DPD of a 3-ways PA with Memory Effects Based on Neural Network

任课老师：余超

小组成员：
钟　源　贾岩森

信息科学与工程学院

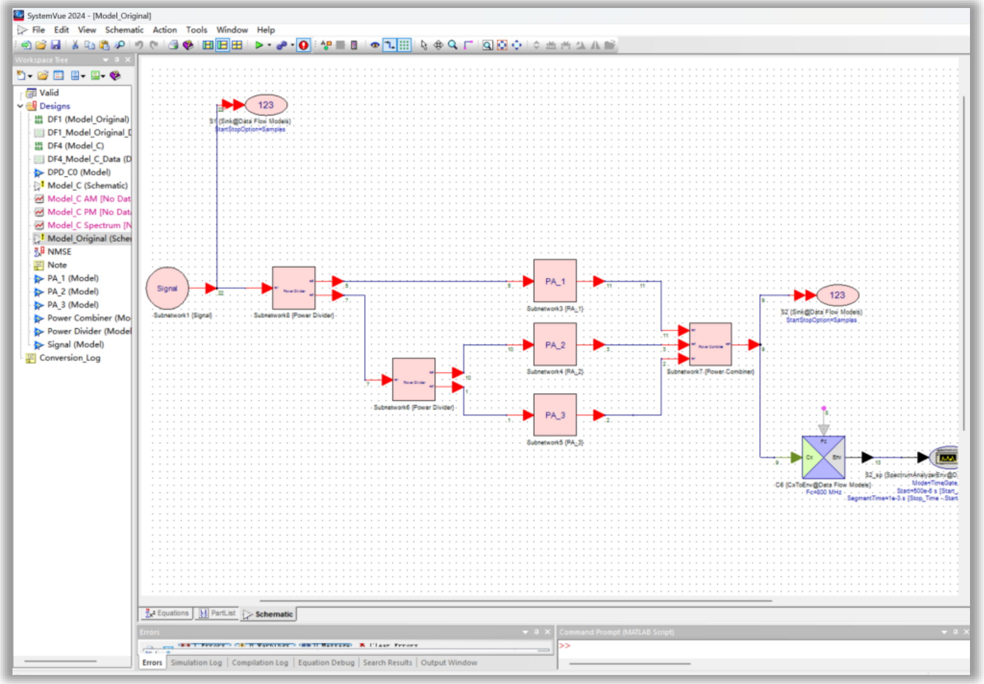
2025年5月23日

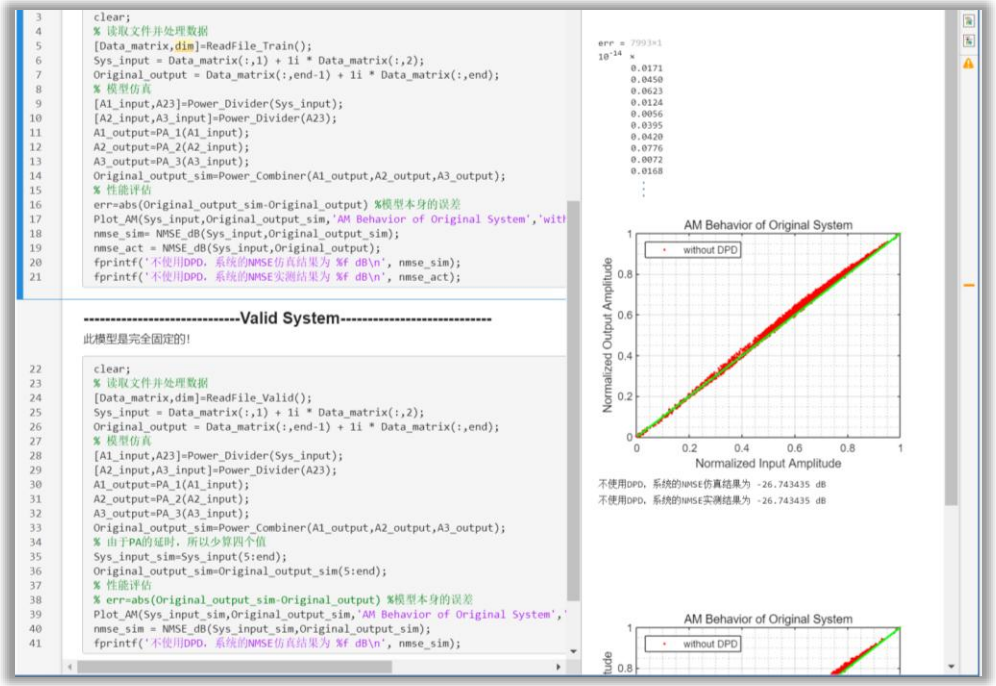
Contents

東南大學 SOUTHEAST UNIVERSITY

Highly consistent results

不使用DPD，MATLAB仿真系统的NMSE为 -26.743435 dB
不使用DPD，System Vue仿真系统的NMSE为 -26.743435 dB

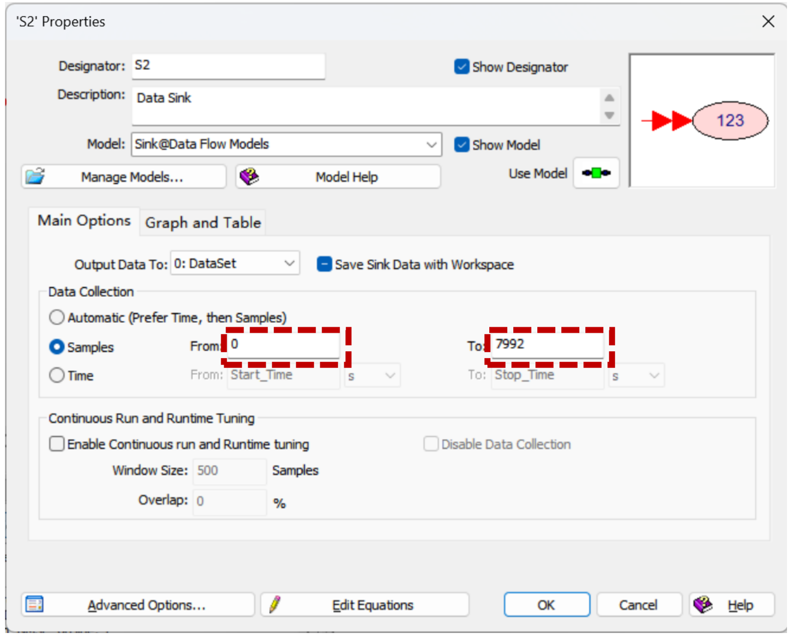




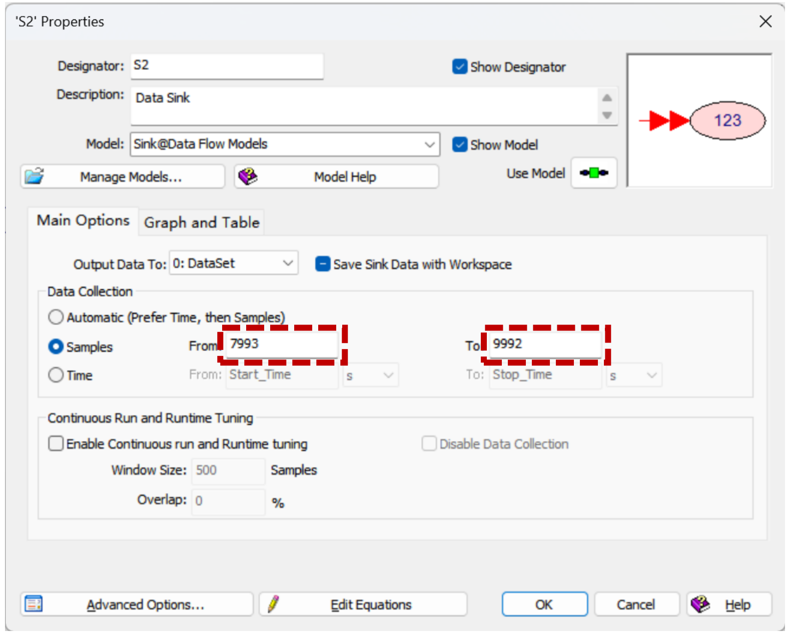
PART 01. Introduction

To better measure the generalization ability of the network model, we set up two data sets.

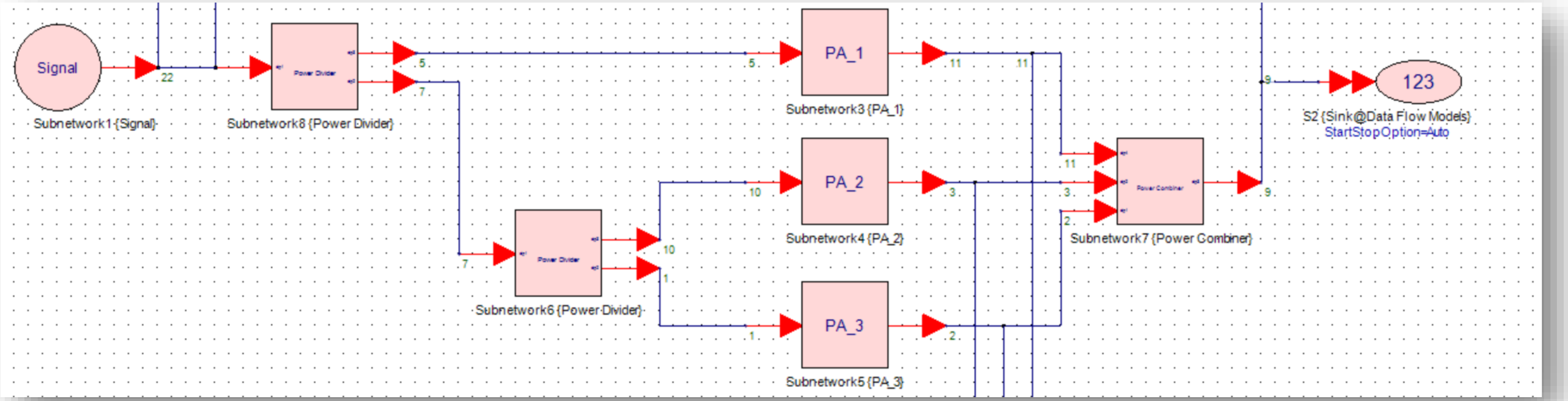
Train Set



Valid Set



It should be noted that the Train Set is **NOT** really the data set for DPD training.



