteristic of HPA. The nonlinear characteristic of the HPA can cause in-band and out-of-band signal distortion in the transmitted signal. Moreover, the MIMO-OFDM system has a high PAPR value. A high PAPR value can cause distortion when the signal is processed on the HPA. This is because a signal with a high PAPR will most likely be input to the nonlinear region of the HPA. Therefore, PD is needed to reduce the distortion caused by the event. An illustration of how PD compensates for HPA can be seen in Figure 3.

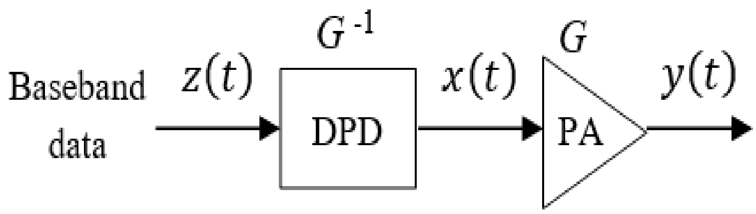

Figure 3. Block Diagram of PD and HPA.

From figure 3, let $x(t)$, $y(t)$ and $z(t)$ be the HPA input signal, the HPA output signal, and the PD input signal. While G is a transfer function of HPA and G^{-1} is a transfer function from PD. Then the output signal equation from the HPA can be represented in equation 5.

$$y(t) = G\left(G^{-1}\big(z(t)\big)\right) = G \circ G^{-1}\big(z(t)\big) = L\,(z(t)) \tag{5}$$

L is the composite function from $G \circ G^{-1}$. Then the HPA linearisation equation can be written like equation 6.

$$y(t) = L\,\big(z(t)\big) = g\,.\,z(t) \tag{6}$$

Where $g > 1$ is the gain value of HPA.

This research will use HPA modelling of the saleh type to model the AM/AM characteristics of the HPA [27]. The saleh model equation for modelling the HPA can be found in equations 7 and 8.

$$y(t) = G(x(t)) \tag{7}$$

$$G\big(x(t)\big) = \frac{\alpha . x(t)}{1+\beta . x^2(t)} \tag{8}$$

where α is the gain value of the HPA, while β is the coefficient of compression region. In the process of linearising the Saleh model in equation 7, a digital PD is used, which has the inverse of this model. In making the DPD based on equation 4, the PD function equation for the input signal can be written in equation 9.

$$x(t) = G^{-1}(z(t)) \tag{9}$$

where G^{-1} is the inverse function from G can be written in equation 10.

$$G^{-1}\big(z(t)\big) = \frac{\alpha - \sqrt{\alpha^2 - 4\ \beta\ z^2(t)}}{2\ \beta\ z(t)} \tag{10}$$

In addition to using digital predistortion to compensate for the HPA. In this research, the creation of PD neural networks will also be carried out. Neural networks can be used to create PD because they can model the input and output relationships of the HPA after the training process. Neural networks are used because they have high flexibility; this can happen because a neural network model can be used for various HPA models.

In making PD neural networks. The first step is to create a dataset. The dataset used is the HPA input and output signals from equation 7. the $y(t)$ signal will be used for input training. At the same time, the $x(t)$ signal will be used as the training output. Note that the signal $x(t)$ is a linear signal without PD.

After the dataset is prepared, the next step is to create a neural network model for PD. The type of neural network used in PD is an artificial neural network (ANN). The ANN model used consists of an input layer, hidden layers, and an output layer. In the input layer, two dimensions represent two features of the input signal. These features are the real and imaginary values of the input signal. The input layer in this model can be written in equation 11.

$$Y_1(n) = f_1\left(\omega_1 \begin{bmatrix} Re(X) \\ Im(X) \end{bmatrix} + b_1\right) \tag{11}$$

X is the signal input, ω_1 is the weight from the input layer, b_1 is the bias from the input layer, and f_1 is the activation function from the input layer. The Rectified Linear Unit (ReLu) type activation function is used in the input layer, which can be seen in equation 12.

$$f_1 = \max\,(0, x) \tag{12}$$

Furthermore, the hidden layer equation in this model is written in equation 13.

$$Y_i(n) = f_1(\omega_i \, BN(h_{i-1}) + b_i) \tag{13}$$

From equation 13, i is the hidden layer sequence where $i \in \mathbb{N}: 1 < i < T$, with T, as the output layer sequence number. From equation 13, BN is the batch normalisation function. Batch normalisation normalises the previous output layer before entering the activation function. With batch normalisation, the model is hoped to be faster in the training process and more stable. In addition, using batch normalisation further increases the generalisation of the neural network model and avoids overfitting [28]. Next, the mathematical equation of the ANN output layer model is written in equation 14.

$$Y_T(n) = f_2(\omega_T \, BN(h_{T-1}) + b_T) \tag{14}$$

where f_2 is the linear activation function. The output of the output layer of the ANN model is two, where the two values represent the real and imaginary values of the complex output signal. The implementation is carried out in the program from the several equations in each layer, and a summary of the ANN model shown in Table 2 and the illustration of this model shown in the figure 4.

