T.C.
IŞIK UNIVERSTIY
SCHOOL OF GRADUATE STUDIES

MASTER THESIS
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELECTRONICS ENGINEERING PROGRAM

Feras ABOU ALLIL

ELECTRICAL CIRCUIT DESIGN BASED ON NEURAL NETWORKS

SUPERVISOR
Assoc. Prof. Ramazan KÖPRÜ

İSTANBUL, January 2026

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

MASTER THESIS
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELECTRONICS ENGINEERING PROGRAM

Feras ABOU ALLIL

(23ELEC5002)

ELECTRICAL CIRCUIT DESIGN BASED ON NEURAL NETWORKS

SUPERVISOR

Assoc. Prof. Ramazan KÖPRÜ

İSTANBUL, January 2026

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

MASTER'S THESIS
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELECTRONICS ENGINEERING PROGRAM

FERAS ABOU ALLIL

(23ELEC5002)

ELECTRICAL CIRCUIT DESIGN BASED ON NEURAL NETWORKS

Date: 23/01/2026

Thesis Supervisor: Assoc. Prof. Ramazan Köprü / Işık University

Jury Members:

Prof. Ümit Güz / Işık University

Assoc. Prof. Lida Kouhalvandi / Doğuş University

İSTANBUL, January 2026

* "6689 Sayılı Kişisel Verilerin Korunması Kanunu Hükümlerine Göre Çevrimiçi Yayın Dosyasında Bulunan Kişisel Veriler Ve Islak İmzalar Silinmiştir."

ÖZET

YAPAY SİNİR AĞLARINA DAYALI ELEKTRİK DEVRE TASARIMI

Yapay Sinir Ağları (YSA), güçlü öğrenme ve genelleme yetenekleri sayesinde hızlı ve doğru performans tahmini gerektiren uygulamalarda önemli ölçüde ilgi görmektedir. Bu tezde, hem pasif hem de aktif filtre topolojilerini içeren analog elektronik devrelerin tasarım ve analizinde sinir ağlarının kullanımına yönelik kapsamlı bir çalışma sunulmaktadır. Ölçüm sinyallerindeki istenmeyen gürültüyü azaltmak ve frekans tepki özelliklerinden devre bileşen değerlerini doğru bir şekilde çıkarmak amacıyla ileri beslemeli bir sinir ağı mimarisi kullanılmaktadır. Her devre tipi için, devre parametreleri ile karşılık gelen genlik tepki özellikleri arasındaki ilişkiyi öğrenmek üzere özel bir sinir ağı eğitilmiştir. Çalışma, pasif elemanların yanı sıra işlem yükselteçleri ve işlemsel transkonduktans yükselteçleri (OTA'lar) gibi aktif aygıtlar kullanılarak gerçeklenen alçak geçiren ve bant geçiren filtreleri içermektedir. Bu kapsamda iki eğitim yöntemi tanıtılıp değerlendirilmiştir: Eleman Yayma Eğitimi (EST) ve Eleman Rastgeleleştirme Eğitimi (ERT). Bu yaklaşımlar, veri kümesi çeşitliliğini artırmakta ve sinir ağının daha geniş bir devre davranışı aralığında genelleme yeteneğini geliştirmekte, böylece daha güvenilir ve sağlam tahminler elde edilmesini sağlamaktadır. Sunulan yaklaşım, sinir ağlarının klasik analog devre tasarımıyla bütünleştirilmesine yönelik potansiyeli ortaya koymakta; performans, avantajlar ve sınırlamalar konusunda önemli bulgular sunmaktadır. Tüm analiz ve simülasyonlar MATLAB ortamında gerçekleştirilmiş ve doğrulanmıştır. Ayrıca, önerilen yöntemler farklı frekans aralıkları ve eleman toleransları altında test edilmiştir.

Anahtar Kelimeler: Yapay Sinir Ağları, İleri Beslemeli Sinir Ağları Mimarisi, GUI, MATLAB, Proteus

ABSTRACT

ELECTRICAL CIRCUIT DESIGN BASED ON NEURAL NETWORKS

Artificial Neural Networks (ANNs) have gained significant attention due to their fast and accurate performance estimation capabilities, particularly in applications requiring strong learning and generalization. In this thesis, a comprehensive study is presented on the use of neural networks for the design and analysis of analog electronic circuits, focusing on both passive and active filter topologies. A feedforward neural network architecture is employed to reduce unwanted noise in measurement signals and to accurately infer component values from frequency response characteristics. For each circuit type, a dedicated neural network is trained to learn the relationship between circuit parameters and their corresponding magnitude responses. The study includes a variety of analog filters—such as low-pass and band-pass filters—implemented using passive elements as well as active devices including operational amplifiers and operational transconductance amplifiers (OTAs). Two training methodologies are introduced and evaluated: Element Spreading Training (EST) and Element Randomization Training (ERT). These approaches enhance dataset diversity and improve the neural network's ability to generalize across a wider range of circuit behaviors, resulting in more reliable and robust predictions. The overall framework demonstrates the potential of integrating neural networks into classical analog circuit design, offering insights into performance, advantages, and limitations. All analyses and simulations are conducted and validated using MATLAB. The proposed methods have been tested under different frequency ranges and component tolerances.

Keywords: Artificial Neural Networks, Feedforward Neural Networks Architecture, GUI, MATLAB, Proteus

ACKNOWLEDGEMENT

First and foremost, I want to sincerely thank Assoc. Prof. Ramazan Köprü, my supervisor, for his invaluable guidance, support, and thoughtful feedback throughout this project. His advice and encouragement were not only crucial during the writing of this thesis, but also instrumental in the circuit designs, implementation, and overall progress of the research work. His expertise and motivation greatly shaped the direction and success of this study.

Additionally, I would like to express my gratitude to the faculty and staff at Isik University's Electrical and Electronics Engineering Department for providing the resources and a helpful academic environment that allowed me to finish this work.

I would especially like to thank my family and friends for their constant support, tolerance, and encouragement during my academic career. Their confidence in me has consistently inspired me.

Lastly, I would like to express my gratitude to everyone who helped this thesis succeed, whether directly or indirectly.

Feras ABOU ALLIL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS LIST

AC: Alternative Current

AI: Artificial Intelligence

ANN: Artificial Neural Networks

BPF: Bandpass Filter

BSIM: Berkeley Short-channel IGFET Model

CNN: Convolutional Neural Networks

DC: Direct Current

DR: Dynamic Range

ERT: Element Randomization Training

EST: Element Spreading Training

FIR: Finite Impulse Response

FNN: Feedforward Neural Network

GPS: Global Positioning System

GUI: Graphical User Interface

IIR: Infinite Impulse Response

KCL: Kirchhoff's Current Law

LPF: Lowpass Filter

ML: Machine Learning

MLP: Multi-Layer Perceptron

MOSFET: Metal-Oxide-Semiconductor Field-effect Transistor

MPC: Model Predictive Control

NN: Neural Networks

OTA: Operational Transconductance Amplifier

PSO: Particle Swarm Optimization

RBF: Radial Basis Function

RNN: Recurrent Neural Networks

SLP: Single Layer Perceptron

THD: Total Harmonic Distortion

UI: User Interface

UKF: Unscented Kalman Filters

WNN: Wavelet Neural Networks

CHAPTER 1

1. INTRODUCTION

1.1. BACKGROUND OF NEURAL NETWORKS

Neural networks are computational models inspired by the brain, utilized in machine learning for pattern recognition (speech and image recognition) and decision-making. Also, it consists of a sequence of algorithms aiming to recognize the relationships in data through strategical processes similar to the way the human brain works to make predictions based on the given data (Rosebrock, 2021).

Neural Networks can be observed in real life analogies. For instance, How do children learn to differentiate between two different animals? How does someone learn to navigate a new city without the use of GPS? The answer depends on how our brain works, where the biological nervous systems (nerve cells) are connected by large numbers of neural networks. The term "neural" serves as an adjective connected to "neuron", whereas "network" refers to a framework that resembles a graph. Hence, an “Artificial Neural Network” is a computational system designed to replicate (or at least draw inspiration from) the neural connections found in our nervous system (Rosebrock, 2021).

The brain contains approximately 10 billion neurons, which improve the ability of training data to learn to rely on their accuracy over time. These neurons collect inputs, analyze them, and generate an output. They are structured in specific layers: an input layer that accepts the data, multiple hidden layers that manage this data, and an output layer that delivers the ultimate decision or prediction (Awan, 2023). Every node executes a basic calculation, and each node transmits a signal from one node to another, marked by a **weight** that shows how much the signal is enhanced or reduced. Certain connections have significant positive and negative weights that boost the signal, suggesting that the signal is

crucial for making a classification, while others are less critical. Weights represent the features of the given input. The greater the weight value, the more significant the features it represents.

As neural networks train, they continually adjust their weights to reduce errors and improve accuracy. Using backpropagation, the network updates these weights layer by layer, gradually learning complex patterns that traditional models may miss. This adaptability allows neural networks to generalize from training data and perform well on new, unseen inputs, making them effective in tasks such as image classification, language processing, and signal analysis.