Design Methods

- FCN
- LSTM
- Bi-LSTM
- GRU
- GNN-LSTM
- Multi-layer FCN
- TCN
- Transformer
- Time Delay Net
- Volterra Series

Optimization Algorithm

- Adam
- SGDM
- RMSProp
- trainlm
- trainscg
- trainbr

PART 02 Design Method: Fully Connected Network (FCN)

SOUTHEAST UNIVERSITY

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 15000 (Complete on 11320)

Network Structure

Hidden Layer 1: 128 FCN hidden units

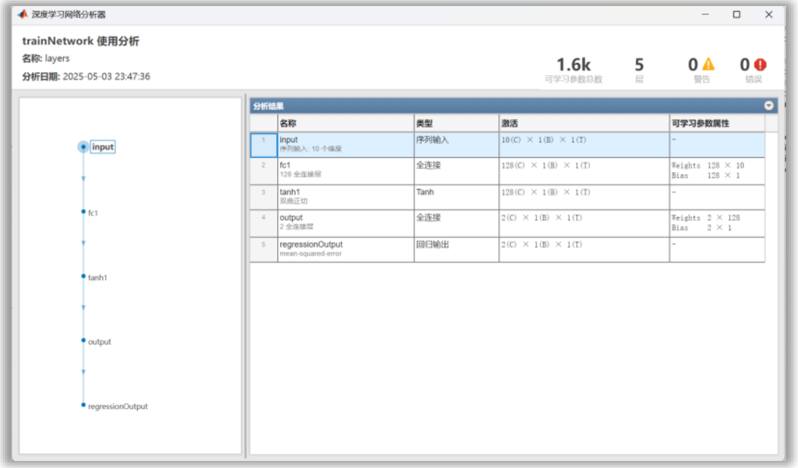
Activation Function 1: tanh

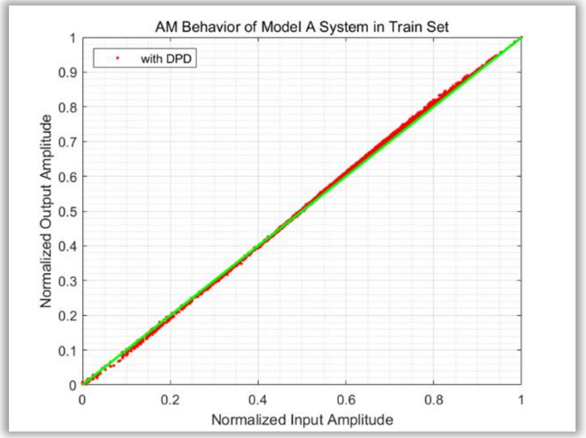
Results

NMSE on Fitting Accuracy: **-35.315dB**

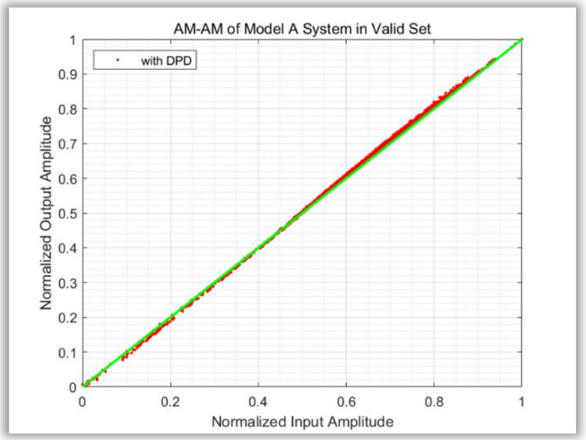
NMSE on Train Set: **-35.276dB**

NMSE on Valid Set: **-35.324dB**





Train Set



Valid Set

PART 02 Design Method: Long Short-Term Memory (LSTM)

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 4000 (Complete on 900)

Network Structure

Hidden Layer 1: 128 LSTM hidden units

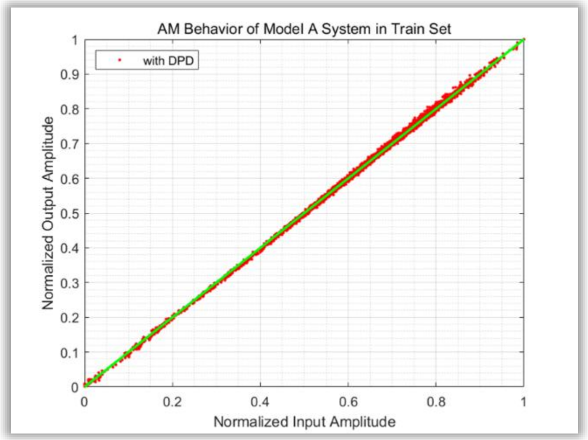
Activation Function 1: tanh

Results

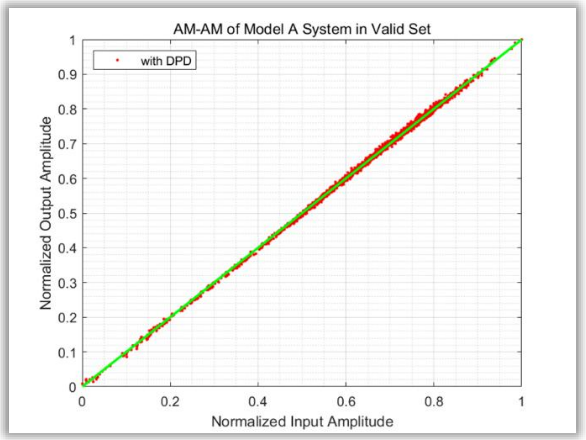
NMSE on Fitting Accuracy: **-42.341dB** (Overfitting)

NMSE on Train Set: **-38.654dB**

NMSE on Valid Set: **-38.929dB**



Train Set



Valid Set

SOUTHEAST UNIVERSITY

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 4000 (Complete on 840)

Network Structure

Hidden Layer 1: 128 Bi-LSTM hidden units

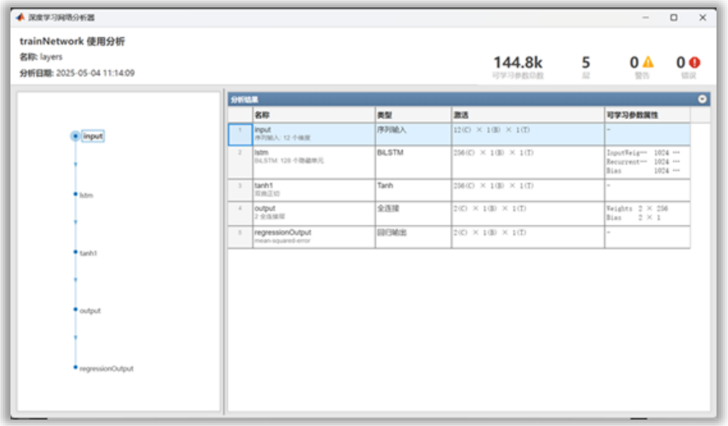
Activation Function 1: tanh

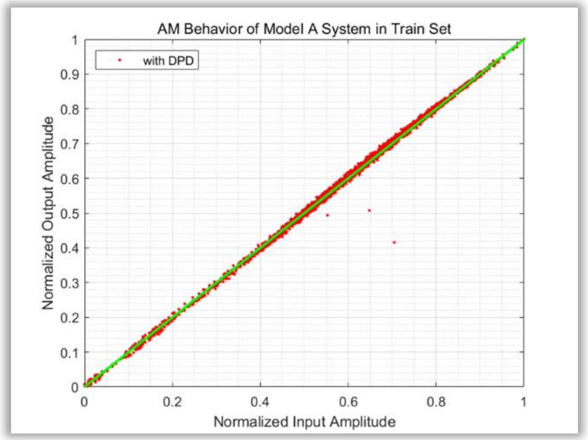
Results

NMSE on Fitting Accuracy: **-44.254dB** (Overfitting)

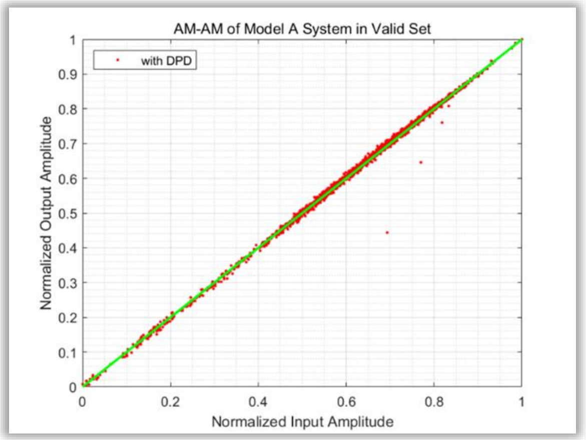
NMSE on Train Set: **-33.932dB**

NMSE on Valid Set: **-33.205dB**





Train Set



Valid Set

PART 02 Design Method: Gated Recurrent Unit (GRU)

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 4000 (Complete on 1640)

Network Structure

Hidden Layer 1: 64 GRU hidden units

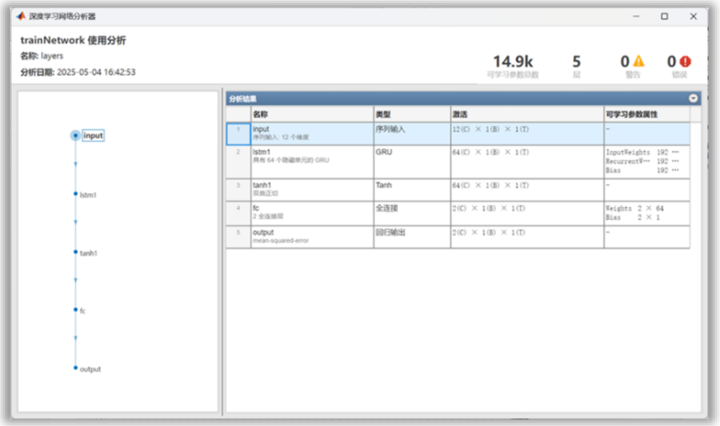
Activation Function 1: tanh

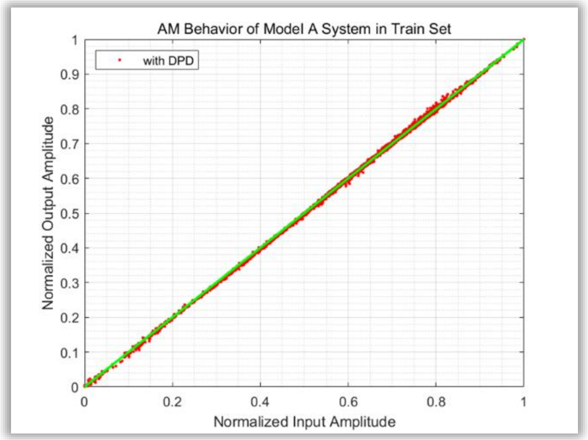
Results

NMSE on Fitting Accuracy: **-47.393dB**(Overfitting)