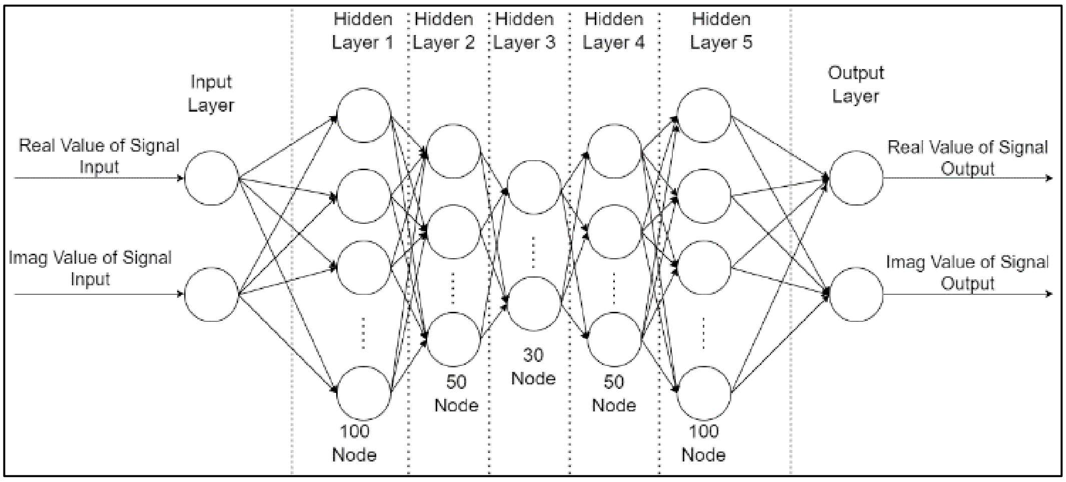

Figure 4. Neural Network Model Architecture Illustrations.

Table 2. Summary Model Neural Network.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 100)               300

 batch_normalization (BatchN  (None, 100)              400
 ormalization)

 activation (Activation)     (None, 100)               0

 dense_1 (Dense)             (None, 50)                5050

 batch_normalization_1 (Batc  (None, 50)               200
 hNormalization)

 activation_1 (Activation)   (None, 50)                0

 dense_2 (Dense)             (None, 30)                1530

 batch_normalization_2 (Batc  (None, 30)               120
 hNormalization)

 activation_2 (Activation)   (None, 30)                0

 dense_3 (Dense)             (None, 50)                1550

 batch_normalization_3 (Batc  (None, 50)               200
 hNormalization)

