IE582- HW 1

Sude Şahin-2024702045

1. Introduction:

Saçın, E. S., & Durgun, A. C. (2023, March) studied about designing and optimizing microwave devices requires computationally intensive electromagnetic (EM) analysis. Neural network-based machine learning methods offer an efficient alternative by creating surrogate models for EM performance. It is developed a spectral transposed convolutional neural network model for a microstrip patch antenna, using geometrical parameters and material properties to predict the S11 parameter.

Based on this research, this project aims to analyze the relationship between antenna design parameters and their electromagnetic performance using data-driven techniques. As high-frequency communication systems, such as those used in 5G technology, require precise antenna designs, understanding the influence of each design parameter becomes crucial. By using Principal Component Analysis (PCA), the dimensionality of the design space is reduced, allowing key parameters that significantly impact antenna behavior to be identified. Linear regression models are then applied to predict the S11 parameter at specific frequency points, offering a data-driven approach to optimize antenna performance. This study provides a framework for simplifying complex design spaces and efficiently predicting antenna performance in modern communication systems.

2. Data Interpretation:

In this study, three primary datasets are utilized: hw1_input.csv, hw1_real.csv, and hw1_img.csv. The hw1_input.csv file contains 385 rows, each representing a unique antenna design configuration with 11 design parameters, such as patch width, substrate height, and dielectric constants. These parameters define the physical and material characteristics of the antenna, which in turn influence its electromagnetic behavior.

The hw1_real.csv and hw1_img.csv files provide information on the real and imaginary components of the S11 parameter across 201 frequency points for each design. The S11 parameter, an essential performance metric, indicates the reflection coefficient, helping us understand how well an antenna radiates at specific frequencies. By analyzing both real and imaginary components, we calculated the S11 magnitude to determine the resonance frequency, where the antenna performance is optimized. The datasets were used to identify key design features with Principal Component Analysis (PCA) and to build regression models for predicting the antenna's performance at selected frequency points.

3. Research Questions:

3.1. Dimensionality Reduction with PCA:

The dimension and complexity of the data can be reduced using PCA. PCA generates a new set of directions (principal components) that explain the maximum variance in the data. As shown in Figure 3.1, the analysis revealed that 9 components are sufficient to explain 90% of the variance, meaning the data can be represented with 90% accuracy without utilizing all 11 original features. This approach allows the design space to be represented with fewer components, resulting in faster model execution while also highlighting the key parameters that affect antenna performance.