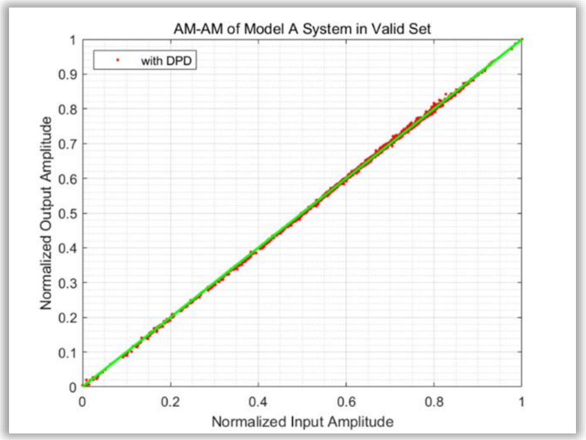
NMSE on Train Set: **-43.417dB**

NMSE on Valid Set: **-43.694dB**





Train Set



Valid Set

PART 02 Design Method: CNN-LSTM

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 4000 (Complete on 1160)

Network Structure

Hidden Layer 1: 64 conv hidden units

Activation Function 1: ReLU

Hidden Layer 2: 64 LSTM hidden units

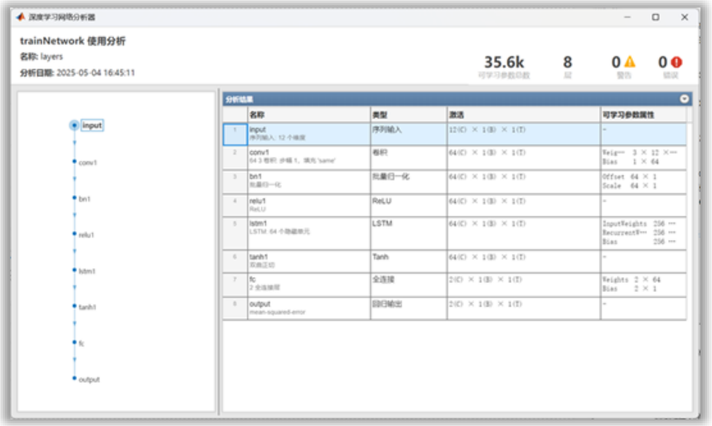
Activation Function 2: tanh

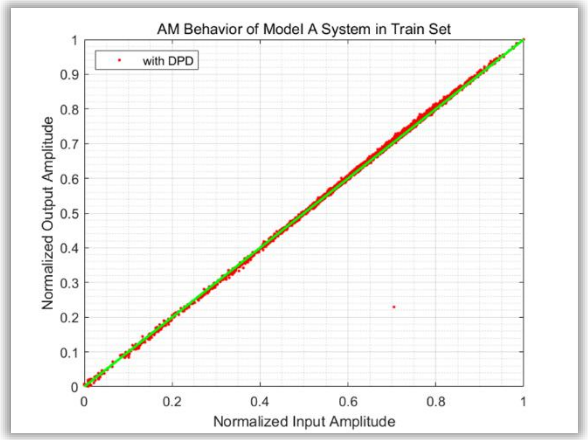
Results

NMSE on Fitting Accuracy: **-38.850dB**

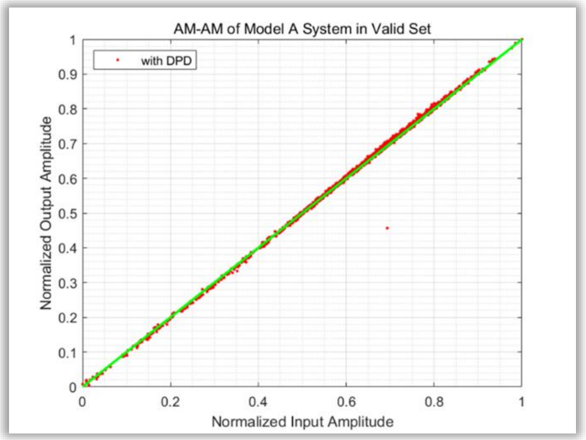
NMSE on Train Set: **-36.615dB**

NMSE on Valid Set: **-36.823dB**





Train Set



Valid Set

PART 02 Design Method: Multi-layer FCN

東南大學 SOUTHEAST UNIVERSITY

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 5

Max Epochs: 15000 (Complete on 12360)

Initial Learn Rate: 0.005

Results

NMSE on Fitting Accuracy: **-46.003dB**

NMSE on Train Set: **-44.859dB**

NMSE on Valid Set: **-45.179dB**

Network Structure

Hidden Layer 1: 128 FCN hidden units

Activation Function 1: tanh

Hidden Layer 2: 64 FCN hidden units

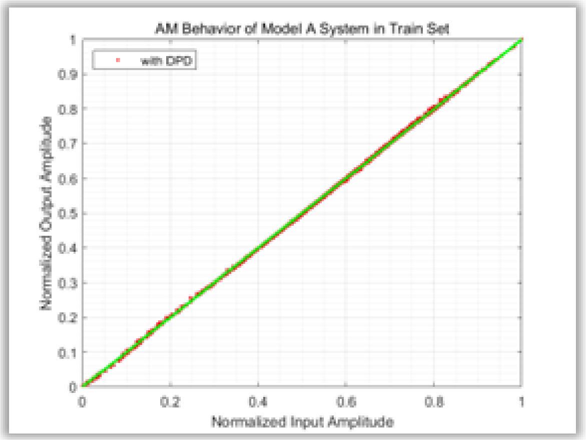
Activation Function 2: tanh

Hidden Layer 3: 32 FCN hidden units

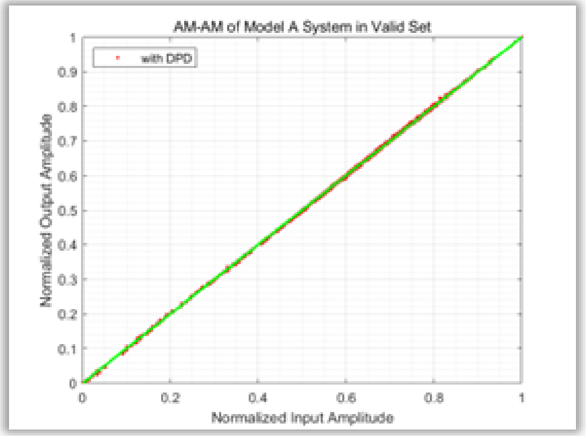
Activation Function 3: tanh

Hidden Layer 4: 16 FCN hidden units

Activation Function 4: tanh



Train Set



Valid Set

PART 02 Design Method: Temporal Convolutional Network (TCN)

東南大學 SOUTHEAST UNIVERSITY

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 4

Max Epochs: 10000 (Complete on 6000)

Network Structure

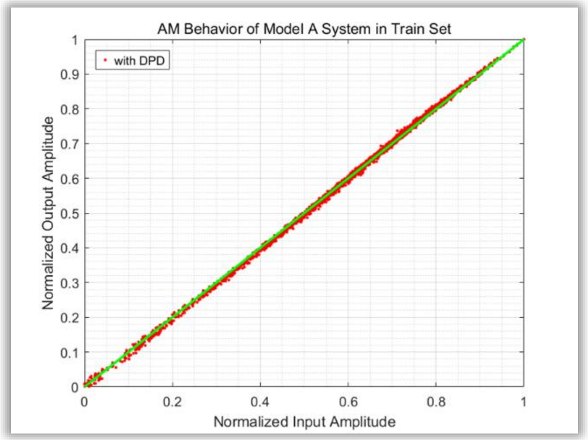
Using default TCN in MATLAB

Results

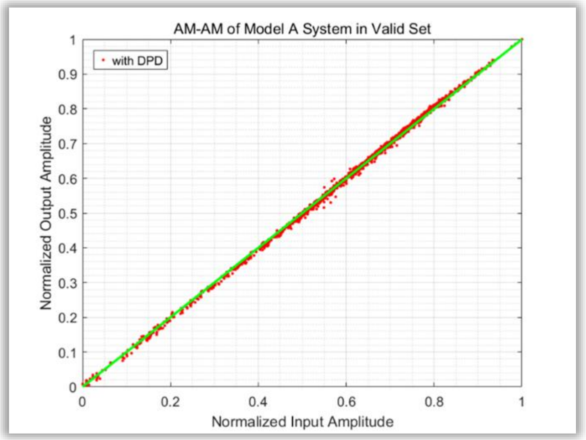
NMSE on Fitting Accuracy: **-42.507dB**(Overfitting)

NMSE on Train Set: **-38.627dB**

NMSE on Valid Set: **-37.308dB**



Train Set



Valid Set

15.6.3.4 Transformer模型

Transformer模型 [Vaswani et al., 2017] 是一个基于多头自注意力的序列到序列模型，其整个网络结构可以分为两部分：

（1）编码器只包含多层的多头自注意力（Multi-Head Self-Attention）模块，每一层都接受前一层的输出作为输入．编码器的输入为序列 $\boldsymbol{x}_{1:S}$，输出为一个向量序列 $\boldsymbol{H}^{\mathrm{enc}} = [\boldsymbol{h}_1^{\mathrm{enc}}, \cdots, \boldsymbol{h}_S^{\mathrm{enc}}]$．然后，用两个矩阵将 $\boldsymbol{H}^{\mathrm{enc}}$ 映射到 $\boldsymbol{K}^{\mathrm{enc}}$ 和 $\boldsymbol{V}^{\mathrm{enc}}$ 作为键值对供解码器使用，即

$$\boldsymbol{K}^{\mathrm{enc}} = \boldsymbol{W}_k' \boldsymbol{H}^{\mathrm{enc}}, \tag{15.118}$$

$$\boldsymbol{V}^{\mathrm{enc}} = \boldsymbol{W}_v' \boldsymbol{H}^{\mathrm{enc}}, \tag{15.119}$$