 activation_3 (Activation)   (None, 50)                0

 dense_4 (Dense)             (None, 100)               5100

 activation_4 (Activation)   (None, 100)               0

 dense_5 (Dense)             (None, 2)                 202

=================================================================
Total params: 14,652
Trainable params: 14,192
Non-trainable params: 460
```

Furthermore, the model that has been designed will be trained to make a PD neural network. Training parameters are used in this training, as shown in Table 3.

Table 3. Neural Networks Training Parameters

Parameter	**Value**
Loss Function	MSE
Optimizer	Adam
Epochs	50
Batch Size	100

System Implementation on SDR

This section will discuss the implementation of the proposed system on SDR devices and measurement setups to evaluate the proposed approach. An SDR device of the NI USRP-2920 type was used to implement the system. The

NI USRP-2920 is a type of SDR device manufactured by National Instruments. This device is often used in several studies regarding wireless communication because of its flexibility and ease of customising programs [29][30].

In the system implementation setup, four NI USRP-2920 devices will be used. By division, two devices are used as transmitters, and two devices are used as receivers. Device setup for system implementation can be seen in Figure 5.

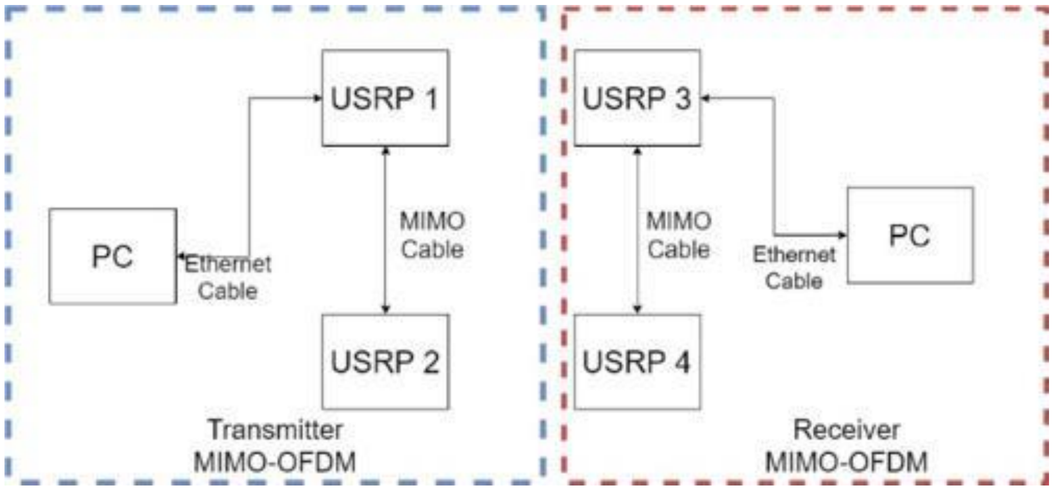

Figure 5. Implementation System Setup.

In Figure 5, it can be seen that the system transmitter and receiver use two NI USRP-2920, which work synchronously. A MIMO cable gets the NI USRP-2920 devices on the transmitter and receiver to work together. The MIMO cable has functions to synchronise the clock and frequency for each USRP. Then, an Ethernet cable connects a PC to the USRP device on the transmitter and receiver. The PC connected to the USRP device will use to run the proposed system transmitter and receiver program.

In implementing this system, programs are created using the LabVIEW programming language and Python. LabVIEW is used because it has NI USRP driver support, which can run programs with NI USRP-2920 devices, while Python is used for training and running the PD Neural Network program. Furthermore, an IP address-based network connects the LabVIEW program and the NI USRP-2920. The LabVIEW program used for system implementation can be seen in Figures 6 and 7.

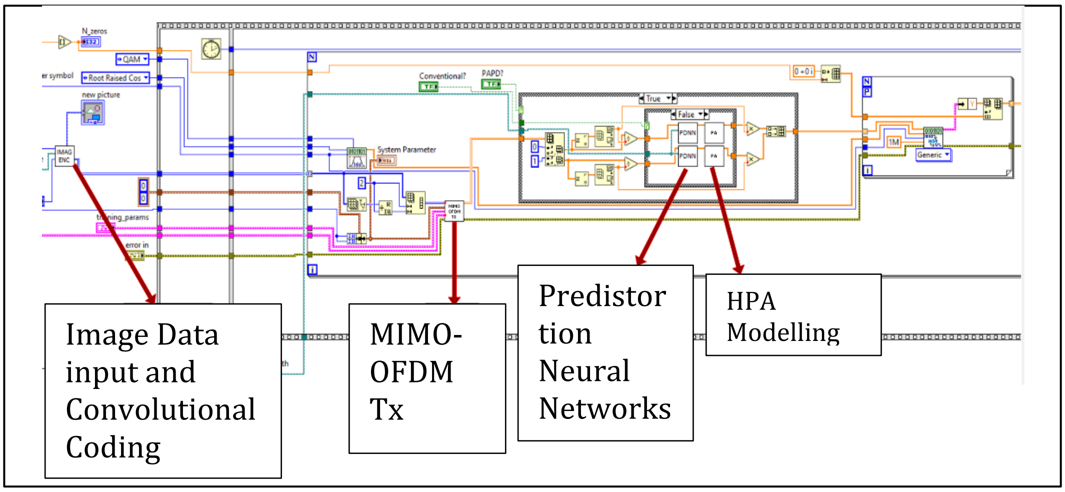

Figure 6. LabVIEW program for transmitter system.

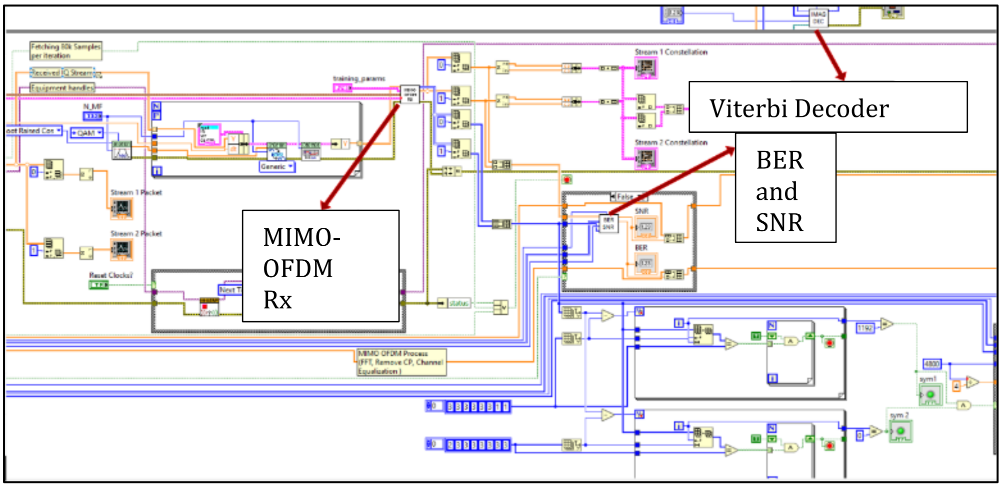

Figure 7. LabVIEW program for the receiver system.

In Figure 6, the HPA modelling used in this system is the godly HPA modelling with parameters α = 2.1587 and β = 1.517 [31]. Meanwhile, the PD neural network program using Python in LabVIEW will be implemented using the Python node module. The Python node implementation for running the PD neural networks program can be seen in Figure 8.

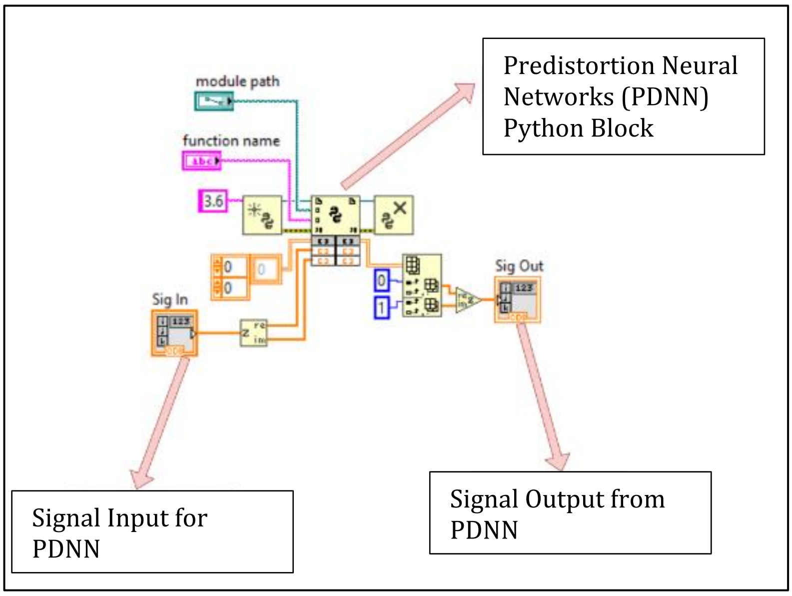