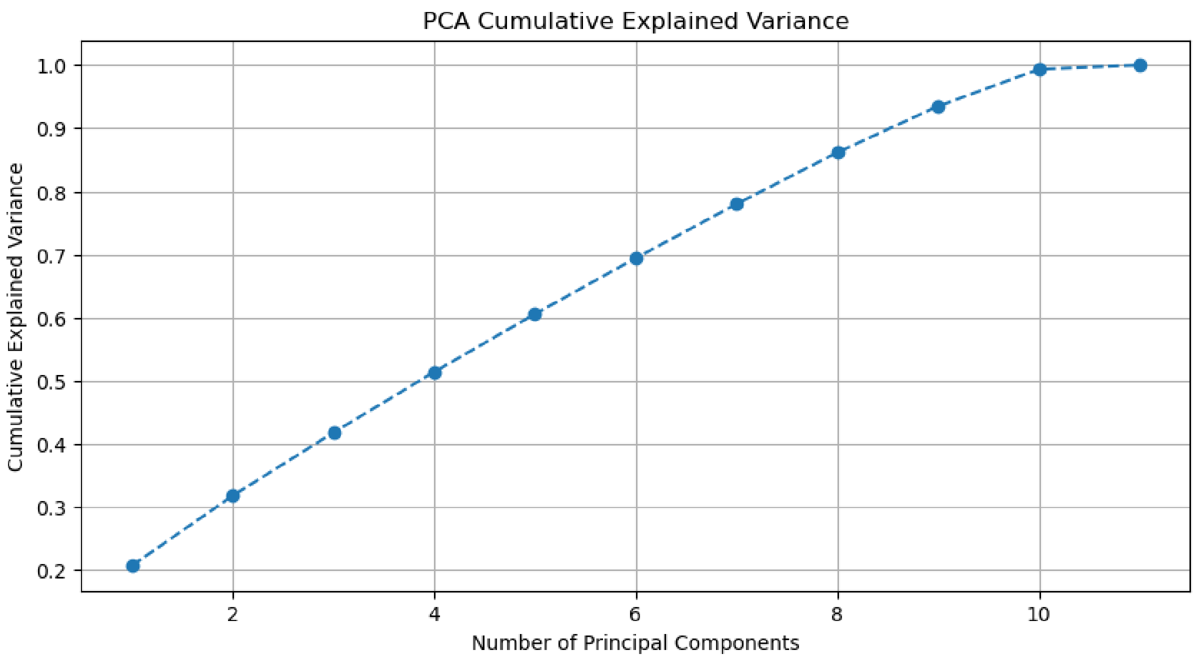

***Figure 3.1 :** PCA cumulative Explained Variance*

The principal components identified through PCA provide insights into which design parameters have the greatest influence on antenna performance. This information enables more informed decisions when evaluating the impact of specific geometric parameters on the S11 parameter during the design process.

For example, in the case of PC1, the three most significant parameters contributing to its explanatory power are the width of the patch, the height of the substrate, and the dielectric constant of the substrate. PC1 indicates that geometric dimensions such as the width and height of the substrate, along with material properties like the dielectric constant of the substrate, directly affect the S11 parameter. Optimizing these features can enhance the antenna's performance at specific frequencies.

Similarly, it is possible to analyze the key parameters for the other selected components individually.

Table 3.1: First Three Significant Parameters for each Principal Component

PC1 :	
width of patch	0.624178
height of substrate	0.624137
dielectric constant of substrate	0.445100
PC2 :	
c_probe	0.564753
c_antipad	0.559499
dielectric constant of solder resist layer	0.350944
PC3 :	
height of patch	0.620460
radius of the probe	0.548381
length of patch	0.369645
PC4 :	
height of solder resist layer	0.510652
radius of the probe	0.454397
height of patch	0.392974
PC5 :	
c_pad	0.580059
c_probe	0.487544
dielectric constant of solder resist layer	0.420314
PC6 :	
dielectric constant of solder resist layer	0.591708

c_pad 0.547579 height of solder resist layer 0.459841
PC7 : height of solder resist layer 0.647199 height of patch 0.490098 c_pad 0.429976
PC8 : radius of the probe 0.491478 dielectric constant of solder resist layer 0.460296 length of patch 0.454266
PC9 : c_probe 0.618694 c_antipad 0.548589 length of patch 0.406238

3.2. Regression Modeling for S11:

By targeting only critical frequencies, the regression task can be significantly simplified. Instead of running the model on unnecessary frequencies, focusing solely on important ones accelerates the analysis process and evaluates the model's performance more effectively. This approach allows for assessing antenna performance at critical frequencies while avoiding the high computational costs associated with multi-target regression.

To identify the resonance frequency, the average of the S11 magnitude values was calculated, and the frequency index with the minimum value was selected as the resonance frequency. Since the S11 magnitude is lowest at this frequency, it can be assumed to be the frequency at which the antenna performs best.