其中 $\boldsymbol{W}_k'$ 和 $\boldsymbol{W}_v'$ 为线性映射的参数矩阵．

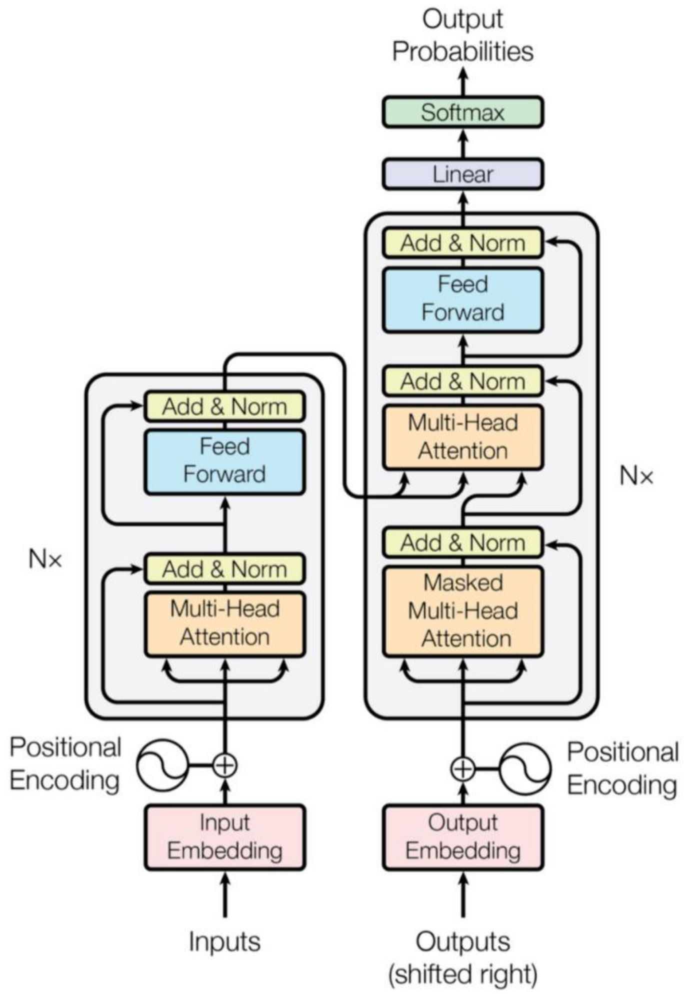
（2）解码器是通过自回归的方式来生成目标序列．和编码器不同，解码器由以下三个模块构成：

a）掩蔽自注意力模块：第 t 步时，先使用自注意力模型对已生成的前缀序列 $\boldsymbol{y}_{0:(t-1)}$ 进行编码得到 $\boldsymbol{H}^{\mathrm{dec}} = [\boldsymbol{h}_1^{\mathrm{dec}}, \cdots, \boldsymbol{h}_t^{\mathrm{dec}}]$．

b）解码器到编码器注意力模块：将 $\boldsymbol{h}_t^{\mathrm{dec}}$ 进行线性映射得到 $\boldsymbol{q}_t^{\mathrm{dec}}$．将 $\boldsymbol{q}_t^{\mathrm{dec}}$ 作为查询向量，通过键值对注意力机制来从输入 $(\boldsymbol{K}^{\mathrm{enc}}, \boldsymbol{V}^{\mathrm{enc}})$ 中选取有用的信息．

c）逐位置的前馈神经网络：使用一个前馈神经网络来综合得到所有信息．

将上述三个步骤重复多次，最后通过一个全连接前馈神经网络来计算输出概率．图15.6给出了 Transformer 的网络结构示例，其中 $N\times$ 表示重复 N 次，“Add & Norm” 表示残差连接和层归一化．在训练时，为了提高效率，我们通常将右移的目标序列（Right-Shifted Output）$\boldsymbol{y}_{0:(T-1)}$ 作为解码器的输入，即在第 t 个位置的输入为 y_{t-1}．在这种情况下，可以通过一个掩码（Mask）来阻止每个位置选择其后面的输入信息．这种方式称为掩蔽自注意力（Masked Self-Attention）．



PART 02 Design Method: Transformer

SOUTHEAST UNIVERSITY 東南大學

Parameter Settings

Optimization Algorithm: Adam

Memory Depth: 4

Max Epochs: 8000 (Complete on 1000)

Network Structure

Shown in the image on the right

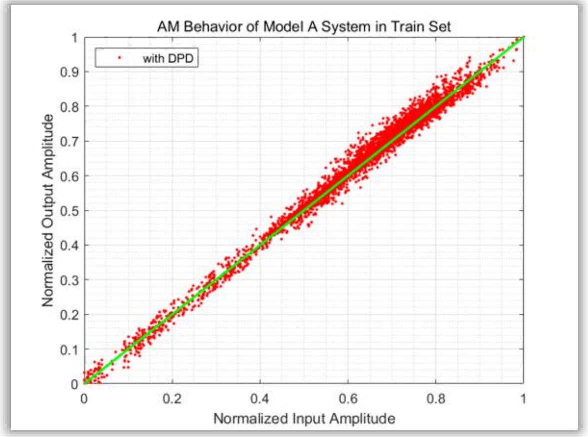
Results

NMSE on Fitting Accuracy: **-28.096dB** **(Failed)**

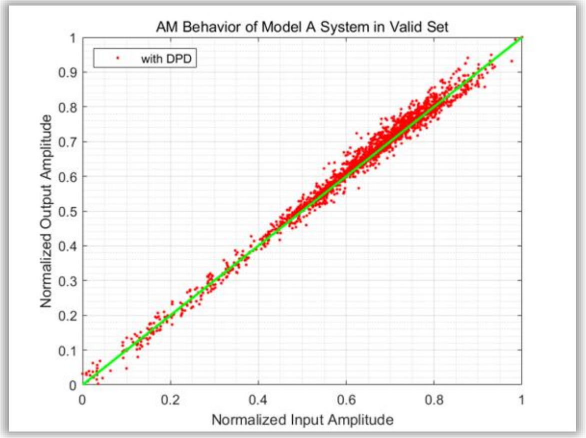
NMSE on Train Set: **-25.923dB**

NMSE on Valid Set: **-26.051dB**

	名称	类型	激活	可学习参数大小
1	input 序列输入: 10 个维度	序列输入	10(C) × 1(B) × 1(T)	-
2	pos-emb 具有 10 个输出通道且最大位置为 8192 ...	位置编码	10(C) × 1(B) × 1(T)	Weights 10 ×
3	add 加法: 将 2 个输入按元素相加	加法	10(C) × 1(B) × 1(T)	-
4	selfattention_1 具有 4 个头和 128 个键和查询通道的自...	自注意力	10(C) × 1(B) × 1(T)	QueryWeig··· KeyWeights ValueWeig··· OutputWei··· QueryBias KeyBias ValueBias OutputBias
5	layer @(x)x	函数	10(C) × 1(B) × 1(T)	-
6	fc1 32 全连接层	全连接	32(C) × 1(B) × 1(T)	Weights 32 × Bias 32 ×
7	fc_relu ReLU	ReLU	32(C) × 1(B) × 1(T)	-
8	fc2 2 全连接层	全连接	2(C) × 1(B) × 1(T)	Weights 2 × Bias 2 ×
9	output mean-squared-error	回归输出	2(C) × 1(B) × 1(T)	-



Train Set



Valid Set

PART 02 Design Method: Time Delay Net

東南大學
SOUTHEAST UNIVERSITY

Parameter Settings

Optimization Algorithm: Levenberg-Marquardt

Memory Depth: 4

Max Epochs: 4000 (Complete on 600)

Network Structure

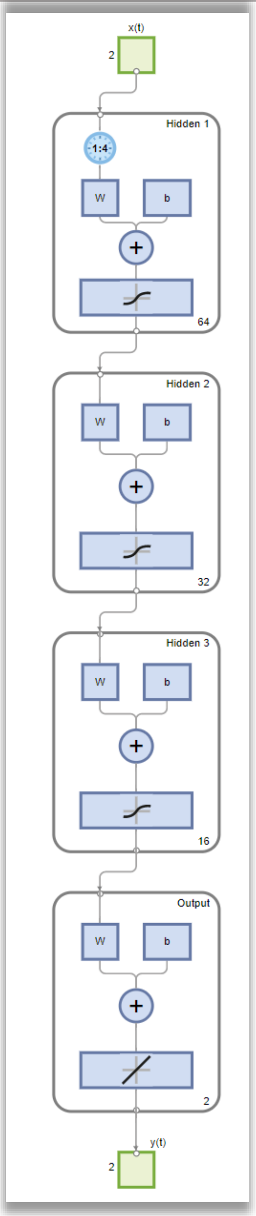
Shown in the image on the right

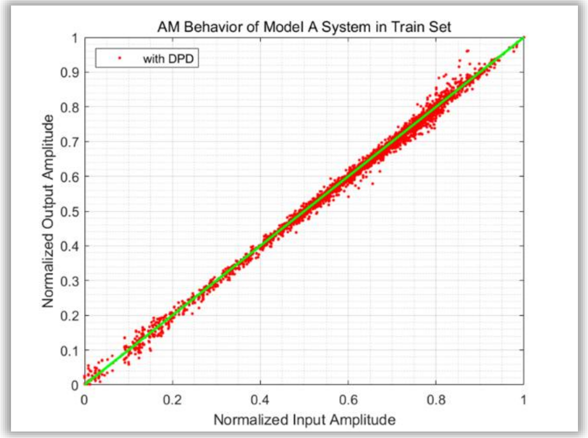
Results

NMSE on Fitting Accuracy: **-44.655dB** (Overfitting)

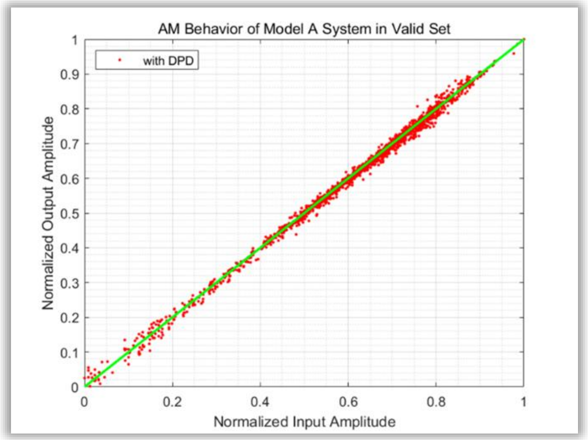
NMSE on Train Set: **-30.378dB**

NMSE on Valid Set: **-30.719dB**





Train Set



Valid Set

PART 02 Design Method: Volterra Series

Core Function

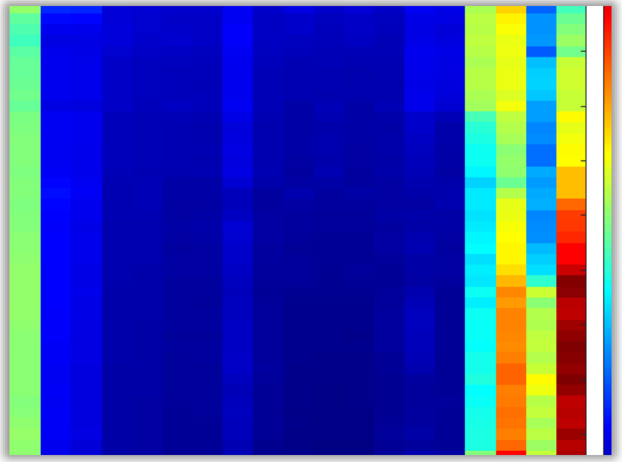
```
function DPD_coefficient=Train_Voltrra(M,N,dim,input,output)
output=output/max(abs(output));
input=input/max(abs(input));
DPD_input_matrix = zeros(dim, (M+1)*N);
for i=1:M+1
    for j= 1:N
        DPD_input_matrix(:,N*(i-1)+j) = Delay(output,…
            i-1).*Delay(output,i-1)).^(2*j-2));
    end
end
DPD_coefficient=DPD_input_matrix\input(:);
end
```