Figure 8. LabVIEW program for PD Neural Networks.

Furthermore, from the system implementation setup in Figure 5, it is realised into a data measurement setup, as shown in Figure 9.

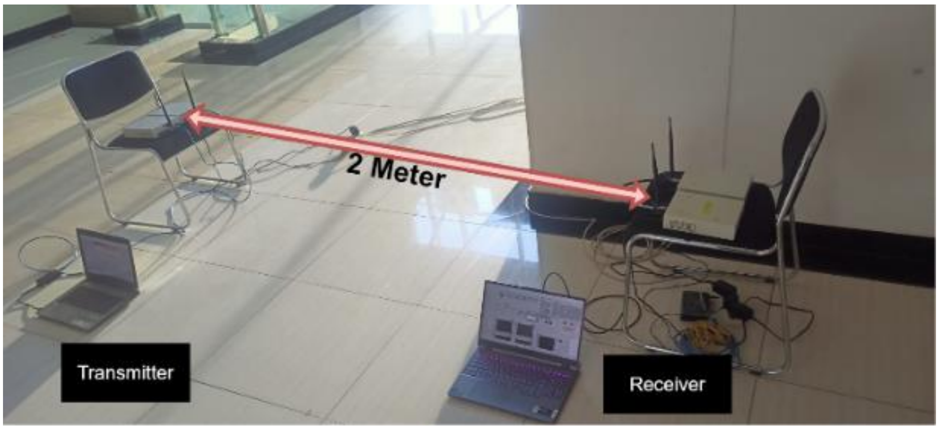

Figure 9. The realisation of the system measurement setup.

As shown in Figure 9, the proposed system measurement is carried out on the 4th floor of the Pascasarjana PENS Building. The distance between this measurement's transmitter and receiver systems is two meters under line-of-sight (LOS) conditions. The specifications of devices used in this measurement process can be seen in Table 4. Meanwhile, the implementation parameters used can be seen in Table 5.

Table 4. Device Specifications

No.	Device Name	Specification
1	NI-USRP 2920	Frequency Range : 50MHz – 2.2 GHz Connector: Ethernet
2	Antenna VERT 900	Frequency Range: 824 MHz – 960 Mhz, 1710 MHz – 1990 MHz
3	Receiver PC	Operating System: Windows 11 RAM: 16GB, Processor: Intel Core i7 12700H 4.7 GHz
4	Transmitter PC	Operating System: Windows 10 RAM: 8GB, Processor: Intel Core i5 8th Gen 2.5 GHz

Table 5. Implementation System Parameters.

Session	Parameter	Value
Transmitter	Frequency	915MHz
	N-IFFT	128
	Data symbol	96
	Pilot symbol	9
	Null symbol	23
	CP	32
	IQ Rate	1M
	Antenna	2
	HPA	Saleh [27]
	PD	DPD, Neural Networks
	Channel Coding	Convolutional
	Rate	1/2, 2/3, 2/4
	Modulation	4-QAM
Receiver	Channel Equalization	*Zero-Forcing*
	Channel Estimation	*Least Square*
	Frame Sync	ZC
	Antenna	2

5. EXPERIMENT AND ANALYSIS

In this session, the experiment results and discussion will be presented. The experiment is done using the setup that was discussed in the previous session. In this experiment, several data measurement scenarios were carried out, as shown in Table 6.

Table 6. Measurement Scenario

No	HPA/PD Scenario	CC Code Rate
1		1/2
2	MIMO-OFDM + HPA	2/3
3		3/4
4		1/2
5	MIMO-OFDM + DPD + HPA	2/3
6		3/4
7		1/2
8	MIMO-OFDM + PDNN + HPA	2/3
9		3/4

From Table 6, there are nine scenarios. This scenario was made to evaluate the proposed system, especially to know the performance of the system using the PD neural network against DPD and HPA.

The proposed receiver system will present the constellation signal data from all these scenarios. The results of these data can be seen in Figures 10 to 18.

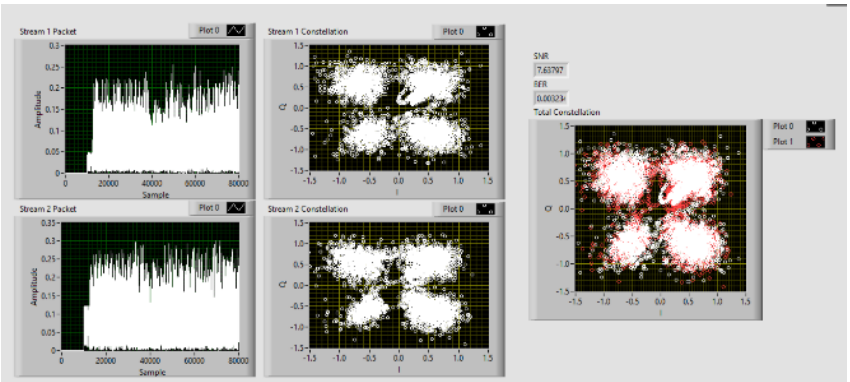

Figure 10. Received signal in scenario 1.

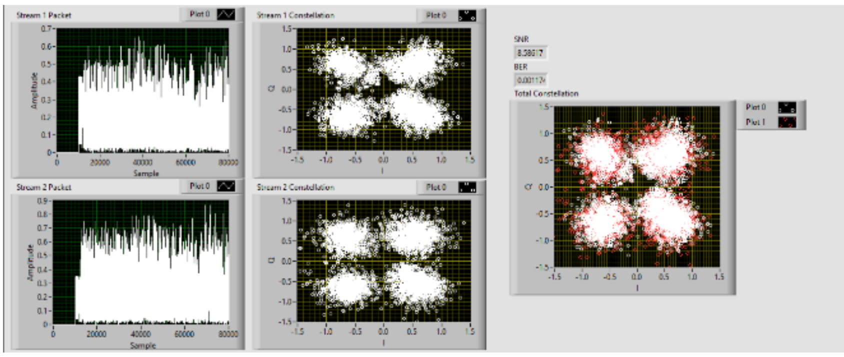

Figure 11. Received signal in scenario 2.

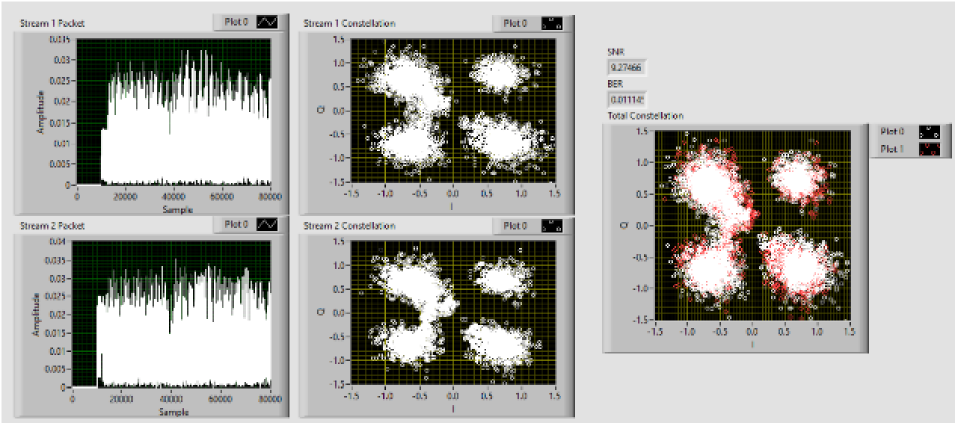

Figure 12. Received signal in scenario 3.

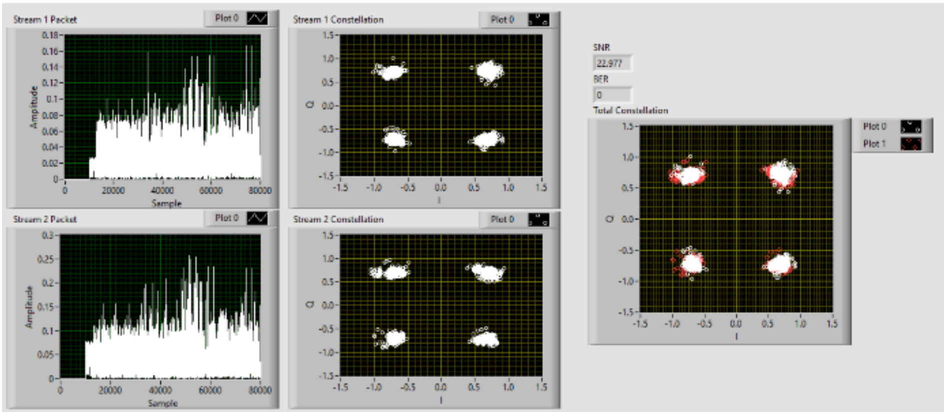

Figure 13. Received signal in scenario 4.

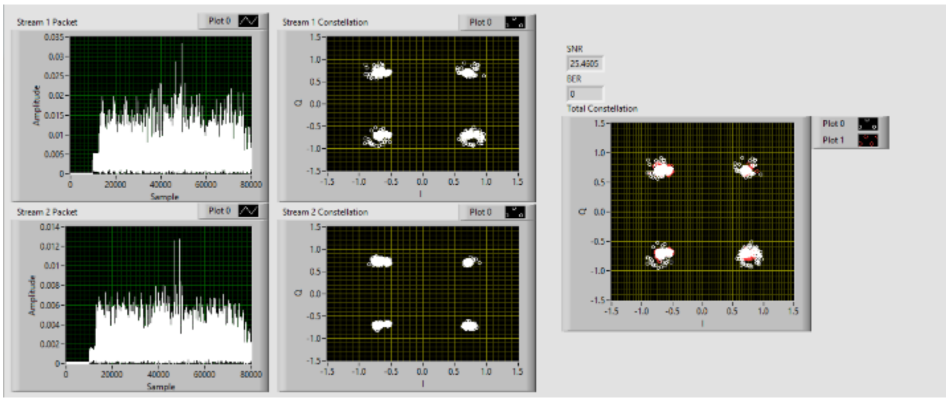

Figure 14. Received signal in scenario 5.

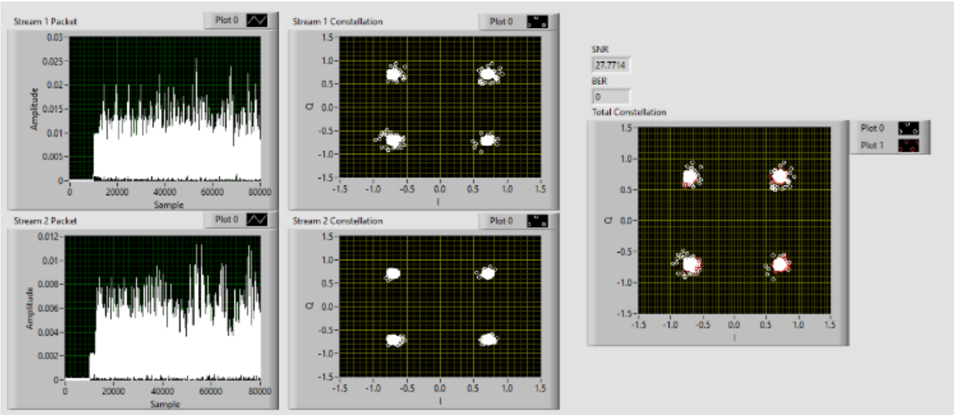

Figure 15. Received signal in scenario 6.

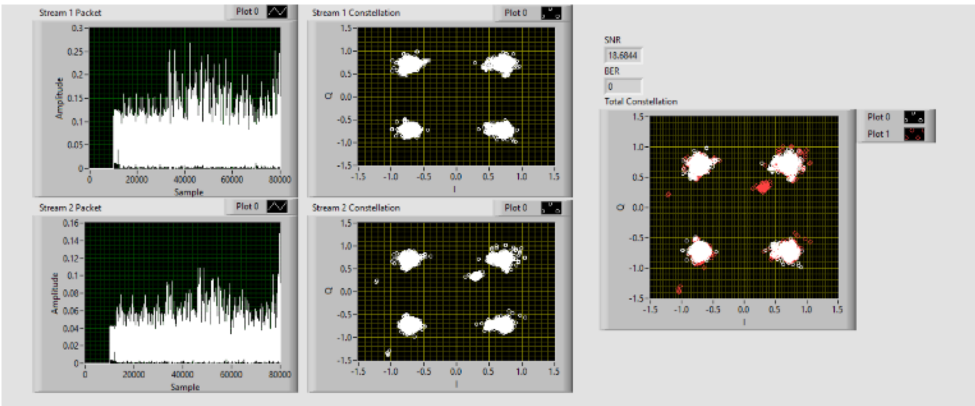

Figure 16. Received signal in scenario 7.

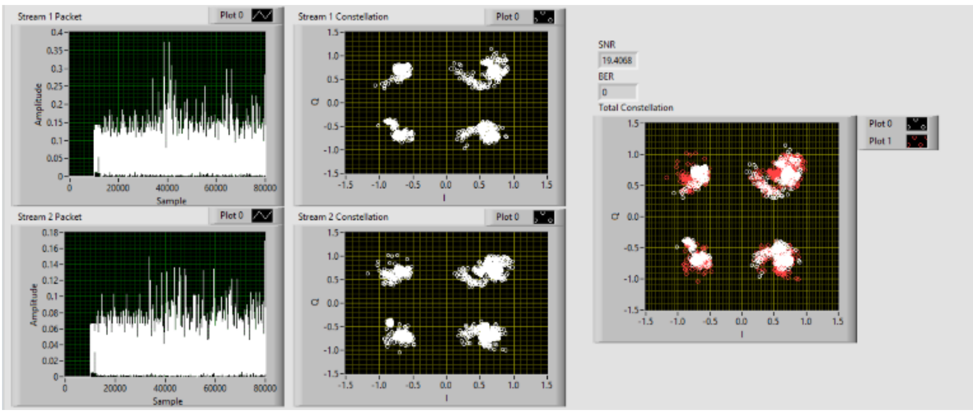