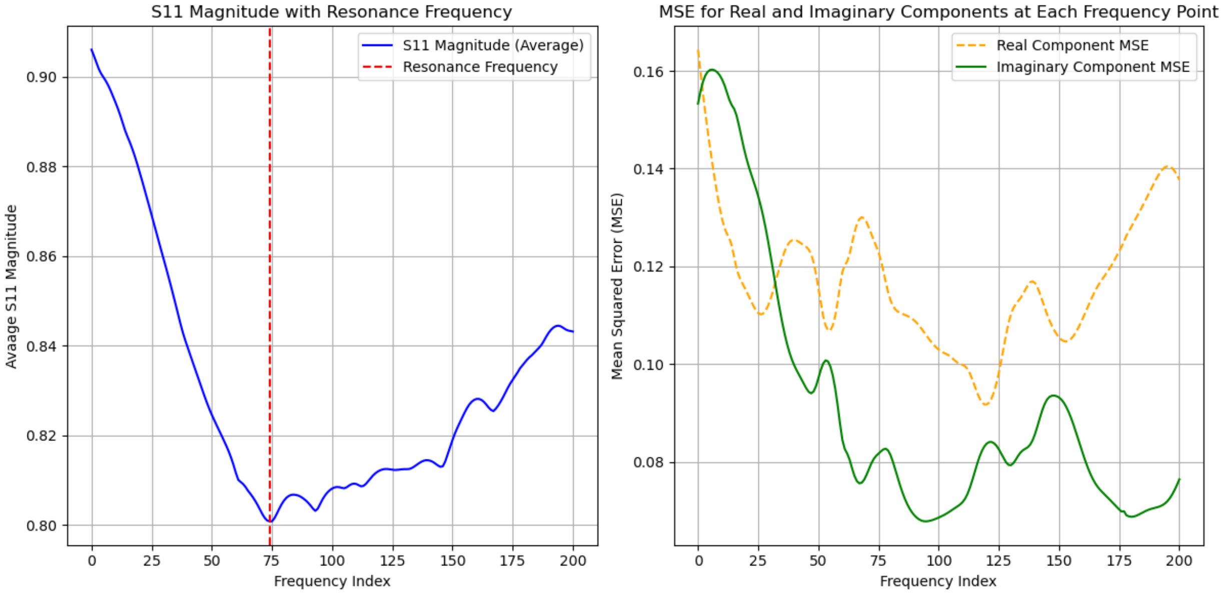

Figure 3.3

S11 Magnitude with Resonance Frequency and MSE for Real and Imaginary Components at Each Frequency Point

To demonstrate the efficiency of linear regression, MSE (Mean Squared Error) values were calculated. For the real component, the MSE value was found to be 0.124, indicating that the model, on average, incurs an error of 0.124 when predicting the real component. While this represents a relatively low error, it also highlights the model's limited accuracy in predicting the real component; a lower MSE value would signify a more accurate prediction.

For the imaginary component, the MSE value was 0.081. This lower error compared to the real component indicates that the model performs better in predicting the imaginary component.

Table 3.2: MSE for Real and Imaginary Components

MSE for real component	0.12418049184112362
MSE for imaginary component	0.08124095715019028

Figure 3.3 provides a comparison of MSE and frequency values for real and imaginary components, as well as the S11 magnitude with respect to frequency values.

The graph titled "MSE for Real and Imaginary Components at Each Frequency Point" shows that at lower frequency regions, the MSE values for both real and imaginary components are relatively low. This indicates that the linear regression model makes more accurate predictions at lower frequencies and suggests a stronger linear relationship between the antenna's geometric parameters and the S11 components at these frequencies.

Notably, the MSE values for the imaginary component drop to as low as 0.08 in the lower frequency range. A similar decline is observed for the real component, though the reduction in error is more pronounced for the imaginary component.

As the frequency increases, particularly in the mid and high-frequency regions, fluctuations and rises in MSE values are observed. This indicates that the linear regression model performs with higher error at higher frequencies, suggesting more complex, non-linear relationships between the geometric parameters and the high-frequency components of S11. Due to the weaker linear relationships at higher frequencies, linear models like regression become less effective in these regions.

3.3. Model Performance and Interpretability:

Applying PCA to the data reduces complexity and simplifies the analysis process. However, PCA does not directly predict antenna performance; instead, it helps identify which design parameters contribute most to the performance. Regression models, on the other hand, are used to predict the S11 parameter (real and imaginary components or magnitude) at a specific frequency. By using PCA-preprocessed data, predictions can be made with only the most significant components, resulting in a streamlined prediction process.

Based on the patterns observed, we can conclude that the relationship between the antenna's design parameters and its electromagnetic performance is complex and highly dependent on frequency. At lower frequencies, the linear regression model showed lower MSE values for both real and imaginary components, suggesting a stronger linear relationship between the design parameters and S11 behavior. This indicates that linear models can be effective for predicting antenna performance at these frequencies. However, at higher frequencies, MSE values increased, suggesting non-linear interactions between design parameters and performance metrics. In real-world applications, this insight implies that while linear models may be suitable for low-frequency antenna designs, more advanced models (e.g., neural networks or non-linear regression) are necessary for accurately modeling high-frequency performance.