Results

NMSE on Fitting Accuracy: **-34.707dB**

NMSE on Train Set: **-32.241dB**

NMSE on Valid Set: **-32.349dB**

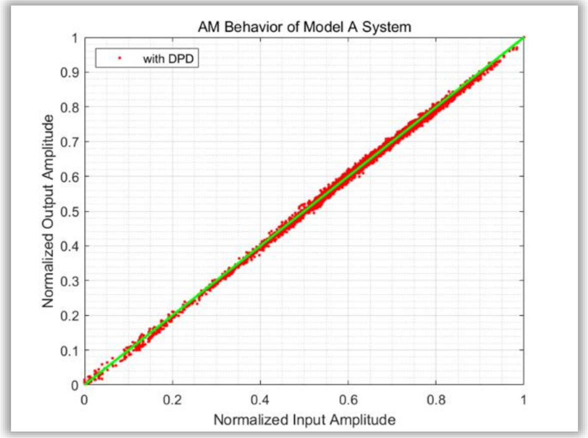


Parameter Searching

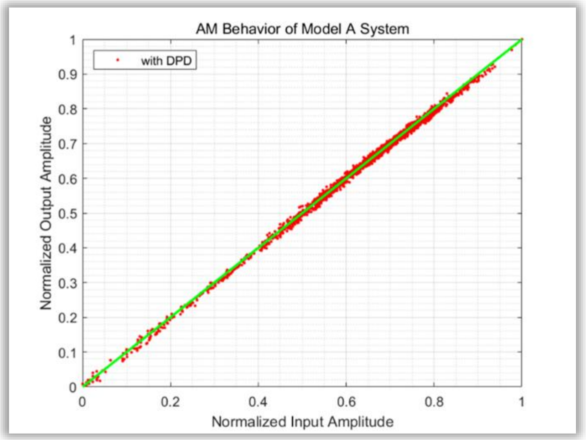
Range: M∈[1,50] N∈[1,20]

When L=13,

Larger M, better performance



Train Set



Valid Set

東南大學 SOUTHEAST UNIVERSITY

Performance comparison of different methods

Method	Fitting NMSE(dB)	Train Set NMSE(dB)	Valid Set NMSE(dB)
Original	---	-26.743	-26.794
FCN (M=5)	-35.315	-35.276	-35.324
LSTM (M=5)	-42.341	-38.654	-38.929
Bi-LSTM (M=5)	-44.254	-33.932	-33.205
GRU (M=5)	-47.393	-43.417	-43.694
CNN-LSTM (M=5)	-38.850	-36.615	-36.823
Multi-layer FCN (M=5)	-46.003	-44.859	-45.179
TCN (M=4)	-42.507	-38.627	-37.308
Transformer (M=4)	-28.096	-25.923	-26.051
Timedelay (M=4)	-44.655	-30.378	-30.719
Volterra (M=100, L=13)	-34.707	-32.241	-32.349

Best Performance: Multi-layer FCN (M=5)

PART 03. Optimization Algorithm (Using in Multi-layer FCN)

Stochastic Gradient Descent with Momentum (sgdm)

- **Parameter Settings**

Memory Depth: 5

Initial Learning Rate: 0.01

Momentum Factor: 0.9

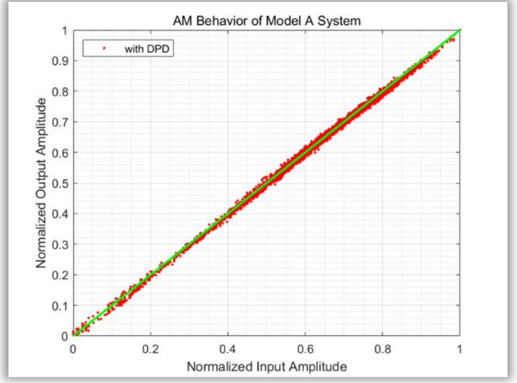
Max Epochs: 5000

- **Results**

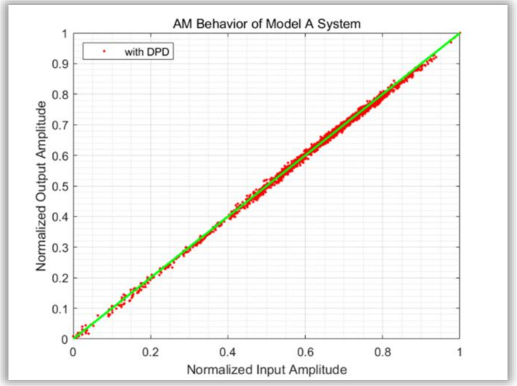
NMSE on Fitting: **-42.238dB**

NMSE on Train Set: **-41.339dB**

NMSE on Valid Set: **-41.572dB**



Train Set



Valid Set

Root-Mean-Square Propagation (rmsprop)

- **Parameter Settings**

Memory Depth: 5

Initial Learning Rate: 0.01

Momentum Factor: 0.9

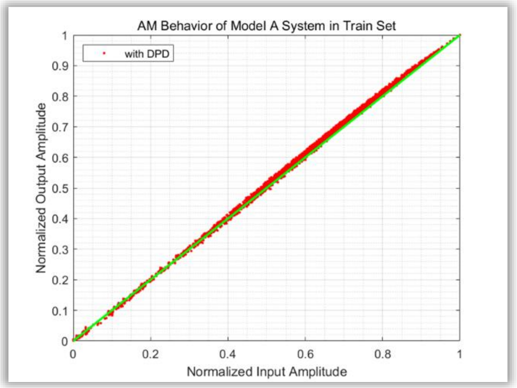
Max Epochs: 15000(Complete on 11900)

- **Results**

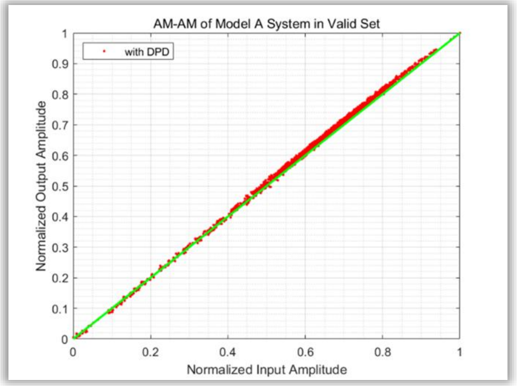
NMSE on Fitting: **-33.357dB**

NMSE on Train Set: **-33.357dB**

NMSE on Valid Set: **-33.364dB**



Train Set



Valid Set

PART 03. Optimization Algorithm (Using in Multi-layer FCN)

Levenberg-Marquardt (trainlm)

- **Parameter Settings**

  Memory Depth: 5

- **Results**

  NMSE on Fitting: **-54.500dB**

  (Overfitting)

  NMSE on Train Set: **-45.662dB**

  NMSE on Valid Set: **-46.046dB**

- **Improve**

  Simplify the network structure!

- **Parameter Settings**

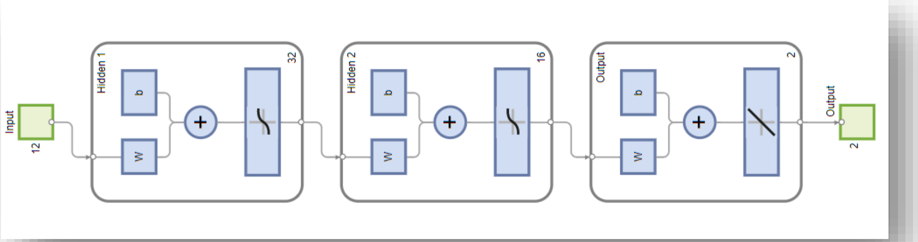
  Memory Depth: 5

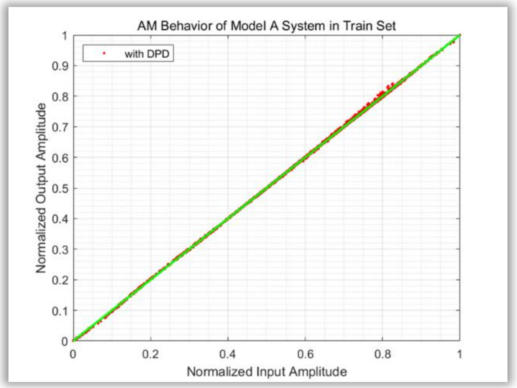
- **Results**

  NMSE on Fitting: **-54.067dB**

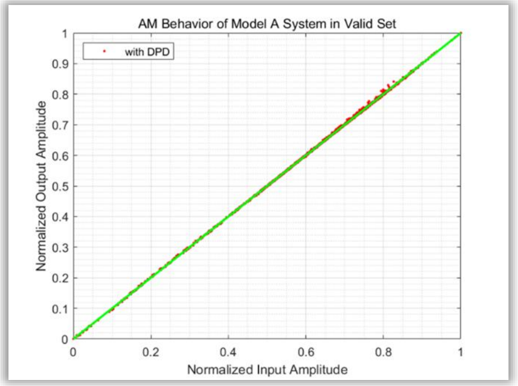
  NMSE on Train Set: **-49.004dB**

  NMSE on Valid Set: **-46.381dB**





Train Set



Valid Set

Scaled Conjugate Gradient (trainscg)