Figure 17. Received signal in scenario 8.

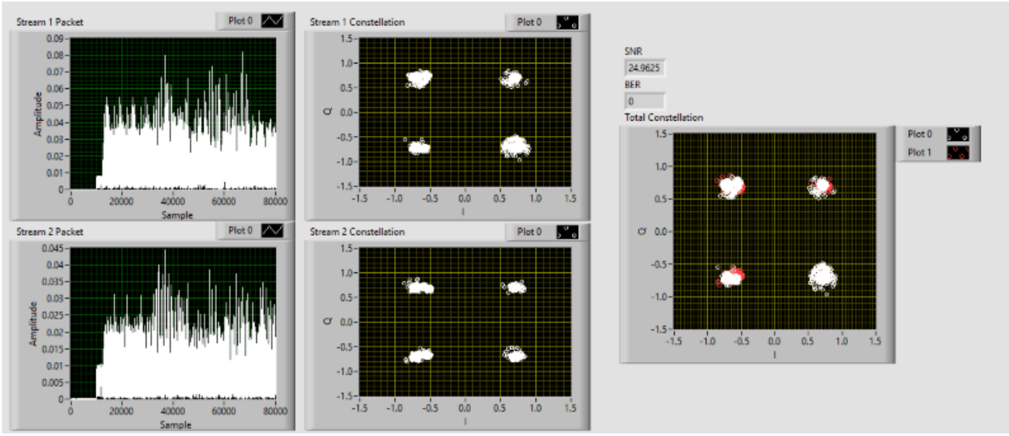

Figure 18. Received signal in scenario 9.

From the signal measurement data at the receiver for system scenarios 1 to 9, the results show that predistortion can improve the performance of the proposed system. This can be seen in the constellation signals from scenarios 4 to 9, which are less dispersive than those from scenarios 1 to 3. The scattered signal constellation indicates that the SNR at the receiver is small. From measurements 1 to 9, using both DPD and PD neural networks, the SNR and BER obtained at the receiver are better than without PD. In the evaluation of scenarios 1 to 9, several measurements were carried out to obtain more stable results. Measurements are made several times because the real-world channel conditions during transmission cannot be predicted or controlled. From the measurement results for each scenario, the average SNR and BER average values are presented in Table 7.

Table 7. SNR and BER value from the measurement process.

Measurement Scenario	Average SNR (dB)	Average BER
1	8.263	7.10×10^{-5}
2	8.228	2.38×10^{-4}
3	13.61	1.51×10^{-3}
4	16.312	2.71×10^{-5}
5	15.351	5.74×10^{-4}
6	22.192	1.14×10^{-3}
7	16.17	2.29×10^{-5}
8	15.86	1.61×10^{-4}
9	20.69	1.13×10^{-3}

From table 7, it can be seen that the best scenario is scenario 7. In this scenario, the system's average BER value is 2.29×10^{-5}, which is better than the other scenarios. Scenario 7 uses a PDNN and a coding rate of 1/2; thus, it can be said that the combination of a PDNN and a lower coding rate produces the best results. This is in accordance with the results of research [21], where the use of PDNN is better than DPD. Furthermore, from table 7, it can be seen that the worst scenario is scenario 3, where the BER value obtained is 1.51×10^{-3}. The data obtained for techniques using PD neural networks (scenarios 7 to 9) had better values than scenarios 1 to 3, but the SNR value was still lower than DPD (scenarios 4 to 6). From this experiment it can be seen that the use of a low code rate produces a better correction.

Because the signal measurement data at the receiver shows that PD improves the system, the following data will focus on the performance of systems that use DPD and PD neural networks. The previous data found that the DPD has a better SNR value than the PD Neural Network; therefore, the maximum SNR value is measured for system scenarios that use the DPD and PD Neural Networks. This data can be seen in Figure 19.

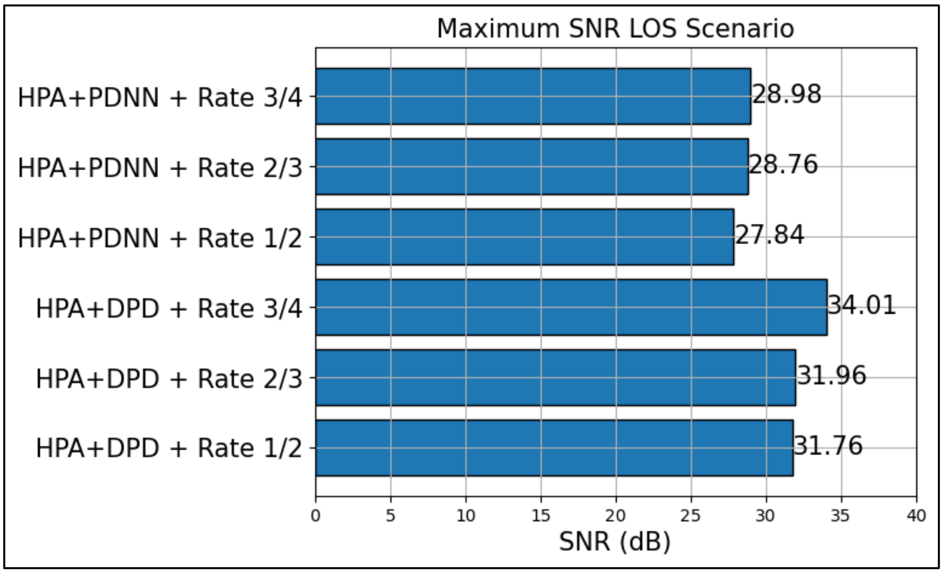

Figure 19. Maximum SNR value from the HPA+DPD and HPA+PDNN scenario.

Figure 19 shows that the maximum SNR value in systems using DPD is higher than in systems using PD neural networks. From figure 19, the highest SNR value is the SNR on the system using the HPA + DPD + Code Rate 3/4 scenario with an SNR value of 34.00789 dB, while the lowest SNR value is on the system using the HPA + PDNN + Code Rate 1/2 technique with an SNR value of 27.84357 dB. However, even though the DPD SNR value is much better than the PD Neural Network, the decrease in the BER graph of the system using PDNN is better than that using DPD. This can be seen in Figure 20.