1.2. THE PURPOSE OF THE RESEARCH

The aim of this research is to investigate and highlight the capabilities of Neural Networks (NNs) in the automatic design and evaluation of analog electronic circuits, specifically targeting filter designs. Conventional analog circuit design necessitates extensive specialist knowledge, repetitive adjustments, and labor-intensive simulations. This presents a smarter, more efficient, and scalable method by utilizing neural networks techniques such as Element Spreading Training (EST) and Element Randomization Training (ERT) to estimate circuit parameters according to the required performance standards. The research focuses on creating both passive and active bandpass filters, including RC and LC types, transconductance-capacitor (Gm-C) bandpass filters, as well as circuits based on MOSFET.

Each type of circuit is trained using neural networks to understand the interconnections between component values (such as resistors, capacitors, transconductance, and inductors) and their frequency response characteristics. By using EST, the model thoroughly investigates a broad spectrum of component values, ensuring that the circuit maintains its performance across different conditions. On the other hand, ERT introduces intentional randomness in component choices, enabling the neural network to uncover unconventional yet effective configurations for bandpass filters (BPF).

CHAPTER 2

2. GENERAL INFORMATION

2.1 NEURAL NETWORKS

A perceptron is an artificial computational unit within an intelligent system that is capable of performing decision-making tasks. It is inspired by the structure and functionality of biological neurons in the human brain, which consists of billions of interconnected neurons forming complex and deeply layered networks (Hathidara and Pandey, 2025).

A perceptron is a foundational model in the field of neural networks, serving as the simplest form of a binary classifier. It simulates a single neuron, mapping a set of input features to a single output or more. Its mathematical operation attempts to mimic how a biological neuron processes information. The core of its operation involves calculating a weighted sum of multiple inputs, which is then passed through an activation function. This function determines the output, typically firing only if the weighted sum exceeds a certain threshold or bias (Popescu et al., 2009). The fundamental components of a perceptron are:

- Inputs ($x_1, x_2, \ldots \ldots, x_i$): The data features
- Weights ($w_1, w_2, \ldots \ldots, w_i$): Numerical values assigned to each input, representing their importance.
- Bias (b): A constant value added to the weighted sum, allowing the activation function to be shifted.
- Activation function (f): A nonlinear function that processes the weighted sum and determines the final output.

NNs have many architecture classes which fit specific types of problems (Wu, 2014).

Single Layer Perceptron (SLP): This is the simplest form of neural network, mainly used for solving linearly separable problems such as basic

pattern classification. However, it is limited in capability since it cannot handle complex, non-linear decision boundaries (See: Figure 2.1). It's a network consisting of inputs, and one output with no hidden layers used (David, 2021).

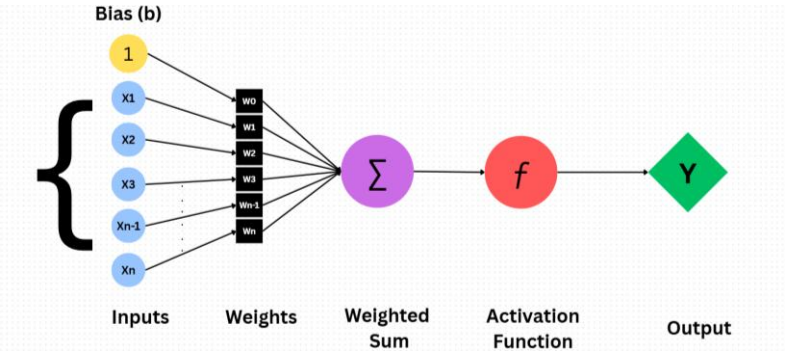


Figure 2.1 Single Layer Perceptron Diagram

Multi-Layer Perceptron (MLP): A widely recognized neural network architecture which extends the structure of the Rosenblatt perceptron (Foote, 2021). MLPs possess universal approximation capability, meaning they can represent any continuous, bounded, differentiable, nonlinear function within a compact input domain to any desired level of accuracy. This capability arises from the additive combination of their fundamental building blocks, which, in the case of MLPs, are ridge functions. Nevertheless, the theorem providing this guarantee does not indicate how many neurons are necessary, nor does it offer a method for determining the optimal set of weight values within the network.

To demonstrate this, assume we have 3 inputs, 2 hidden layers and two outputs. (See: Figure 2.2). This illustrates the hidden layers of each of its neurons working by giving inputs to produce an output, or two (Macukow, 2016).

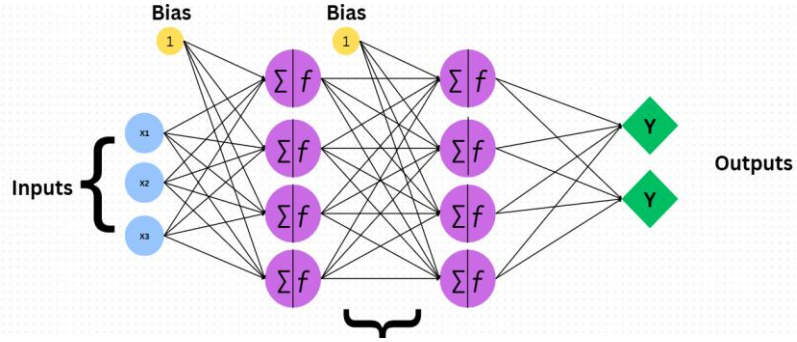


Figure 2.2 Multi-Layer Perceptron Diagram

They rely on Feedforward propagation for training, and backpropagation which adjusts the weights to minimize error. In forward propagation the data flows from the input layer to the output layer, passing through any hidden layers. Each neuron in the hidden layers processes the input as follows:

The Weighted Sum

$$z = \sum_i (w_i \times x_i) + b \qquad (\mathbf{2.1})$$

where z is the weighted sum of the inputs also given as,

$$z = w_1 x_1 + w_2 x_2 + \ldots w_i x_i + b \qquad (\mathbf{2.2})$$

Activation Function (f): The weighted sum z is passed through an activation function to determine non-linearity as in equation (**2.3**). The neuron can adjust the activation function along the stimulus axis, which helps it identify patterns in the data that may not be directly related to origin. It can be a step function, Sigmoid, Rectified Linear Unit (ReLU) or Hyperbolic Tangent (tanh).

This illustrates the operation of every neuron in the covert layer, with additional information provided. The equation (**2.3**) represents the output of the network. The activation function equations (**2.4**), (**2.5**), (**2.6**) and (**2.7**) will be clarified in equations to show their conditions (GeeksforGeeks, 2025).

$$y = f(z) \qquad (\mathbf{2.3})$$

$$f(z) = \begin{cases} 1 & , z > 0 \\ 0 & , z < 0 \end{cases} \quad (\mathbf{2.4})$$

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\sum w_i x_i + b)}} = \frac{1}{1+e^{-(w_1 x_1 + w_2 x_2 + \cdots w_n x_n + b)}} \quad (\mathbf{2.5})$$

$$f(z) = \max(0,z) \quad (\mathbf{2.6})$$

$$\tanh(z) = \frac{2}{1+e^{-2z}} - 1 \quad (\mathbf{2.7})$$

where f(z): Step function, σ(z): Sigmoid function, f(z): ReLU function and tanh(z): Hyperbolic tangent function.

Note: The activation function type is chosen depending on the problem.

Loss Function: After the network produces an output, the next step is to compute the loss using a loss function. In supervised learning as shown in equation (**2.8**), this process compares the predicted $\hat{y}_i$ result with the real answer y_i (Berzal, 2025).

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (\mathbf{2.8})$$

Mean Square Error (MSE): A statistical metric that measures the average of the squares of the differences between predicted $\hat{y}_i$ and actual values y_i. It quantifies how far an estimator is from the true value, with lower MSE indicating a more accurate model or estimator shown in equation (**2.9**). The process involves squaring errors, which eliminates negative values and gives more weight to larger errors, making MSE sensitive to outliers (See: Figure 2.3) (GeeksforGeeks, 2025).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (\mathbf{2.9})$$

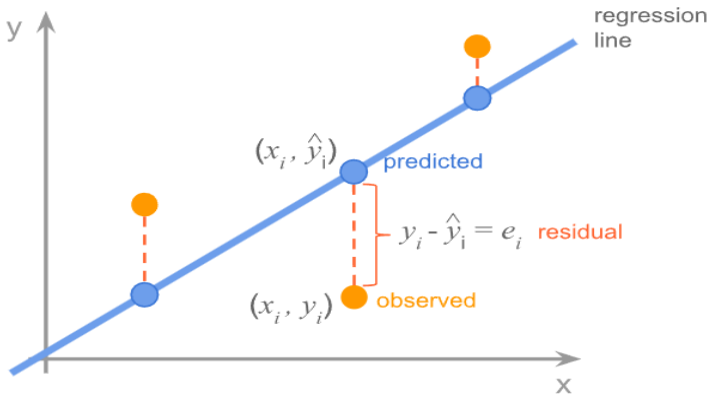


Figure 2.3 MSE Regression Line

Radial Basis Function (RBF): A radial basis function network is a type of artificial neural network that employs radial basis functions for its activation functions. The network's output is a linear combination of these radial basis functions based on the inputs and the parameters of the neurons. RBFs have not been incorporated into modern deep learning frameworks or computer vision systems based on conventional convolutional neural networks (CNNs), primarily because they are not well-suited to contemporary architectural designs (Amirian et al., 2022).