- **Parameter Settings**

  Memory Depth: 5

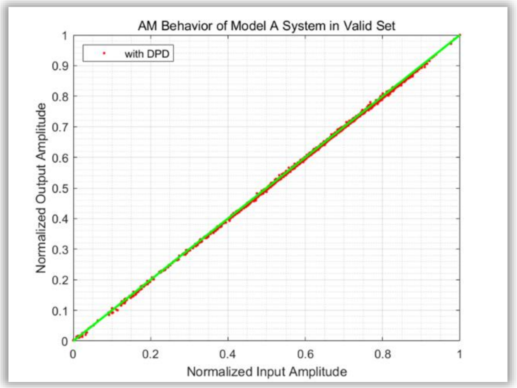
  Momentum Factor: 0.9

- **Results**

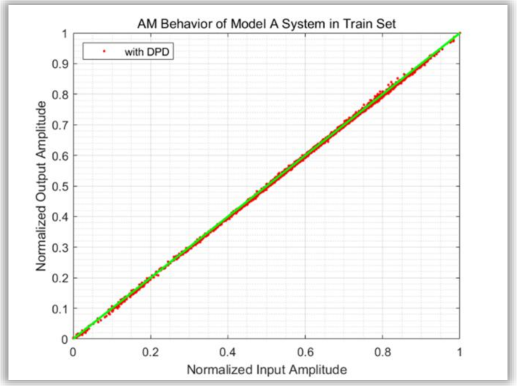
  NMSE on Fitting: **-41.556dB**

  NMSE on Train Set: **-41.540dB**

  NMSE on Valid Set: **-41.833dB**



Train Set



Valid Set

Bayesian Regularization (trainbr)

- **Parameter Settings**

  Memory Depth: 4

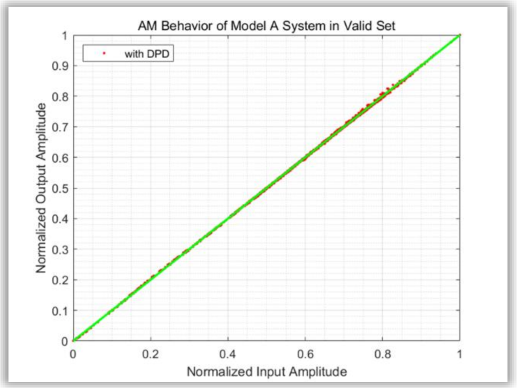
  Momentum Factor: 0.9

- **Results**

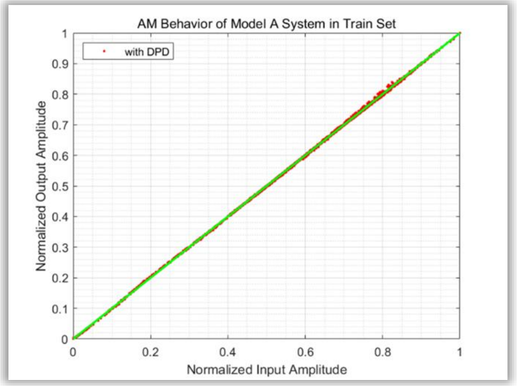
  NMSE on Fitting: **-52.952dB**

  NMSE on Train Set: **-49.211dB**

  NMSE on Valid Set: **-49.750dB**



Train Set



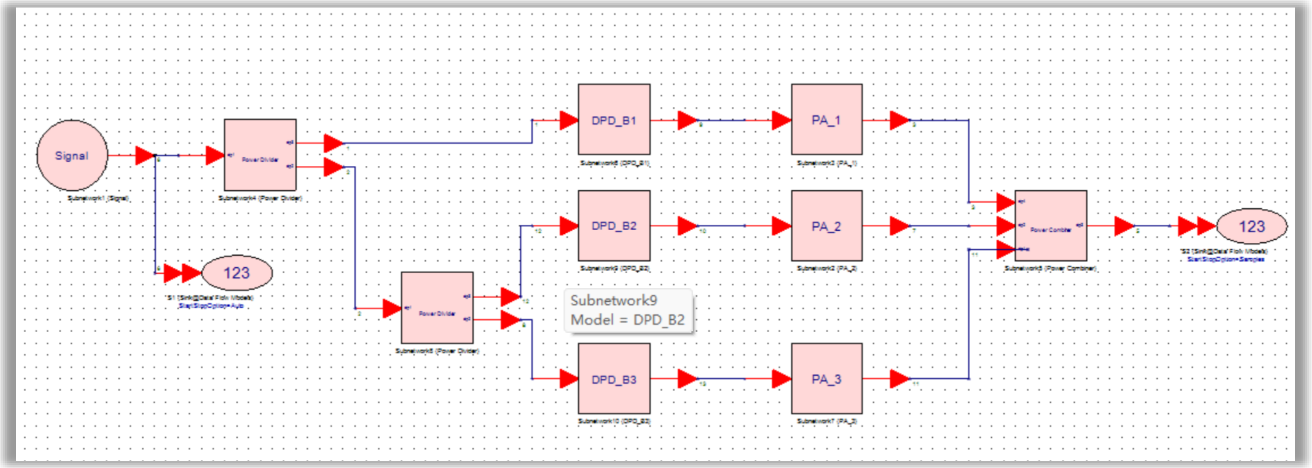
Valid Set

Performance comparison of different algorithms

Optimization Algorithm	Fitting NMSE(dB)	Train Set NMSE(dB)	Valid Set NMSE(dB)
adam	-46.003	-44.859	-45.179
sgdm	-42.238	-41.339	-41.572
rmsprop	-31.994	-33.357	-33.364
trainlm with complex network	-54.500	-45.662	-46.046
trainlm with simple network	-54.067	**-49.004**	**-49.381**
trainscg with simple network	-41.556	-41.540	-41.833
trainbr with simple network	-52.952	**-49.211**	**-49.750**

Best Performance: trainbr with simple network

PART 04. Composite Model



- 3-way Optimization Model

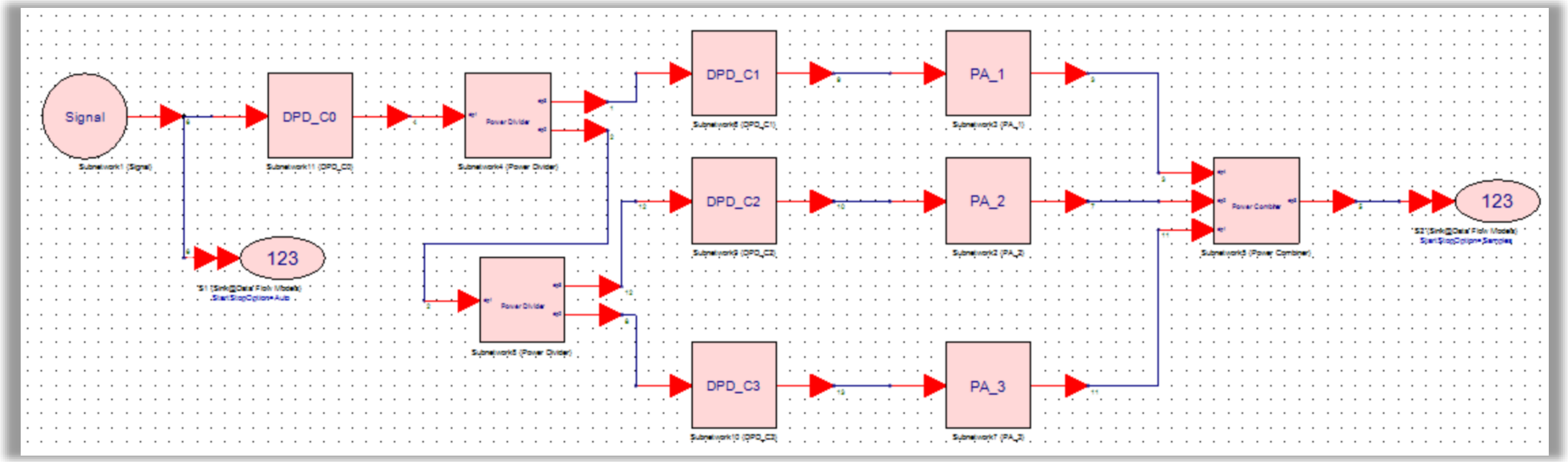
NMSE on Train Set: **-41.131dB**

NMSE on Valid Set: **-41.240dB**

- 3-way Optimization Plus Overall Optimization Model

NMSE on Train Set: **-49.155dB**

NMSE on Valid Set: **-49.693dB**



If the training effect of each branch is good, the effect will be even **WORSE**.

PART 05. Best Model: Multi-layer FCN

Best Model: Multi-layer FCN

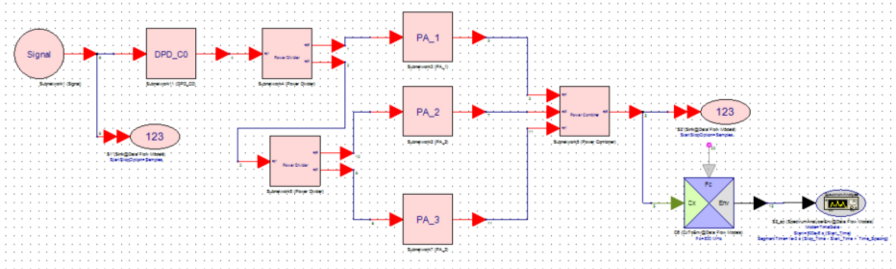
Single overall DPD

Network Structure: Multi-layer FCN (cascadeforwardnet ([36,24]))

Optimization Algorithm: trainbr

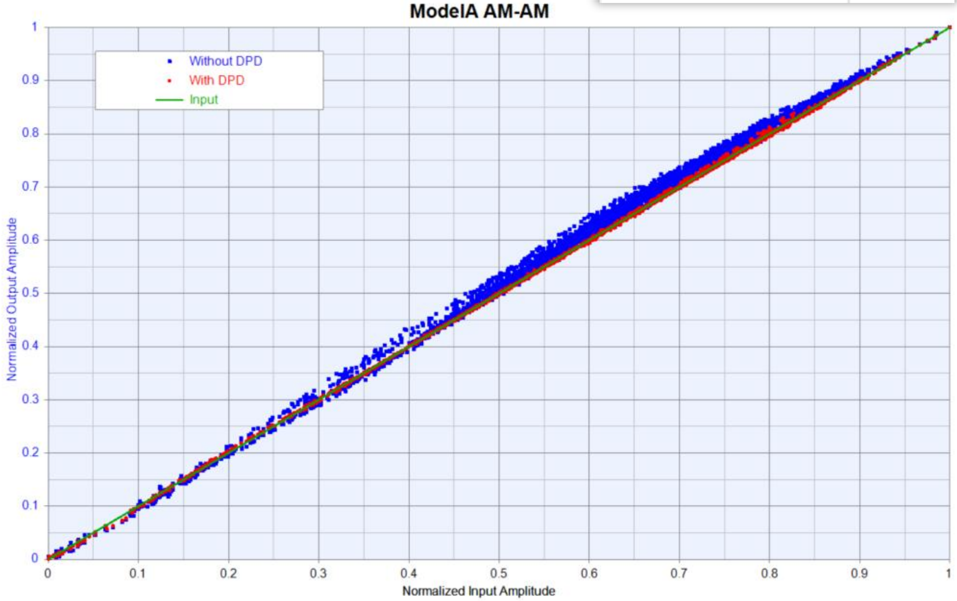
Data Segment: **2100 for train set, 1900 for valid set**