One of the main challenges encountered was managing the high dimensionality of design parameters, which made it difficult to identify which features most influenced performance. We addressed this by applying **Principal Component Analysis (PCA)**, which successfully reduced the data to a manageable number of components while preserving most of the variance. Another challenge was capturing the non-linear relationships evident at higher frequencies. To overcome this, we analyzed performance across individual frequency points, which allowed us to identify the specific regions where linear models were insufficient.

4. Conclusion:

In this study, it is explored the relationship between antenna design parameters and electromagnetic performance using Principal Component Analysis (PCA) and linear regression. PCA helped us reduce the complexity of the design space by identifying key components that capture most of the variance, enabling more focused analysis. Linear regression models allowed us to predict the S11 parameter at selected frequency points, demonstrating effective performance at lower frequencies but revealing limitations at higher frequencies due to non-linear interactions.

Kitao, A. (2022) used Principal Component Analysis (PCA) in order to reduce the dimensionality of high-dimensional datasets, such as those generated by molecular simulations and experiments on proteins. Proteins' structural flexibility and conformational changes result in extensive variations in high-dimensional spaces whic PCA can help.

Mustapha, A., & Abdu, A. (2012) used Principal Component Analysis (PCA) and multiple linear regression to analyze surface water quality data, identifying pollution sources and their impact on water quality variation.

Similar methods can be applied in fields like telecommunications, in automotive and aerospace industries for sensor and radar system designs. By combining PCA with regression models, we can simplify complex design spaces and predict performance metrics in real-world systems where multi-dimensional design parameters impact outcomes, ultimately supporting more efficient and targeted optimization in engineering.