Recurrent Neural Network (RNN): Recurrent Neural Networks (RNNs) represent a class of neural network architectures designed to identify patterns within sequential data. Such sequences may include handwriting, genetic information, textual data, or numerical time-series commonly encountered in industrial applications. The key distinction between RNNs and Feedforward Neural Networks—also known as Multi-Layer Perceptrons (MLPs) —lies in the way information flows through the network. Unlike MLPs, RNNs incorporate feedback connections that allow them to retain information from previous inputs, giving them a form of short-term memory. This capability makes RNNs particularly effective for tasks where temporal dependencies and context play a critical role (Schmidt, 2019). This class contains memory, which means that the hidden layers neurons have self-connections. Furthermore, the hidden layers receive an activation from the lower layers and the ones before (See: Figure 2.4).

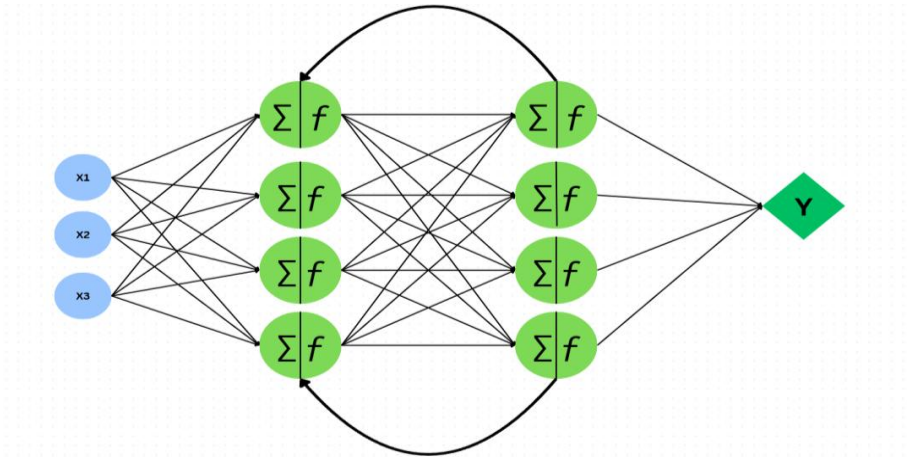


Figure 2.4 Recurrent Neural Networks

Convolution Neural Network (CNN): It's a deep learning technique particularly tailored for handling image data. Convolutional neural networks are applied in the fields of image recognition and processing. The author (Taye, 2023) concludes Convolutional neural networks enable the automatic learning of a series of features that can be employed for classification, rather than having to manually extract features for segment detection in images. The neural networks are structured in three layers:

- **Input layers:** Receive the input data
- **Hidden layers:** Processes the data through neurons
- **Output layers:** Produces the final output

Both classes utilize feedforward and backpropagation mechanisms. During the feedforward stage, information moves from the input layer through the hidden layers to the output layer, producing a prediction before any evaluation of accuracy occurs. Every neuron in one layer is linked to all neurons in the subsequent layer, enabling complete transmission of information. The backpropagation phase follows, in which the network compares its predicted output with the actual target, computes the resulting error, and updates the weights by sending this error backward through the layers. Through this iterative process, the network progressively refines its predictions and enhances overall learning performance (See: Figure 2.5).

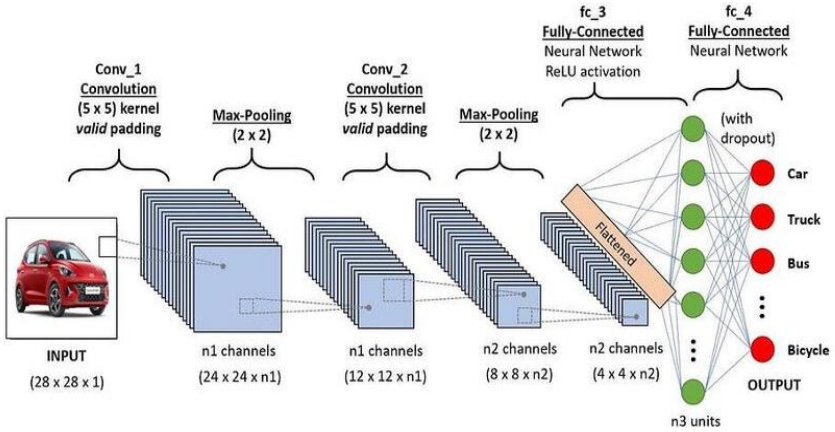


Figure 2.5 Convolutional Neural Networks

According to O'Shea and Nash, the fundamental operation of the CNN example can be divided into four main components:

- Similar to other neural network models, the input layer stores the raw pixel intensities of the image.
- The convolutional layer computes neuron activations by applying weight filters to small, localized sections of the input, effectively calculating a dot product to extract spatial features. Following this, a rectified linear unit (ReLU) applies an elementwise nonlinearity—such as a sigmoid-like activation—to introduce nonlinear behavior into the network.
- Next, the pooling layer performs spatial down sampling, reducing the dimensionality of the feature maps and decreasing the number of trainable parameters.
- Finally, the fully connected layers operate similarly to those in traditional ANNs, transforming the extracted features into class probabilities or output scores. ReLU activations may also be inserted between these layers to enhance the network's learning efficiency and overall performance.

Analog circuits are essential in contemporary electronic systems, particularly in applications that require signal amplification, frequency selection,

modification, and modulation. A crucial element in analog signal processing is the filter, which serves the purpose of selectively allowing or rejecting specific frequency components of an incoming signal. Filters are common in communication systems, audio processing, medical devices, sensor interfaces, and control systems (Oxeltech, 2022).

Historically, designing analog filters requires manual calculations, repetitious simulations, and expansive expertise in circuit theory. As the complexity of circuits grows, the design process becomes increasingly difficult and time- consuming. In recent years, Artificial Intelligence and Machine Learning methods, especially Neural Networks, have emerged as effective tools for learning and modeling intricate systems. This thesis investigates the use of neural networks in the design of analog circuits to streamline, enhance, and optimize the process of determining component values for a variety of analog filters. In this thesis, circuit types will be explored and designed as follows,

Passive RC lowpass Filter: Circuit made up of a resistor R and a capacitor C. It allows low frequence signals to pass through while reducing the attenuation of high frequencies.

Sallen-Key Bandpass Filter: Comprises an operational amplifier along with resistors capacitors. This configuration provides a precise frequency-selective response with adjustable gain, allowing it to amplify signals within a specific frequency band while attenuating signals outside the band. The Sallen-Key topology is widely used in analog signal processing due to its simplicity, stability, and ease of tuning. By selecting appropriate resistor and capacitor values, both the center frequency and bandwidth of the filter can be controlled accurately, making it ideal for communication and measurement systems.

Single-Stage LC Bandpass Filter: Comprises an inductor L and a capacitor C. The resonant frequency of the circuit is determined by the values of L and C. It's capable of selecting specific frequency range while attenuating the levels of other frequencies.

Two Stage LC Bandpass Filter: Made up of two inductors, and two capacitors. A more advanced filter is formed by cascading Two LC bandpass

sections together. This configuration improves selectivity and provides a clearer bandpass response, which is beneficial for high- performance communication systems.

Gm-C Bandpass Filter is composed of a transconductance Gm element and a capacitor C. This type is very adjustable and suitable for use in CMOS technologies. By varying the transconductance, the center frequency and bandwidth of the filter can be easily tuned without changing passive components. This tunability makes Gm-C filters highly attractive for integrated circuit applications, especially in communication systems where flexibility and compactness are essential. Furthermore, their compatibility with low-voltage, low-power operation ensures efficient performance in modern portable devices.

Active Enhanced JFET MOSFET features gate, drain, and source terminals to achieve both direct and nonlinear characteristic actions. They allow precise control of current flow through the channel, making them suitable for both analog and digital signal processing. In addition, their scalability and compatibility with integrated circuit technology make them essential components in modern electronic systems.

2.2 THEORETICAL APPROACHES

2.2.1 The Role of Artificial Neural Networks in Analog Circuits

ANNs are strong tools that can closely match complicated nonlinear functions and find detailed patterns in big sets of data. This is especially useful in designing analog circuits, where the way different component values affect the circuit's performance, such as its frequency response, is often very nonlinear and hard to describe with standard math equations.

For example, the exact behavior of a multi-stage active filter depends on many factors like resistor, capacitor, and op-amp settings, which makes it hard to optimize by hand and takes a lot of time and repeated attempts. The method is called inverse neural networks (Jafari et al., 2010). It involves the following steps:

- Generate data containing the circuit parameters along with their corresponding magnitude frequency response, cutoff frequency, quality factor, and gain characteristics.
- Train the network with the specific dataset.
- Predict appropriate component values for new specifications by validations.

In the context of filter circuit design, an ANN can be trained to learn the inverse relationship between desired performance metrics and the required component values. The training process essentially teaches the network to "think like an experienced engineer" but at a speed and scale that is humanly impossible. By mapping desired circuit performance to component values, ANNs enable a "synthesis-by-example" approach, which stands in contrast to traditional, equation-based design methods. This allows for the rapid exploration of vast design spaces, potentially discovering novel and non-intuitive solutions that might be missed by human designers. This data-driven methodology not only accelerates the design cycle but also provides a robust framework for managing the inherent complexities and trade-offs of modern analog circuits.

2.2.2 The Deep Learning Methodology For Circuit Designs

Data Generation: The quality and quantity of the training data are critical. This data will be generated programmatically using MATLAB, which provides a robust and flexible environment for circuit analysis and data synthesis. The process involves scripting the circuit's transfer function to systematically vary component values (e.g., resistor, capacitor, and inductor values) and calculating the resulting circuit performance. This computational approach allows for the efficient capture of key parameters, such as magnitude response, cutoff frequency, gain and quality factor. By doing this, we create a rich dataset that maps the component values to the circuit performance, which is essential for training the neural network (Gencer et al., 2020).