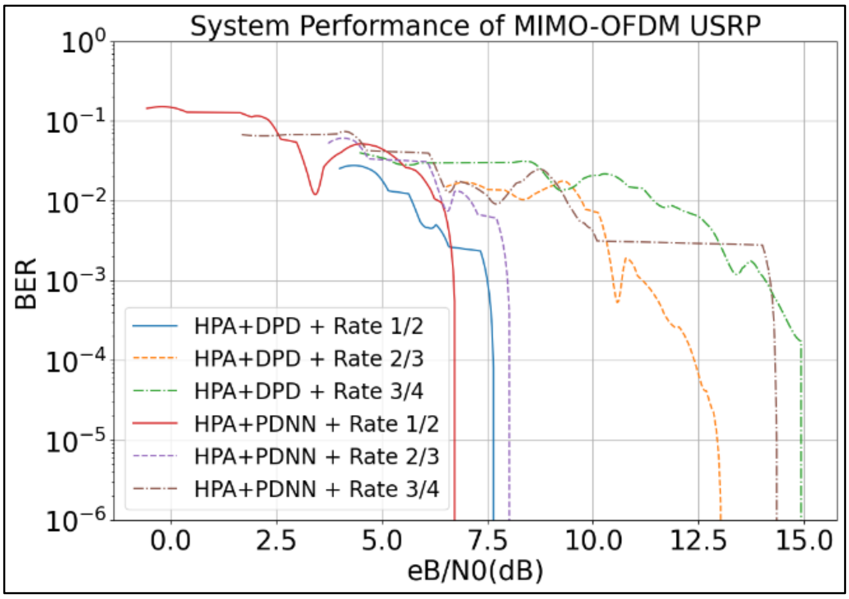

Figure 20. BER Plot from the HPA+DPD and HPA+PDNN scenario.

From Figure 20, it can be seen that the system scenario using PDNN has a better BER reduction graph compared to the system scenario using DPD. It can be seen that the system using the join technique of PDNN + code rate 1/2 c produces a better BER reduction graph than the other scenarios. Meanwhile, the join technique DPD + code rate 3/4 has a graph of a decrease in BER that is worse than the others. Apart from that, using several different code rates also produces a different BER graph. In figure 20, code rate 1/2 produces a graph of a decrease in BER that is better than the other code rates. This happens because the code rate 1/2 has more redundant bits than the different code rates. Because it has more redundant bits, the error detection and correction process at a code rate of 1/2 is better than using other code rates.

In addition, it can be seen that using PDNN at code rate 1/2 can reduce the SNR value to reach BER = 1×10^{-6} by 12.037% of the SNR value for BER = 0 on systems using DPD and code rate 1/2. Meanwhile, the PDNN and code rate 2/3 scenarios reduced the SNR value in the BER = 1×10^{-6} condition by 37.8% compared to the system values using the DPD and code rate 2/3 scenarios. Furthermore, the PDNN system scenario and code rate 3/4 can reduce the SNR value at BER = 1×10^{-6} by 4.10% compared to the system using the DPD scenario and code rate 3/4.

6. CONCLUSION

In this study, a joint technique evaluation of the PD and convolutional coding on MIMO-OFDM has been successfully carried out using the SDR device. With real-world channel conditions at the time of measurement, it was found that the use of PD and convolutional coding improved the quality of the MIMO-OFDM system, which was degraded due to the use of nonlinear HPA. Furthermore, from the evaluation of measurement data, the use of digital distortion (DPD) produces a better maximum SNR value than the use of PDNN. However, even so, the graph of decreasing BER on PDNN shows better results than DPD for all code rate scenarios. In addition, at code rates 1/2, 2/3, and 3/4, PD neural networks can reduce SNR values to reach BER = 1×10^{-6} in 12.037%, 37.8%, and 4.10% of the systems using DPD, respectively. Thus, it can be said that the combination of PD neural networks and convolutional coding can produce better system performance than the combination of conventional DPD and convolutional coding techniques in the proposed system. This result is the same as the research in [21]. Furthermore, this study showed that a code rate of 1/2 resulted in a better BER reduction graph than the other code rates because the code rate of 1/2 had more redundant bits than the different code rates.