Regularization Coefficient: 0.0001



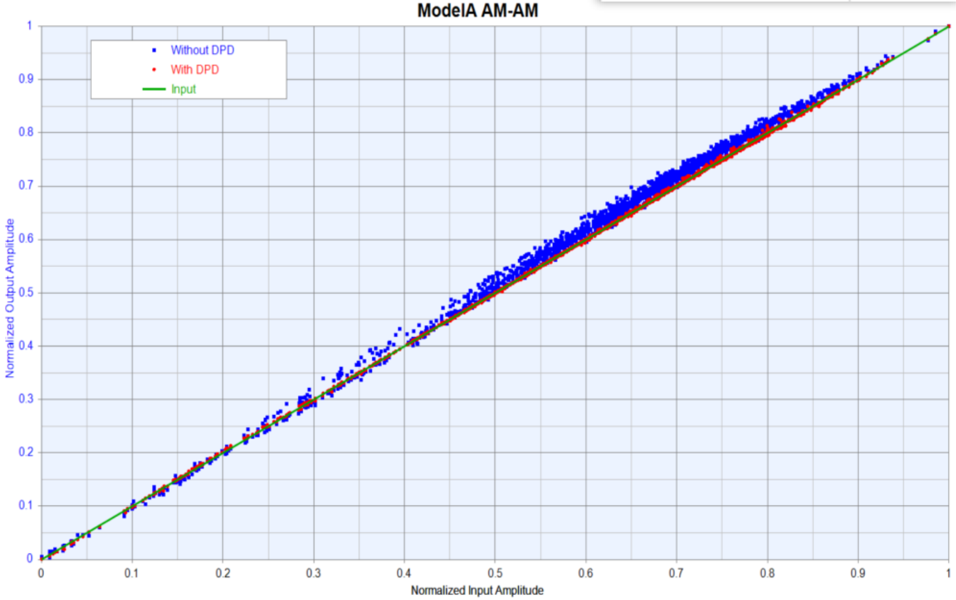
- Train Set

NMSE_Original	-26.743
NMSE_with_DPD	-49.667



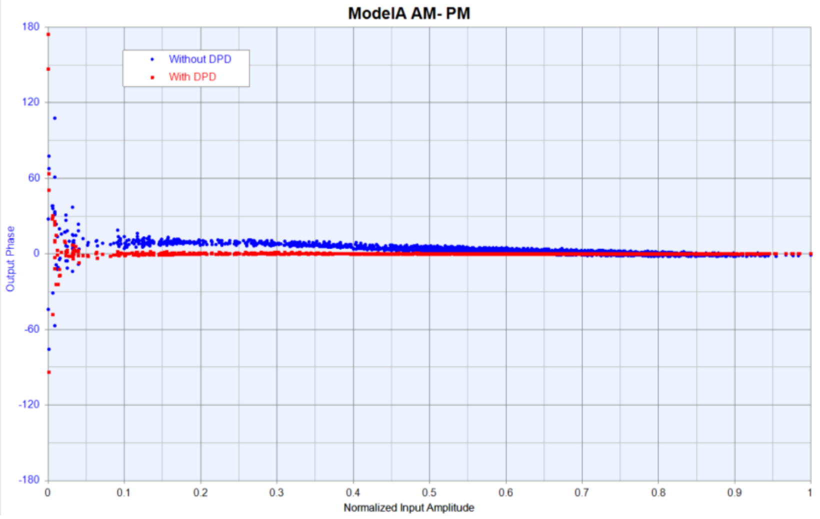
- Valid Set

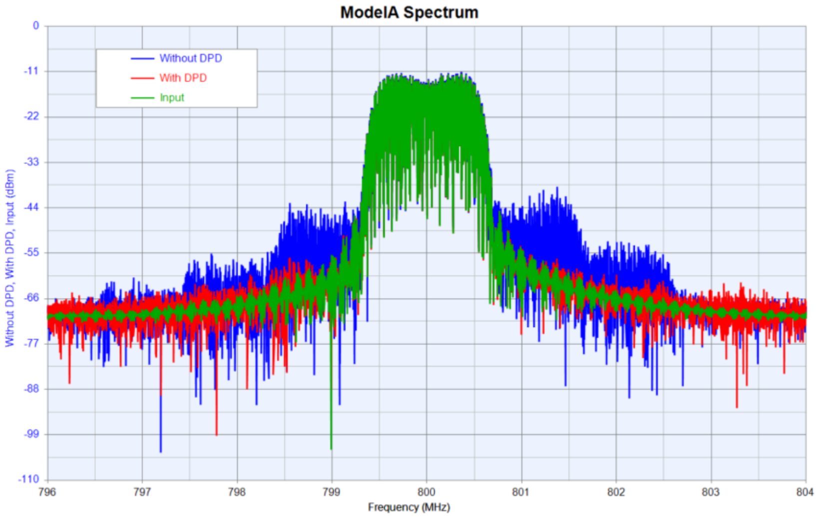
NMSE_Original	-26.794
NMSE_with_DPD	-49.9



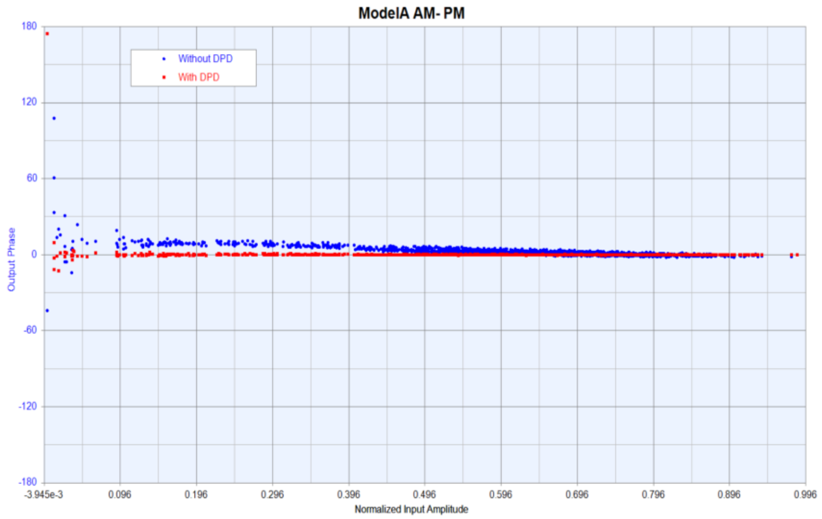
PART 05. Best Model: Multi-layer FCN

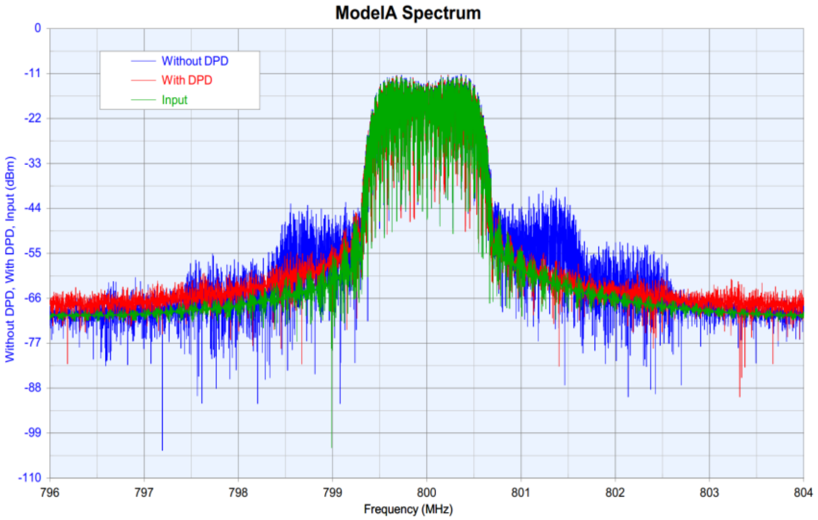
- Train Set





- Valid Set





PART 05. Best Model: GRU

Best Model: GRU

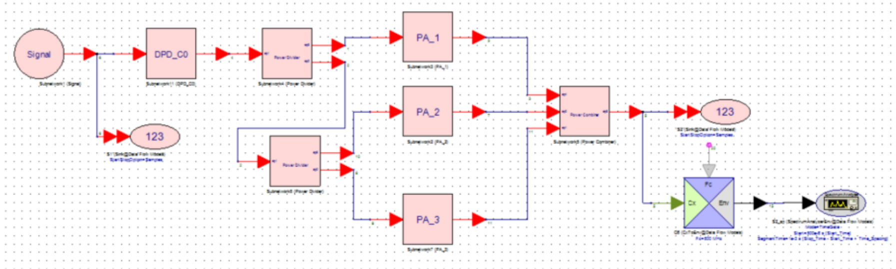
Single overall DPD

Network Structure: GRU

(Hidden Layer 1: 64 GRU hidden units; Activation Function 1: tanh)