Network Training and Optimization: During training, the ANN's internal weights and biases are adjusted to minimize the error or loss function — the difference between the network's predicted output and the actual output from the simulation data. The network learns to generalize from the provided examples. Techniques like backpropagation and stochastic gradient descent are used to iteratively update the network's parameters, pushing the model toward higher accuracy. This is where the "deep" in deep learning becomes relevant, as multiple hidden layers allow the network to learn progressively more abstract and complex features of the input-output relationship (Daylak et al., 2025).

Prediction and Validation: Once trained, the network becomes a powerful predictive model. For new, unseen design specifications (e.g., a specific cutoff frequency and gain), the network can instantaneously predict the corresponding component values. This rapid prediction capability dramatically reduces the design cycle time. The validation step involves simulating the new circuit with the predicted values to confirm that it meets the required performance specifications (Boonseng et al., 2019).

2.2.3 Choosing The Right Neural Network Architecture

Feedforward Neural Networks (FNNs): They are excellent for direct mapping problems, such as predicting a fixed set of component values from a fixed set of performance metrics. The structure is straightforward: input layer, one or more hidden layers, and an output layer. It will be used to create an inverse mapping method, which is to predict component values based on given circuit specifications. Unlike usual design methods, this approach can directly find the circuit's parameters from its desired performance features (Schmidt et al., 1992).

2.2.4 EST and ERT Strategies

2.2.4.1 Element Spreading Training (EST)

A systematic and deterministic method designed for comprehensive data generation in engineering and machine learning applications. The core principle of EST is to guarantee a uniform distribution of training data across the entire defined design space. This is achieved by creating a full-factorial grid of all possible component value combinations (Chen et al., 2025).

EST approach involves discretizing each parameter's operational range into a defined, finite number of steps. This process ensures that the training data uniformly covers the complete design space. thereby:

- Minimizing sampling bias: Every region of the parameter space is equally represented.
- Providing a comprehensive view: The input-output relationship is fully captured, ensuring consistent and predictable learning for subsequent surrogate modeling.

For a system with N components and K discrete values (or "spread values") selected per component, the total number of unique training samples generated is given by K^N. This strategy is highly effective for problems characterized by a low to moderate number of parameters (McCaffrey, 2015).

2.2.4.1.1 Design Space Mapping – 3Dimansional Cube

To systematically explore the behavior of the RC filter for instance, a full-factorial design was generated using EST. This method discretizes the parameter space by selecting multiple spread values for each design variable — in this case, the resistance R and capacitance C. With three discrete values for each component, the total number of configurations (3×3=9). Each point in the 3D design space plot represents one such configuration (See: Figure 2.6).

- The i-axis represents R values.
- The j-axis represents C values.
- The k-axis represents the angular cutoff frequency values.

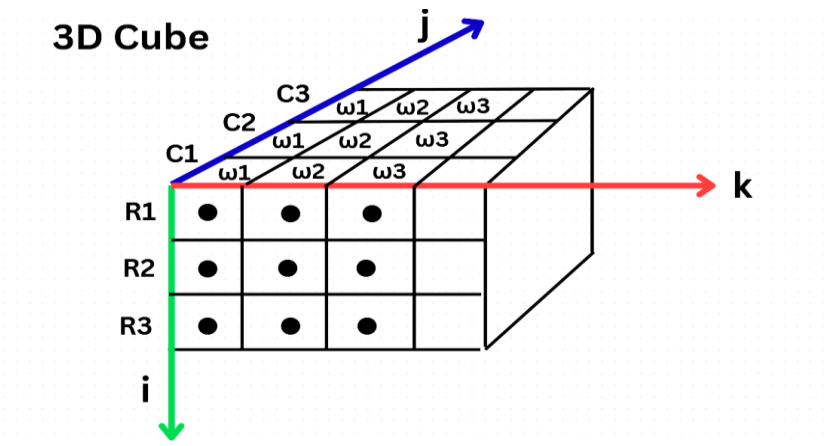


Figure 2.6 3D Cube Mapping

This cube-shaped visualization provides an immediate and clear view of how the cutoff frequency varies across the full range of R and C combinations. As expected from the formula, the plot visually confirms that increasing either R or C leads to a lower cutoff frequency. This 3D mapping serves to confirm the uniform and bias-free sampling of the design space, which is a crucial first step for training accurate machine learning models (Krennrich, 2020). If the filter has more than 2 components such as 4, the cube won't be able to show the 3D because it can only visualize 3 variables at once. With 4 variables or more, the

design becomes a Hypercube which cannot fully visualize in 3D unless fixing components happens. Furthermore, The frequency response mapping method comes right now.

2.2.4.1.2 Frequency Response Mapping

In addition to mapping the design space, the frequency-domain behavior of all nine generated configurations was analyzed across a wide range of angular frequencies, $\omega \in [10^0, 10^\infty]$ rad/s. The angular frequency can either be swept between a specific range of values or chosen as a vector (Lo et al., 2023). Each curve on the magnitude response plot corresponds to a specific pair, demonstrating how the filter attenuates frequencies above its characteristic cutoff angular frequency. By visualizing all nine magnitude response curves on a single plot, this representation supports critical design decisions by showing the complete spectrum of filter performance achievable within the defined parameter ranges. This provides an intuitive and comprehensive understanding of how component variation directly impacts filter performance over frequency (See: Figure 2.7).

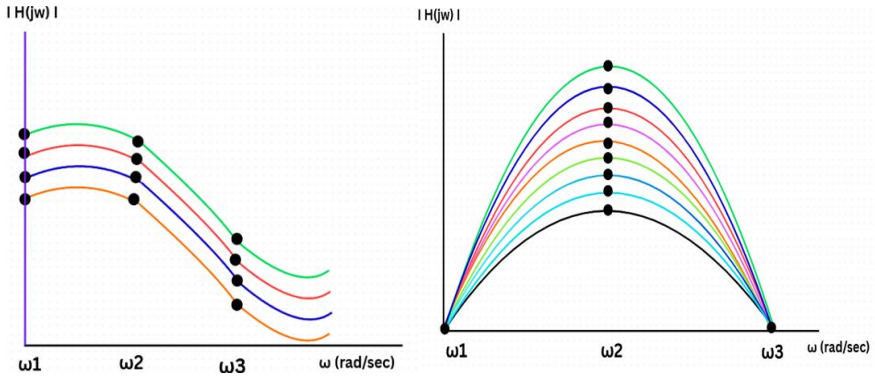


Figure 2.7 Magnitude Response Mapping

2.2.4.2 Element Randomization Training

A stochastic data generation strategy in which component values are randomly sampled from continuous or discrete ranges to create diverse and

extensive datasets. This random sampling approach is highly effective for high-dimensional circuit design problems where exhaustive enumeration, such as EST, becomes computationally prohibitive. Although ERT does not guarantee uniform coverage of the design space, techniques like Latin Hypercube Sampling can mitigate this by ensuring more even distribution of samples. Figure 2.8 explains the ERT method, which is widely used for generating training data to enable machine learning models, particularly deep neural networks, to generalize better across a broad range of realistic scenarios.

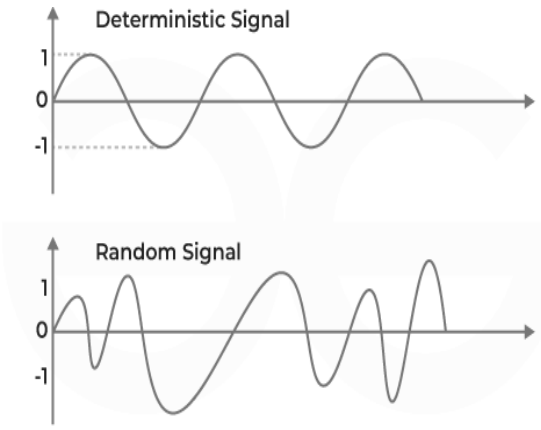


Figure 2.8 Element Randomization Training

When components vary continuously (e.g., resistors, capacitors, inductors), ERT can sample realistic operating conditions for training robust predictive models. However, randomized training allows exploration of complex and nonlinear parameter spaces, improving the likelihood of discovering global optima or robust designs. Furthermore, ERT supports the creation of training datasets for models that need to generalize well, especially when data dimensionality and variability are high.

2.3 LITERATURE REVIEW

Many researchers have investigated the effectiveness of machine learning algorithms in predicting resistor, capacitor and inductor values in electrical circuits. Allakhverdiyeva (2016) designed low-pass digital filters (IIR and FIR) and optimized the filters parameters using Gradient descent algorithm by back propagation method with the use of C++ program. The software calculates the required filter coefficients and presents them along with the frequency response of the generated filter. The effectiveness and reliability of the resulting filter are established by the approach of reducing the error at each iteration step during the synthesis process.

The design of low-pass FIR filters was also done by the authors Singh and Thakare (2015), who used three artificial neural network algorithms: regression, radial basis function, and feedforward and reverse propagation. MATLAB's Hamming window method is used to identify the filter coefficients for data organization. However, the methods used cutoff frequency (ω_c) and Scale (S) as input to obtain the frequency response h(n) of the filters. In comparison with the three methods, regression basis function (RBF) gives an accuracy of 99.9% and back propagation results approximately 98.37% and regression methods of 99.45%. RBF is the most accurate, efficient and less complex method.