REFERENCES

[1] **The State of Mobile Internet Connectivity Report 2022 [Internet]**, *GSMA*, 2022 [cited 9 March 2023], Available from: https://www.gsma.com/r/somic/.

[2] **Ericsson Mobility Report: More than half a billion 5G subscriptions by the end of 2021 [Internet]**, *erricson*, 2021 [cited 9 March 2023], Available from: https://www.ericsson.com/en/press-releases/2/2021/6/ericsson-mobility-report-indonesia-overview.

[3] Arun KS, Neelam S, Saurabh D. **Optimizing Resource Allocation of MIMO-OFDM in 4G and Beyond Systems**. *Advances in VLSI, Communication, and Signal Processing*, 2020.

[4] Borges D, Montezuma P, Dinis R, Beko M. **Massive MIMO Techniques for 5G and Beyond—Opportunities and Challenges**. *Electronics*, 2021.

[5] Cisco Systems Inc. **802.11ac: The Fifth Generation of Wi-Fi**. *Cisco*, 2012.

[6] Cisco Systems Inc. **IEEE 802.11ax: The Sixth Generation of Wi-Fi**. *Cisco*, 2019.

[7] Ali F, Tan JM, Liau CF, Ervin GM, Manas KH. **The Nonlinearity Effect of High Power Amplitude on OFDM Signal and Solution to Solve by Using PAPR Reduction**. *International Conference on Advances in Computing, Communication, & Automation (ICACCA*), 2018.

[8] Wang X, Li Y, Yu C, Hong W, Zhu A. **Digital predistortion of 5G massive MIMO wireless transmitters based on indirect identification of power amplifier behaviour with OTA tests**. *IEEE Trans Microw Theory Tech*, 2020.

[9] Liu Z, Hu X, Xu L, Wang W, Ghannouchi FM. **Low Computational Complexity Digital Predistortion Based on Convolutional Neural Network for Wideband Power Amplifiers**. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022.

[10] Srinu P, Padma RK, Bala SN. **Capacity and BER performance improvement in integrated MIMO-OFDM system using optimal**

power allocation, channel estimation, and turbo coding. *International Journal of Communication Systems*, 2021.
[11] Aravinda BT, Deergha RK. **Performance Analysis of LDPC Coded Massive MIMO-OFDM System**. *International Conference for Emerging Technology (INCET)*, 2020.
[12] Arun A, Saurabh NM. **PC-CC: An advancement in forward error correction using polar and convolutional codes for MIMO-OFDM system**. *Journal of King Saud University - Computer and Information Sciences*, 2020.
[13] Parag A, Vivek AB. **End-to-End Theoretical Evaluation of a Nonlinear MIMO-OFDM System in the Presence of Digital Predistorter**. *IEEE System Journal*, 2019.
[14] Tingyu H, Weijun H, Tingxiao C, Huanhuan L. **RLS-DPD Algorithm for Hybrid Precoding Architecture in MIMO-OFDM systems**. *IEEE 3rd International Conference on Electronic Information and Communication Technology*, 2020.
[15] Arun A, Saurabh NM. **Performance analysis and design of MIMO-OFDM system using concatenated forward error correction codes**. *Journal of Central South University*, 2017.