5. Python Codes:

```
# In[1]:


import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt


# In[2]:


#Enter the input data
input_data = pd.read_csv("hw1_input.csv")


# In[3]:


# Standardization of data
scaler = StandardScaler()
standardized_data = scaler.fit_transform(input_data)
```

```
# In[4]:


#Define PCA model and transform data
pca = PCA()
pca_components = pca.fit_transform(standardized_data)


# In[5]:


# Calculate explained variance and cumulative variance

explained_variance = pca.explained_variance_ratio_
cumulative_variance = explained_variance.cumsum()


# In[6]:


# Plot the variance ratios
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(cumulative_variance) + 1), cumulative_variance, marker='o',
linestyle='--')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('PCA Cumulative Explained Variance')
plt.grid(True)
plt.show()


# In[7]:


# To get a better result, take threshold value as 90 percent
target_variance = 0.90
n_components = next(i for i, cumulative_variance in enumerate(cumulative_variance) if
cumulative_variance >= target_variance) + 1
print(f"Number of components required to explain 90% of the variance:
{n_components}")
```

```
# In[8]:


# Applying PCA by selecting only the first 9 components
pca_reduced = PCA(n_components=9)
reduced_data = pca_reduced.fit_transform(standardized_data)


# In[9]:


# Converting PCA component loads to DataFrame
pca_components_df = pd.DataFrame(pca.components_,
columns=input_data.columns)

# Display the top 9 components and their loadings on each parameter
print("PCA Component Loadings:")
print(pca_components_df.head(9))


# In[10]:


# Find the parameters with the highest absolute loads for each component
for i in range(9):
   component = pca_components_df.iloc[i]
   max_loading = component.abs().nlargest(3)  # Select the 3 largest loads for each
component
   print(f"PC{i+1} :\n{max_loading}\n")


# In[11]:


import pandas as pd
import numpy as np

# Enter the real and imag components of S11 data
s11_real = pd.read_csv("hw1_real.csv")
```

```
s11_imag = pd.read_csv("hw1_img.csv")

# S11 magnitude calculation
s11_magnitude = np.sqrt(s11_real.values**2 + s11_imag.values**2)

# Finding the resonance frequency (frequency with minimum value on S11 magnitude)
min_s11_index = np.argmin(s11_magnitude.mean(axis=0))  #The minimum index is
found by taking the average
print("Index at resonance frequency:", min_s11_index)


# In[12]:


from sklearn.model_selection import train_test_split

# Use PCA data as input variable
X = reduced_data  # 9 components reduced by PCA

# Real and imaginary components at the resonance frequency
y_real = s11_real.values[:, min_s11_index]
y_imag = s11_imag.values[:, min_s11_index]

# Splitting into training and test sets
X_train, X_test, y_real_train, y_real_test = train_test_split(X, y_real, test_size=0.2,
random_state=42)
_, _, y_imag_train, y_imag_test = train_test_split(X, y_imag, test_size=0.2,
random_state=42)


# In[13]:


from sklearn.linear_model import LinearRegression

# Creating and training the model for the real component
model_real = LinearRegression()
model_real.fit(X_train, y_real_train)

# Creating and training a model for the imaginary component
model_imag = LinearRegression()
```

```
model_imag.fit(X_train, y_imag_train)


# In[14]:


from sklearn.metrics import mean_squared_error

# Prediction and MSE calculation for the real component
y_real_pred = model_real.predict(X_test)
mse_real = mean_squared_error(y_real_test, y_real_pred)
print("MSE for real component:", mse_real)

# Prediction and MSE calculation for imaginary component
y_imag_pred = model_imag.predict(X_test)
mse_imag = mean_squared_error(y_imag_test, y_imag_pred)
print("MSE for imaginary component:", mse_imag)


# In[15]:


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt


mse_real_list = []
mse_imag_list = []

for freq in range(s11_real.shape[1]):
  # Real and imaginary components in frequency are taken as targets
  y_real = s11_real.values[:, freq]
  y_imag = s11_imag.values[:, freq]

  # Splitting into training and test set
  X_train, X_test, y_real_train, y_real_test = train_test_split(reduced_data, y_real,
test_size=0.2, random_state=42)
  _, _, y_imag_train, y_imag_test = train_test_split(reduced_data, y_imag, test_size=0.2,
random_state=42)
```

```
    # Creating and training a linear regression model
    model_real = LinearRegression()
    model_real.fit(X_train, y_real_train)

    model_imag = LinearRegression()
    model_imag.fit(X_train, y_imag_train)

    # Prediction and MSE calculation
    y_real_pred = model_real.predict(X_test)
    y_imag_pred = model_imag.predict(X_test)

    mse_real = mean_squared_error(y_real_test, y_real_pred)
    mse_imag = mean_squared_error(y_imag_test, y_imag_pred)


    mse_real_list.append(mse_real)
    mse_imag_list.append(mse_imag)

# Visualizing results
plt.figure(figsize=(12, 6))
plt.plot(range(len(mse_real_list)), mse_real_list, label='Real Component MSE',
linestyle='--')
plt.plot(range(len(mse_imag_list)), mse_imag_list, label='Imaginary Component MSE',
linestyle='-')
plt.xlabel('Frequency Index')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('MSE for Real and Imaginary Components at Each Frequency Point')
plt.legend()
plt.grid()
plt.show()


# In[16]:


import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

# S11 magnitude graph
```

```
plt.subplot(1, 2, 1)
plt.plot(s11_magnitude.mean(axis=0), label='S11 Magnitude (Average)', color='blue')
plt.axvline(x=min_s11_index, color='red', linestyle='--', label='Resonance Frequency')
plt.xlabel('Frequency Index')
plt.ylabel('Avaage S11 Magnitude')
plt.title('S11 Magnitude with Resonance Frequency')
plt.legend()
plt.grid()

# MSE values graph
plt.subplot(1, 2, 2)
plt.plot(range(len(mse_real_list)), mse_real_list, label='Real Component MSE',
linestyle='--', color='orange')
plt.plot(range(len(mse_imag_list)), mse_imag_list, label='Imaginary Component MSE',
linestyle='-', color='green')
plt.xlabel('Frequency Index')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('MSE for Real and Imaginary Components at Each Frequency Point')
plt.legend()
plt.grid()

plt.tight_layout()
plt.show()
```

6. References:

Mustapha, A., & Abdu, A. (2012). Application of principal component analysis & multiple regression models in surface water quality assessment. *Journal of environment and earth science*, *2*(2), 16-23.

Kitao, A. (2022). Principal component analysis and related methods for investigating the dynamics of biological macromolecules. *J*, *5*(2), 298-317.

OpenAI. (2024). *ChatGPT (November 2024 Version)*. Retrieved from https://openai.com

Saçın, E. S., & Durgun, A. C. (2023, March). Neural network modeling of antennas on package for 5G applications. In *2023 17th European Conference on Antennas and Propagation (EuCAP)* (pp. 1-5). IEEE.