The effectiveness of neural network methods for the Sallen-Key BPF circuit has been investigated by researchers. In order to improve the accuracy of neural network (NN)-based analog circuit design, Gan et al. (2019) combined Wavelet Neural Networks (WNN) with the Unscented Kalman Filter (UKF) technique to optimize WNN parameters. This method combines deep learning capacity for self-learning with time-frequency features. Standard tolerances for capacitance and resistance were specified at 10% and 5%, respectively. However, because tolerance ranges have an impact, this method produces better precision. Furthermore, Particle Swarm Optimization (PSO) was created by the author (Kumar et al., 2016) using MATLAB and a second order Sallen-Key bandpass filter. PSO has been used in a variety of fields, and because it is easier

to apply than other designs, parameters need to be changed. But in this architecture, the quality factor Q is too sensitive to gain. Because of the tolerances involved, the component spacing values have a significant impact on the filter's performance. Consequently, the PSO algorithm's accuracy and validations are supported by MATLAB.

Single-stage and Two-stage LC Ladders are unique designs, especially when the parameters can be predicted using neural networks. As a Single-stage LC bandpass filter design, the author (Mohamed et al., 2019) established Model Predictive Control (MPC) method for three-phase inverter connected with LC circuit based on feedforward neural networks strategy. It predicts the future behavior of the components to be controlled at a certain time. With the prediction, the goal is to provide a low Total Harmonic Distortion (THD) and better performance for many diverse types of loads. The linear and nonlinear load conditions of ANN-based controllers are compared by the performance with MPC in MATLAB. As a result, the simulations show that more than 50 tests have been done, and ANN-based controller is fast, safe and illustrates great steady and dynamic performance. Although artificial neural networks have been utilized in creating several types of analog filters, including microstrip and waveguide structures, a review of the literature indicates a notable absence of studies focusing on Two-Stage passive LC bandpass filters designed with neural networks. As a result, this current research seeks to address this gap by exploring the application of feedforward neural networks for forecasting the component values of Two-Stage LC bandpass filters based on their frequency response, providing a unique contribution to both analog filter design and the application of neural networks in the field of electronics.

The researcher Ivanova (2021) has introduced a tree-based algorithm for Gm-C filters utilizing machine learning. It operates at high frequencies, ensuring low power consumption while delivering excellent performance across a wide frequency range. A single OTA is employed to create a second-order bandpass filter with two different configurations, which are evaluated through RapidMiner Studio. The objective is to assess filter performance by utilizing various

parameters, including supply voltage (V_{dd}), Threshold voltage (V_{th}), Total Harmonic Distortion (THD), Dynamic Range (DR), active area, and filter order. Consequently, the results demonstrate that classification machine learning algorithms like Random Forest efficiently manage tasks related to the design and analysis of Gm-C filters, thus supporting the decision-making process. Likewise, the author (Hashim et al., 2014) developed a bandpass Gm-C filter using two operational transconductance amplifiers (OTAs), operating at frequencies ranging from 1 kHz to several MHz The magnitude responses illustrate how the bandpass filter operates as the capacitor values C_1 and C_2 change, with the quality factor (Q) influencing the filter's sharpness or smoothness in relation to the center frequency (ω_0) and bandwidth (β). While the simulation and theoretical methods show minor discrepancies, deep learning has not been utilized.

MOSFETs play a vital role in modern electronic industry. Wei et al., (2020) modeled an advanced MOSFET circuit using pre- and post-processing on the training data methodology based on neural networks achieving high precision. The input parameters undergo preprocessing prior to being input into the ANN, and the corresponding output is then post-processed to derive the drain current (I_D). The training dataset, consisting of (I_D) values across six different parameters, is produced through Spice circuit simulation software. The ANN is structured with three layers, each containing 32 neurons. The model training is conducted using Python. Pre- and post-processing ensures an accurate fit across a wide dynamic range of drain current. As a result, both n-type and p-type MOSFETs exhibit strong agreement between the ANN predictions and the BSIM model.

CHAPTER 3

3. THESIS STUDY

3.1 SYSTEM MODEL

This section presents the system model developed to design and analyze various analog electronic circuits using neural network–based methodologies. The approach integrates classical circuit theory with modern machine learning techniques to establish a data-driven framework capable of predicting optimal component values from a desired frequency response. The overall system consists of several stages, including synthetic data generation, dataset preprocessing, neural network training, and performance validation.

The generated datasets span a wide range of circuit parameters and their corresponding frequency responses, enabling the network to learn the underlying relationships between component variations and circuit behavior. Once trained, the neural network is capable of generalizing unseen specifications, allowing rapid estimation of component values with high accuracy. This predictive capability significantly accelerates the design process, minimizes manual trial-and-error procedures, and reduces the potential for human error. Furthermore, the model allows designers to explore a broader design space efficiently and provides insight into the sensitivity of circuit performance with respect to component variations. The system model incorporates the following components:

MATLAB Analytical and Technical Verification: Theoretical magnitude response evaluation using the derived transfer function.

Proteus Circuit Design and Simulation: Schematic diagram construction and magnitude response analysis.

3.2 RC LOWPASS FILTER

3.2.1 Filter Structure, Transfer Function and Magnitude Response

The RC circuit is a simple electronic circuit, which consists of a resistor, and a capacitor. It allows low frequencies to pass through, while attenuating high frequency signals (Corchete, 2019). The circuit is drawn (See: Figure 3.1).

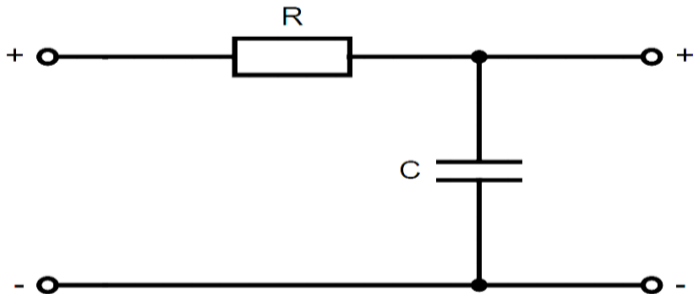


Figure 3.1 RC LPF Circuit

The input voltage V_{in} is in series with the resistor R, and the capacitor is parallel with the output voltage $Vout$. So, we need to convert the parameters to the frequency response. The components are written as impedances. The resistor impedance Z_R does not change, but the capacitor impedance $Z_C = \frac{1}{sC}$. The relation between the magnitude response of the frequency is important to be observed.

$$H(s) = \frac{V_o(s)}{V_{in}(s)} = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{\frac{1}{sC}}{\frac{sRC + 1}{sC}} = \frac{1}{sRC + 1} \qquad (\mathbf{3.1})$$

Dividing every item by RC gives,

$$H(s) = \frac{\frac{1}{RC}}{s + \frac{1}{RC}} = \frac{\omega_c}{s + \omega_c} \qquad (\mathbf{3.2})$$

$$H(j\omega) = \frac{V_o(j\omega)}{V_{in}(j\omega)} = \frac{\omega_c}{j\omega + \omega_c} \qquad (\mathbf{3.3})$$

where $\omega_c = \frac{1}{RC}$

The Magnitude Response $|H(j\omega)|$ then,

$$|H(j\omega)| = \frac{\omega_c}{\sqrt{\omega^2 + {\omega_c}^2}} \quad (3.4)$$

Phase Response,

$$\theta = -\tan^{-1}\left(\frac{\omega}{\omega_c}\right) = -\tan^{-1}(\omega RC) \quad (3.5)$$

3.2.2 MATLAB and Proteus Simulation Analysis

Assume that the cut-off frequency = 100 rad/s, resistor R = 1 kΩ, and the capacitor C = 10 µF. The magnitude response of the filter shows cut-off frequency at **15.92 Hz ≅ 16 Hz** (See: Figure 3.2).

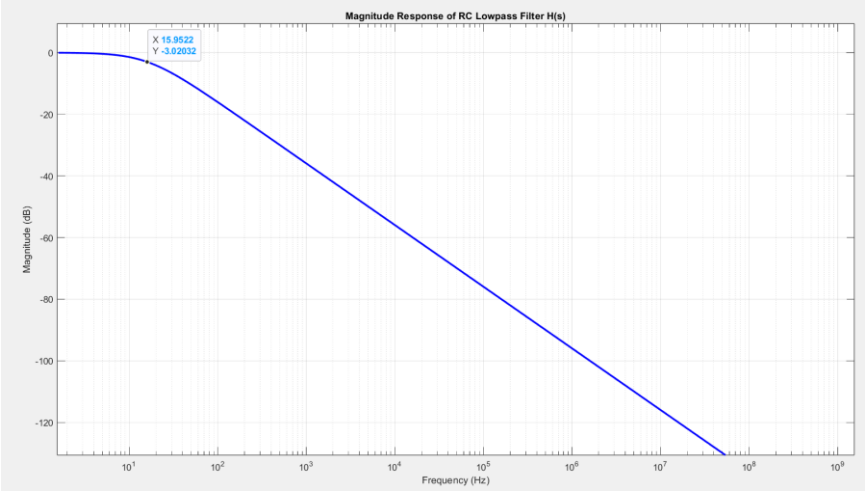


Figure 3.2 RC LPF Magnitude Response Analysis

$$20\log|H(j\omega)| = 20\log\left(\frac{\omega_c}{\sqrt{\omega^2 + {\omega_c}^2}}\right) \quad (3.6)$$

The magnitude response of the RC low-pass filter was first derived analytically and plotted using MATLAB. To validate the theoretical results, the

circuit will be simulated in Proteus, allowing the comparison of the simulated and analytical responses to verify the accuracy of the derived characteristics (See: Figure 3.3).