[16] Ashish T, Munish R. **Implementation of convolution coded OFDM through different channel models on SDR platform**. *International Journal of Information Technology*, 2018.
[17] Hendy B, I GPA, Amang S. **An Implementation of Error Minimization Data Transmission in OFDM using Modified Convolutional Code.** *EMITTER International Journal of Engineering Technology*, 2015.
[18] Zhenzi W, Chen G, Dake L. **Computational Complexity Analysis of FEC Decoding on SDR Platforms**. *Journal of Signal Processing Systems*, 2016.
[19] Ridlo Q, I WM, Satriyo D. **Performance Comparison of SISO and MIMO-OFDM Based on SDR Platform**. *3rd International Conference on Science and Technology - Computer (ICST)*, 2017.
[20] Krishna KSP, Sagar S, Shine KP, Prem KN. **Implementation of Digital Pre-Distortion for Power Amplifier Linearisation in Software Defined Radio**. *Twenty-third National Conference on Communications (NCC)*, 2017.
[21] Muhammad WG, Naufal AP, Melki MG, Arifin, Yoedy M, Hendy B. **Evaluation of the predistortion technique by neural network algorithm in MIMO-OFDM system using USRP**. *Journal INFOTEL*, 2022.
[22] C. Tarver, A. Balatsoukas-Stimming and J. R. Cavallaro, "**Design and Implementation of a Neural Network Based Predistorter for Enhanced Mobile Broadband**," *2019 IEEE International Workshop on Signal Processing Systems (SiPS), Nanjing, China*, 2019.
[23] Ravibabu T, Dharma RC. **BER Analysis of spatial multiplexing and STBC MIMO-OFDM System**. *4th International Conference on Devices, Circuits and Systems (ICDCS)*, 2018.

[24] Ali SA, Mustafa MH, Mohammed SA, Arshad MK, Maha F, Shihab HK. **Channel Estimation using LS and MMSE Channel Estimation Techniques for MIMO-OFDM Systems**. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022.

[25] Abdelhamid R, Mohamed B, Moha MH. **Least Squares Channel Estimation of an OFDM Massive MIMO System for 5G Wireless Communications**. *Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18)*, 2018.

[26] Kun CH, Ana GA. **Low-Complexity Computation of Zero-Forcing Equalizers for Massive MIMO-OFDM**. *IEEE Conference on Vehicular Technology (VTC)*, 2019.

[27] Adel AMS. **Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers**. *IEEE Transactions on Communications*, 1981.

[28] Christian G, Xingquan Z, Oge M. **Dropout vs batch normalisation: an empirical study of their impact to deep learning**. *Multimedia Tools and Applications*, 2020.

[29] Gaurav S, Gaurav M. **Experimental Investigation of Spectrum Sensing for LTE Frequency Band based on USRP 2920/VST 5644**. *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2016.

[30] Gaurav S, Gurpreet K, Banga VK**. Implementation & BER Analysis of 2×2 MIMO Using USRP 2920-Universal Software Radio Peripheral**. *Second International Conference on Computational Intelligence & Communication Technology*, 2016.

[31] Fadhel MG, Oualid H, Mohamed H. **Behavioral Modeling and Predistortion of Wideband Wireless Transmitters**. *Wiley*, 2015.