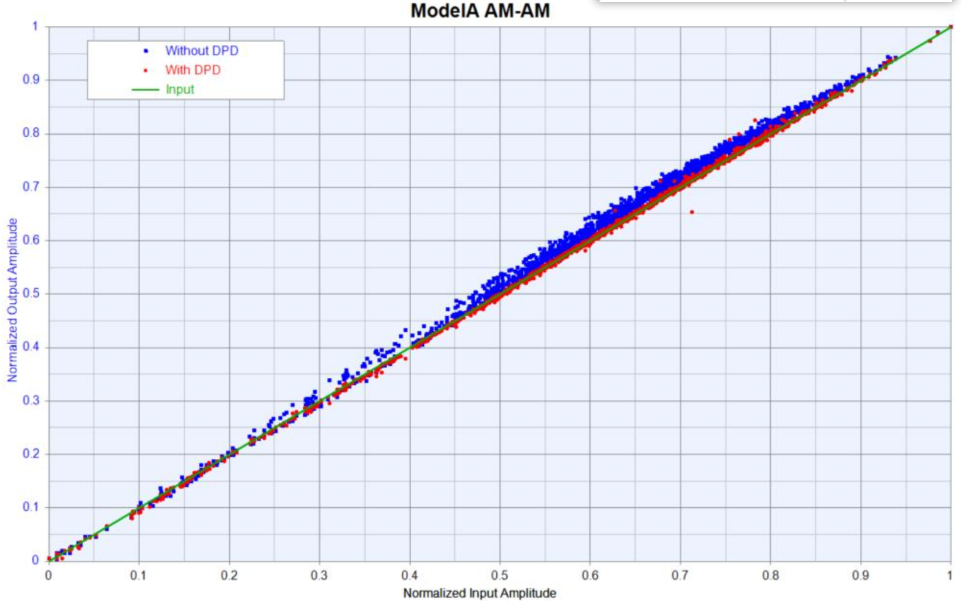
Optimization Algorithm: adam

Regularization Coefficient: 0.0001



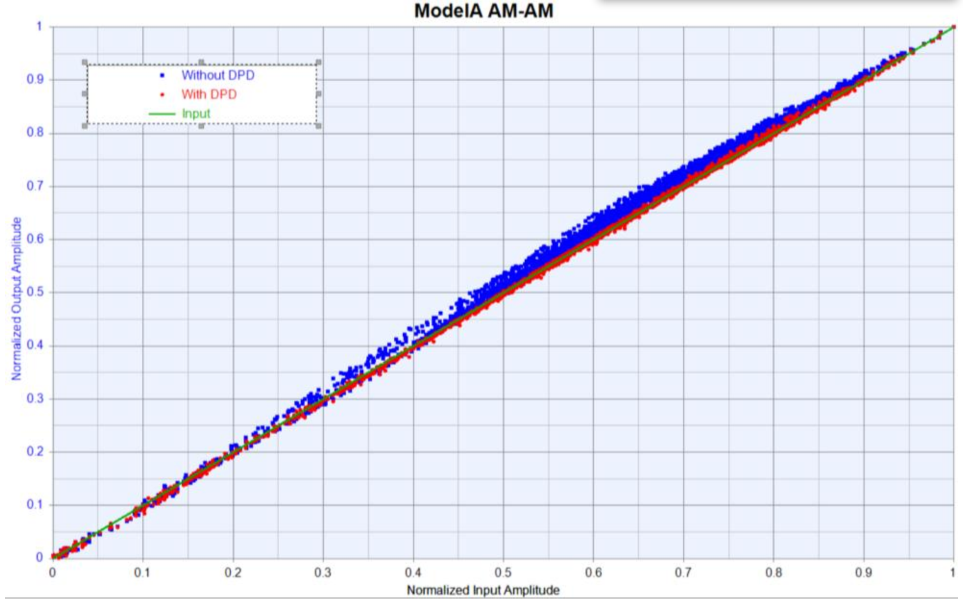
Train Set

NMSE_Original	-26.743
NMSE_with_DPD	-43.611



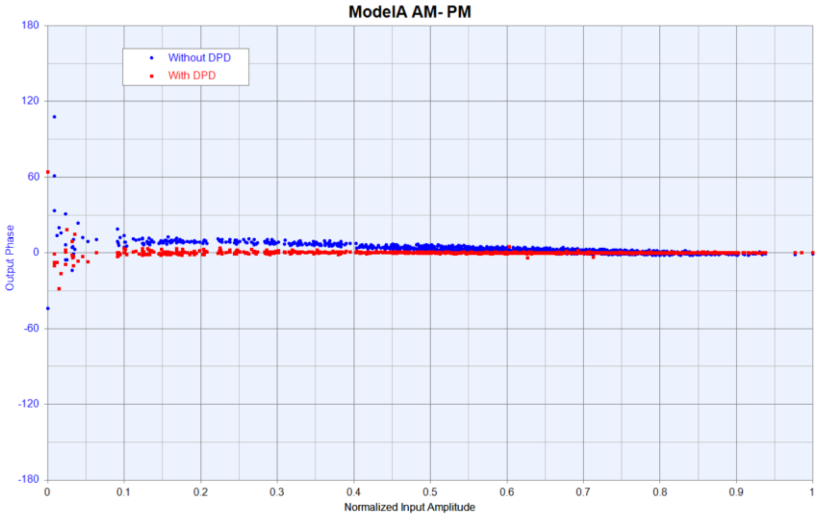
Valid Set

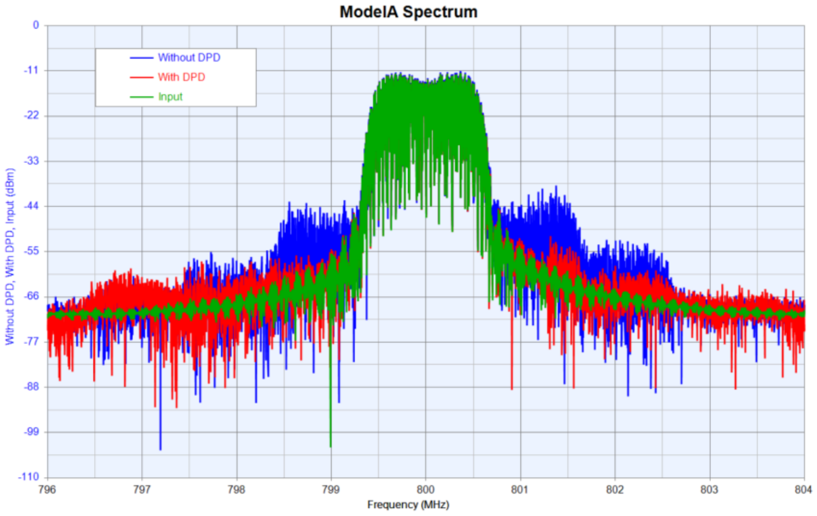
NMSE_Original	-26.794
NMSE_with_DPD	-43.878



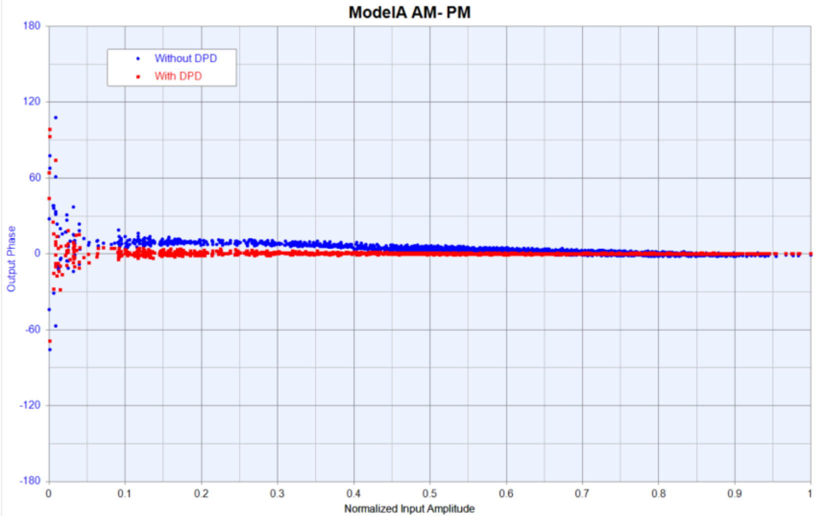
PART 05. Best Model: GRU

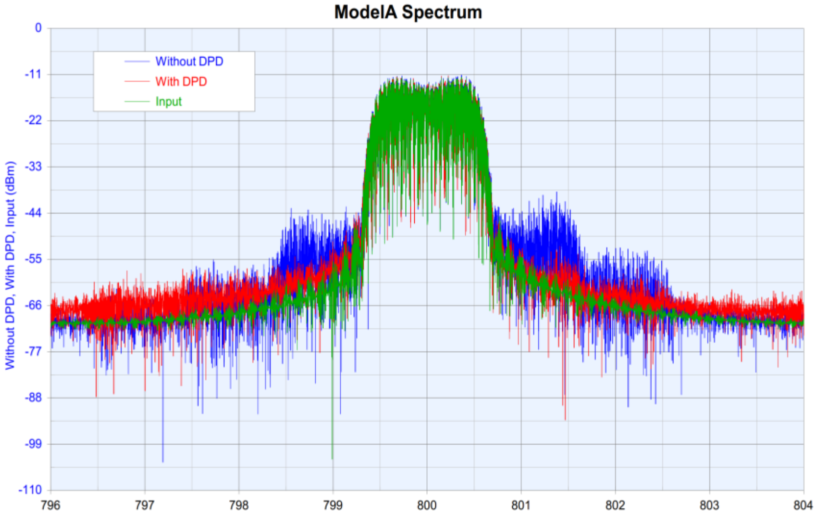
- Train Set





- Valid Set





PART 06. Conclusion

SOUTHEAST UNIVERSITY

01

This experiment focuses on digital predistortion (DPD) modeling of power amplifiers (PA). A total of **9 neural networks** were tried and the limitations of Volterra models were compared. We then explored the differences between **6 different optimization algorithms** and tried **other composite models**.

02

Among many networks, the Multi-layer FCN with **simple structure** performs best, while complex networks (such as Bi-LSTM) have high fitting accuracy, but due to **overfitting**, the actual application performance is not as good as that of simple models. Due to its limited nonlinear modeling capabilities, traditional Volterra models significantly lag behind neural network methods.

03

In terms of optimization algorithms, Levenberg-Marquardt algorithm, Bayesian regularization are suitable for processing **small-scale data**, so the effect of reducing the amount of data is better. The Adam algorithm is balanced in training efficiency and stability, and is suitable **for medium-scale data** scenarios.

SOUTHEAST UNIVERSITY

Some Difficulties

- Parameter adjustment is very complex, and small changes in network hyperparameters (such as memory depth, hidden layers) may significantly affect performance.
- Some algorithms (such as LM) rely on Jacobi matrices and cannot take advantage of GPU acceleration, limiting their application in large-scale scenarios.
- The adjustment direction of network parameters is very difficult to capture. When adjusting parameters, a table should be used to record parameters for reference.

Further Works

- Read more information and try more different methods (such as SVM).
- Improve the transfomer model and adjust the network structure and training parameters.

Division of labor

钟源(50%)：MATLAB程序搭建、网络训练、撰写报告、PPT优化

贾岩森(50%)：工作纲要记录、网络训练、撰写报告、PPT制作

THANKS

任课老师：余超

小组成员：
钟　源　贾岩森

信息科学与工程学院

2025年5月23日