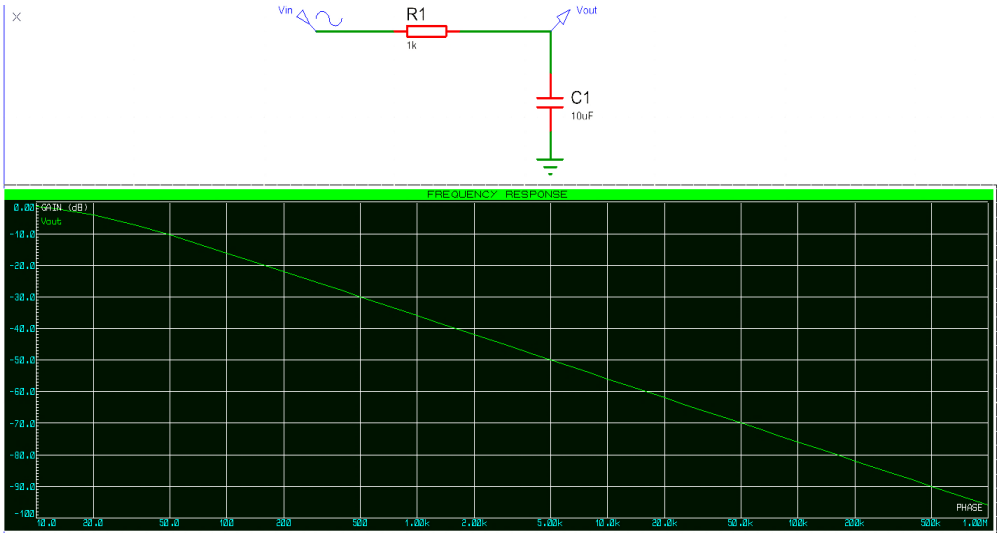


Figure 3.3 RC LPF Proteus Simulation

The theoretical transfer function of the RC low-pass filter was checked by comparing its magnitude response, calculated using MATLAB, with the simulated response from Proteus. With resistor value R = 1 kΩ and capacitor value C = 10 μF, both methods gave the same frequency response. This match shows that the transfer function is correct and dependable.

3.2.3 Parameter Sweeping of R and C

To Generate a comprehensive dataset, both R and C values are swept **(EST)** within user-defined ranges. The user selects the number of training points (Nt), which determines the resolution of the sweep. For Example, using Nt = 3 generates **9 possible combinations**.

$$N_t = (\text{Training Data}^{\text{ Number of Parameters}}) \qquad (3.7)$$

The resistor and capacitor values are distributed evenly across their ranges:

$$R_i = \in [R_{min}, R_{max}], \quad C_i = \in [C_{min}, C_{max}] \tag{3.8}$$

where i = (1,2,3……N_t)

For each pair (R_i, C_i), a frequency vector is generated and used to compute the magnitude response of the filter **(EM)**. This approach ensures uniform coverage of the design space and produces data suitable for building a reliable catalogue.

3.2.4 Generation of Frequency Response Catalogue

Each (R, C) pair produces a unique magnitude response curve over the selected frequency range. All responses are stored in a catalogue, forming a matrix of dimensions:

$$\text{Catalogue Size} = N_t^2 \times N_f \tag{3.9}$$

where N_f is the number of sampled frequency points.

3.2.5 User Interface Workflow

The developed MATLAB GUI integrates all steps of the process as follows:

- The customer enters the values of R, C, Nt, and their spreading ranges.
- The magnitude responses are computed and plotted.
- The program automatically generates the Element Matrix and the Catalogue.
- The customer may choose to view a single curve or compare selected curves.

3.2.6 Curve Selection and Modulation

The user may browse the frequency response catalogue and select any curve for detailed inspection. Upon selection, the modulation stage is applied

optionally, depending on the user's preference. The modulation process allows for the introduction of controlled variations to the selected response. Available modulation types include:

- No Modulation: The curve remains unchanged, representing the ideal response.
- Cosine Modulation: A cosine-shaped ripple is applied to the response.
- Sine Modulation: A sinusoidal variation modifies the curve.
- Triangle Modulation: A triangular waveform is used to adjust the response.

This modulation capability enables the simulation of non-idealities or specific design scenarios, providing a richer dataset for training the neural network and ensuring accurate prediction of component values even under perturbed conditions.

3.2.7 Neural Network Training Methodology

To predict the circuit parameters R and C by ANN of the RC filter, the neural network is employed to predict the optimal resistor R and capacitor C values corresponding to the curve selected by the customer. In the standard scenario, the magnitude response of the customer-selected curve is used to train the network, enabling it to predict the most accurate values of R and C.

If the customer applies modulation to the selected curve, the modulated response is used instead. In this case, the neural network is trained on the modulated data, and the prediction process produces the component values that best correspond to the modulated characteristics. This approach ensures that the ANN can accurately handle both unmodified and modulated responses, providing reliable parameter estimation under varying design conditions (See: Figure 3.4).

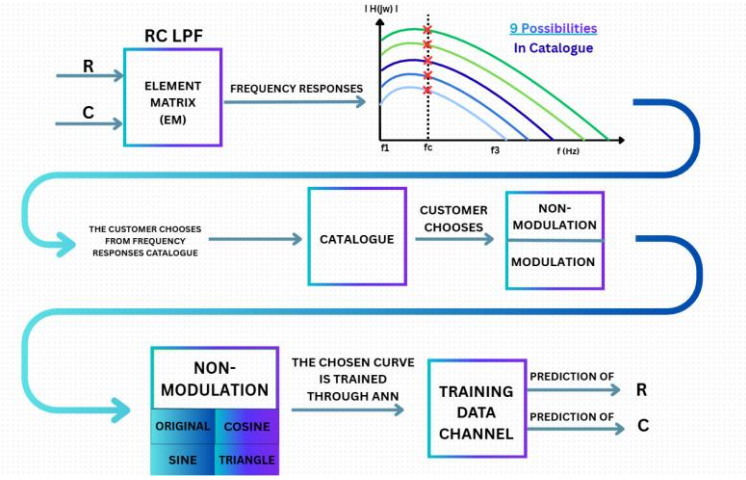


Figure 3.4 RC LPF ANN-Model

3.2.8 ANN Prediction of R and C

When the user provides an experimental magnitude response, the trained neural network predicts the corresponding values of R and C. The output is displayed in MATLAB along with the best-matched curve from the catalogue.

NOTE1: When ANN predicts non-modulated curve, it sees smooth curve from the original from the catalogue. However, the error becomes less with training due to the overlapping between the original and predicted curve.

$$0 < MSE < 1\ (dB^2) \qquad (\mathbf{3.10})$$

NOTE2: When Cosine ripple modulation is applied for instance,

$$H_{Modulation}(f) = H_{Selected}(f) + Acos(\emptyset(f)) \qquad (\mathbf{3.11})$$

This shows high-frequency oscillations that don't correspond to any physical combinations. Consequently, the ANN is being asked to fit responses outside the original RC behavior. The ANN tries its best to match the original chosen curve. Therefore, the MSE naturally increases−sometimes significantly, which is normal for the modulation scenarios.

$$MSE > 1\ (dB^2) \qquad (\mathbf{3.12})$$

3.3 SALLEN-KEY BANDPASS FILTER

3.3.1 Filter Structure, Transfer Function and Magnitude Response

A Voltage-controlled voltage-source Sallen-Key bandpass filter is a type of active filter design frequently applied in analog signal processing. It utilizes operational amplifiers (Op-Amps) in conjunction with resistors and capacitors to create bandpass second-order filters. The VCVS filter eliminates the need for inductors by enabling high Q factor and passband gain. Another benefit of a VCVS filter is its independence; it can be cascaded without the stages influencing one another's tuning. A VCVS filter version that makes use of a unity gain amplifier, sometimes known as a buffer amplifier, is called a Sallen–Key filter (Olalekan and Toluwani, 2017). Because it can precisely regulate frequency response parameters including the center frequency ω_0, bandwidth β, and quality factor Q, the Sallen-Key bandpass filter is significant. The passband's midpoint is defined by the center frequency ω_0, and the overall width of the permitted frequency range is determined by the bandwidth β.

The filter's sharpness of frequency selectivity is measured by the quality factor Q. The Sallen–Key design provides reduced sensitivity to loading effects, improved gain control, and amplification in contrast to passive bandpass filters, which frequently have insertion loss and restricted tuning possibilities. As a result, real-world systems operate more consistently (Denisenko et al., 2022).

The second order Sallen-Key bandpass filter allows range of frequencies to pass and attenuates frequencies outside the specific range. The targeted design contains five resistors (R_1, R_2, R_f, R_a, R_b), two capacitors (C_1 and C_2) and one operational amplifier (Zumbahlen, 2008). The general transfer function of a second order bandpass filter is,

$$\mathrm{H(s)} = \frac{\beta \mathrm{s}}{\mathrm{s}^2 + \beta \mathrm{s} + \omega_0^2} \qquad (\mathbf{3.13})$$

Sallen-Key bandpass filter is shown below (See: Figure 3.5). There are 4 equations and a constraint due to the existence of non-inverting amplifier.

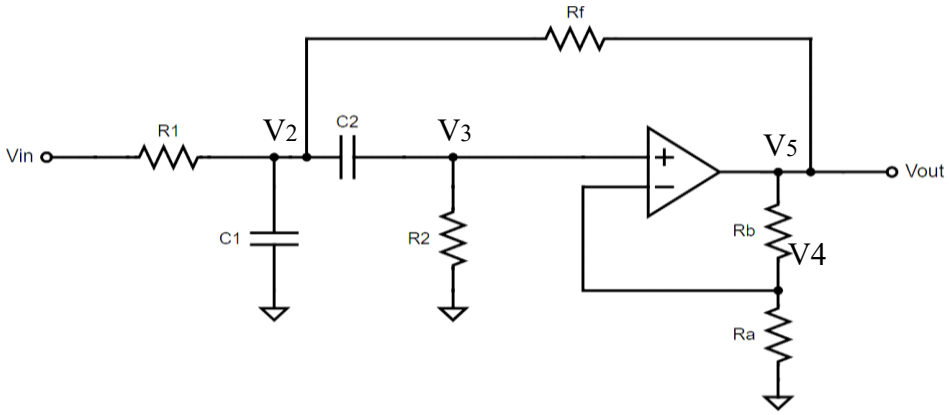


Figure 3.5 Sallen-Key BPF Circuit

The transfer function is going to be as,

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{V_5(s)}{V_2(s)} \cdot \frac{V_2(s)}{V_1(s)} \qquad (3.14)$$

Applying Kirchhoff's Current Low (KCL) at node V_2, V_3, and V_5 as follows,

$$\frac{V_2 - V_1}{R_1} + sC_1(V_2) + sC_2(V_2 - V_3) + \frac{V_2 - V_5}{R_f} = 0 \qquad (3.15)$$

$$sC_2(V_3 - V_2) + \frac{V_3}{R_2} = 0 \qquad (3.16)$$

$$\frac{V_5 - V_2}{R_f} + \frac{V_5 - V_4}{R_b} = 0 \qquad (3.17)$$

There is negative feedback in the op-amp configuration. As a constraint,

$$V_+ = V_- = V_{out} \qquad (3.18)$$

$$V_3 = V_4 = kV_5 \qquad (3.19)$$

$$V_3 = V_4 = \left(\frac{R_a}{R_a + R_b}\right) V_5 \qquad (\mathbf{3.20})$$

Simplifying the equation (**3.16**) as,

$$sC_2V_3 + \frac{V_3}{R_2} = sC_2V_2 \qquad (\mathbf{3.21})$$

Taking V_3 as a common factor,

$$V_3\left(sC_2 + \frac{1}{R_2}\right) = sC_2V_2 \xrightarrow{\text{yields}} V_3\left(\frac{sC_2R_2 + 1}{R_2}\right) = sC_2V_2 \qquad (\mathbf{3.22})$$

Dividing all terms by sC_2, V_2 is expressed in terms of V_3 as,

$$\boxed{V_2 = \left(\frac{sC_2R_2 + 1}{sC_2R_2}\right) \cdot V_3 \text{ or } V_2 = \left(1 + \frac{1}{sC_2R_2}\right) \cdot V_3} \qquad (\mathbf{3.23})$$

The constraint equation is involved to obtain a relation between V_2 and V_5. Subsequently, V_3 will be substituted by kV_5,

$$V_2 = \left(\frac{sC_2R_2 + 1}{sC_2R_2}\right) \cdot (V_3 = V_4 = kV_5) \qquad (\mathbf{3.24})$$

$$\boxed{V_2 = (kV_5)\left(\frac{sC_2R_2 + 1}{sC_2R_2}\right)} \qquad (\mathbf{3.25})$$

Substituting $V_4 = kV_5$ in equation (**3.17**) gives another relation between V_2 and V_5,

$$\frac{V_5 - V_2}{R_f} + \frac{V_5 - kV_5}{R_b} = 0 \qquad (\mathbf{3.26})$$

Factoring V_5 will result as follows,

$$\frac{V_5 - V_2}{R_f} + \frac{V_5\,(1 - k)}{R_b} = 0 \qquad (\mathbf{3.27})$$

Solving the gain K will result in a change in the second fraction,

$$\frac{V_5 - V_2}{R_f} + \frac{V_5}{R_a + R_b} = 0 \qquad (\mathbf{3.28})$$

Multiplying the denominator for simplification,

$$(V_5 - V_2)(R_a + R_b) + (R_f)(V_5) = 0 \qquad (\mathbf{3.29})$$

$$V_5R_a + V_5R_b - V_2R_a - V_2R_b + R_fV_5 = 0 \qquad (\mathbf{3.30})$$

$$V_5(R_a + R_b + R_f) = V_2(R_a + R_b) \qquad (\mathbf{3.31})$$

Dividing the whole equation by $R_a + R_b$ in order to obtain V_2 in terms of V_5

$$V_2 = \left(\frac{R_a + R_b + R_f}{R_a + R_b}\right)(V_5), \text{or } V_2 = \left(1 + \frac{R_f}{R_a + R_b}\right)(V_5) \qquad (\mathbf{3.32})$$

After the simplification of equations (**3.16**) and (**3.17**), substituting the resulted V_2 in the equation (**3.15**) that expresses the input and output voltage relation.

$$\frac{(kV_5)\left(\frac{sC_2R_2 + 1}{sC_2R_2}\right) - V_1}{R_1} + sC_1\left((kV_5)\left(\frac{sC_2R_2 + 1}{sC_2R_2}\right)\right) + sC_2\left((kV_5)\left(\frac{sC_2R_2 + 1}{sC_2R_2}\right) - kV_5\right) + \frac{(kV_5)\left(\frac{sC_2R_2 + 1}{sC_2R_2}\right) - V_5}{R_f} = 0 \qquad (\mathbf{3.33})$$

Symbolizing $\left(\frac{\mathbf{sC_2R_2+1}}{\mathbf{sC_2R_2}}\right) = \mathbf{A}$ for clarification,

$$\frac{(kAV_5) - V_1}{R_1} + sC_1(kAV_5) + \underbrace{\mathbf{sC_2(kAV_5 - kV_5)}} + \frac{kAV_5 - V_5}{R_f} = 0 \qquad (\mathbf{3.34})$$

Reducing **The Third Term** as follows,

$$sC_2kV_5(A\ -1\ ) = sC_2kV_5\left(\frac{sC_2R_2+1}{sC_2R_2}-1\right) =$$

$$sC_2kV_5\left(\frac{sC_2R_2+1-sC_2R_2}{sC_2R_2}\right) = sC_2kV_5\left(\frac{1}{sC_2R_2}\right) = \mathbf{\frac{kV_5}{R_2}} \qquad (\mathbf{3.35})$$

Rewriting the equation (**3.34**) with the new finding,

$$\frac{(kAV_5)-V_1}{R_1} + sC_1(kAV_5) + \left(\mathbf{\frac{kV_5}{R_2}}\right) + \frac{kAV_5\ -\ V_5}{R_f} = 0 \qquad (\mathbf{3.36})$$

The equation below resulted in two unknowns as input and output voltage,.

$$V_5\left(\frac{kA}{R_1} + sC_1kA + \frac{k}{R_2} + \frac{kA-1}{R_f}\right) = \frac{V_1}{R_1} \qquad (\mathbf{3.37})$$

Dividing $\frac{V_5}{V_1}$ in both sides results,

$$\frac{V_5}{V_1} = \frac{\frac{1}{R_1}}{\frac{kA}{R_1} + sC_1kA + \frac{k}{R_2} + \frac{kA-1}{R_f}} \qquad (\mathbf{3.38})$$

where $\mathbf{k} = \left(\mathbf{\frac{R_a}{R_a+R_b}}\right)$ and $\boldsymbol{A} = \left(\boldsymbol{\frac{sC_2R_2+1}{sC_2R_2}}\right)$

The denominator consists of four terms. Each term is substituted by their values in order to restore the denominator to its original term.

First Term,

$$\frac{kA}{R_1} = \frac{\left(\frac{R_a}{R_a+R_b}\right)\left(\frac{sC_2R_2+1}{sC_2R_2}\right)}{R_1} =$$

$$\frac{R_a}{(R_a+R_b)R_1}\cdot\frac{sC_2R_2+1}{sC_2R_2} = \boxed{\mathbf{\frac{R_a}{(R_a+R_b)R_1} + \frac{R_a}{(R_a+R_b)R_1sC_2R_2}}} \qquad (\mathbf{3.39})$$

Second Term,

$$sC_1kA\ =\ sC_1\left(\frac{R_a}{R_a+R_b}\right)\left(\frac{sC_2R_2+1}{sC_2R_2}\right) =$$

$$\left(\frac{R_a}{R_a+R_b}\right)\left(sC_1+\frac{C_1}{C_2R_2}\right)=\boxed{\left(\frac{\mathbf{R_a}}{\mathbf{R_a+R_b}}\right)\mathbf{sC_1}+\frac{\mathbf{R_a}}{\mathbf{R_a+R_b}}\left(\frac{\mathbf{C_1}}{\mathbf{C_2R_2}}\right)} \quad (\mathbf{3.40})$$

Third Term,

$$\frac{k}{R_2}=\left(\frac{\frac{R_a}{R_a+R_b}}{R_2}\right)=\boxed{\frac{\mathbf{R_a}}{(\mathbf{R_a+R_b})\mathbf{R_2}}} \quad (\mathbf{3.41})$$

Fourth Term,

$$\frac{kA-1}{R_f}=\frac{\left(\frac{R_a}{R_a+R_b}\right)\left(\frac{sC_2R_2+1}{sC_2R_2}\right)-1}{R_f}$$

$$=\boxed{\frac{\mathbf{R_a}}{(\mathbf{R_a+R_b})\mathbf{sC_2R_2R_f}}-\frac{\mathbf{R_b}}{(\mathbf{R_a+R_b})\mathbf{R_f}}} \quad (\mathbf{3.42})$$

All denominator terms show three types of fractions $(s^1), (s^0)$, and $\left(\frac{1}{s}\right)$ term. Collecting the like-terms gives the transfer function.

$$\frac{\frac{1}{R_1}}{\left(\frac{R_a}{R_a+R_b}\right)C_1s+\left(\frac{R_a}{(R_a+R_b)R_1}+\frac{R_a}{R_a+R_b}\left(\frac{C_1}{C_2R_2}\right)+\frac{R_a}{(R_a+R_b)R_2}-\frac{R_b}{(R_a+R_b)R_f}\right)+\frac{R_a}{(R_a+R_b)sC_2R_1R_2}+\frac{R_a}{(R_a+R_b)sC_2R_fR_2}} \quad (\mathbf{3.43})$$

Lastly, multiplying all terms by $\left(\frac{R_a+R_b}{R_a}\right)$, then by (s), and after that, by C_1 gives the H(s) of the Sallen-Key BPF.

$$\boxed{H(s)=\frac{\left(1+\frac{R_b}{R_a}\right)\frac{s}{R_1C_1}}{s^2+s\left(\frac{1}{R_1C_1}+\frac{1}{C_2R_2}+\frac{1}{R_2C_1}-\frac{R_b}{R_a\,R_f\,C_1}\right)+\frac{R_1+R_f}{C_1\,C_2\,R_1\,R_2\,R_f}}} \quad (\mathbf{3.44})$$

The center frequency of the band is determined by the constant term as,

$$\omega_0{}^2 = \frac{R_1 + R_f}{C_1 C_2 R_1 R_2 R_f} \xrightarrow{\text{Solving for } \omega_0} \omega_0 = \sqrt{\frac{R_1 + R_f}{C_1 C_2 R_1 R_2 R_f}} \qquad (\mathbf{3.45})$$

The Bandwidth β is determined by the s term of the H(s) as,

$$\beta = \frac{\omega_0}{Q} = \frac{1}{R_1 C_1} + \frac{1}{C_2 R_2} + \frac{1}{R_2 C_1} - \frac{R_b}{R_a R_f C_1} \qquad (\mathbf{3.46})$$

The quality factor Q relies on the center frequency and bandwidth,

$$Q = \frac{\omega_0}{\beta} = \frac{\omega_0}{2\xi\omega_0} = \frac{\omega_0}{\frac{\omega_0}{Q}} = \frac{\sqrt{\frac{R_1 + R_f}{C_1 C_2 R_1 R_2 R_f}}}{\frac{1}{R_1 C_1} + \frac{1}{C_2 R_2} + \frac{1}{R_2 C_1} - \frac{R_b}{R_a R_f C_1}}$$

$$= \frac{\sqrt{(R_1 + R_f) C_1 C_2 R_1 R_2 R_f}}{R_1 R_f (C_1 + C_2) + R_2 C_2 \left(R_f - \frac{R_b}{R_a} R_1\right)} \qquad (\mathbf{3.47})$$

The voltage divider Gain that is obtained from the negative feedback shown in Figure 3.5 that controls the Op-amp's inner Gain $\left(G = 1 + \frac{R_b}{R_a}\right)$. The bandwidth β is determined by four terms,

$\frac{\mathbf{1}}{\boldsymbol{R_1 C_1}}$ ≅ Lower cut-off frequency for High pass ω_1.

$\frac{\mathbf{1}}{\boldsymbol{C_2 R_2}}$ ≅ Upper cut-off frequency for Low pass ω_2.

$\frac{\mathbf{1}}{\boldsymbol{R_2 C_1}}$ Adjust the band shape.

$-\frac{\boldsymbol{R_b}}{\boldsymbol{R_a R_f C_1}}$ Identifies how sharp the band is.

With the use of the bandwidth β and the center frequency ω_0, the damping ratio ξ can be established to explain how much oscillations die out in the system.

$$2\xi\omega_0 = \beta \xrightarrow{\text{Solving for } \xi} \xi = \frac{\beta}{2\omega_0}$$

$$2\xi\omega_0 = \frac{\omega_0}{Q} \xrightarrow{Solving\ for\ \xi} \xi = \frac{1}{2Q} \qquad (\mathbf{3.48})$$

If damping ratio is large (Large damping), then β is wide which result small sharpness Q (Flat peak)

If damping ratio is small, then β is narrow which results in large sharpness Q (Tall sharp peak)

If damping ratio $\boldsymbol{\xi \cong \frac{1}{\sqrt{2}} \cong 0.707}$, the Q $\cong$ 1.414, then the peak is balanced with no ripples, and this is called as 'Butterworth Case'

As an example, the characteristics of the band-pass filter are examined by varying the quality factor Q over different values [0.5, 0.707, 1, 2, 5, 10]. The center frequency is set to ω_0 =14849 rad/s, and the parameter β changes accordingly with variations in Q (See: Figure 3.6).

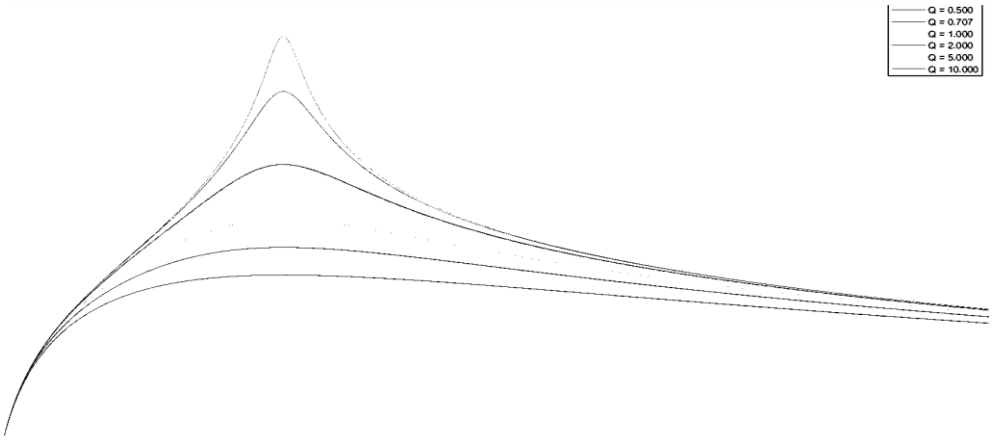


Figure 3.6 Sallen-Key BPF Magnitude Responses Varying Q

Q observations were important to differentiate the band shape. Furthermore, the transfer function H(s) is converted to H(jω) in order to obtain the magnitude response of the circuit. Each s is substituted by jω, where s = jω,

$$\mathrm{H(j\omega)} = \frac{\mathrm{V_o(j\omega)}}{\mathrm{V_{in}(j\omega)}} \frac{\left(1+\frac{\mathrm{R_b}}{\mathrm{R_a}}\right)\frac{\mathrm{j\omega}}{\mathrm{R_1C_1}}}{(\mathrm{j\omega})^2 + (\mathrm{j\omega})\left(\frac{1}{\mathrm{R_1C_1}} + \frac{1}{\mathrm{C_2R_2}} + \frac{1}{\mathrm{R_2C_1}} - \frac{\mathrm{R_b}}{\mathrm{R_aR_fC_1}}\right) + \frac{\mathrm{R_1+R_f}}{\mathrm{C_1C_2R_1R_2R_f}}} \qquad (\mathbf{3.49})$$

$$H(j\omega) = \frac{(K)\frac{j\omega}{R_1 C_1}}{(j\omega)^2 + (j\alpha\omega) + (\omega_0)^2} \qquad (3.50)$$

The magnitude response is expressed as,

$$|H(j\omega| = \frac{K\,\omega\,\omega_1}{\sqrt{(\omega_0{}^2 - \omega^2)^2 + (\alpha\omega)^2}} \qquad (3.51)$$

The phase angle is also expressed as,

$$\theta = \frac{\pi}{2} - \tan^{-1}\left(\frac{\text{Imaginary}}{\text{Real}}\right) = \frac{\pi}{2} - \tan^{-1}\left(\frac{\alpha\omega}{\omega_0{}^2 - \omega^2}\right) \qquad (3.52)$$

3.3.2 MATLAB and Proteus Simulation Analysis

In order to illustrate the characteristics of the filter using MATLAB, assume the resistors' values given as $R_1 = R_2 = R_f = R_a = 1.5k\Omega$, $R_b = 3k\Omega$ and $C_1 = C_2 = 100nF$ (See: Table 3.1).

Table 3.1 Sallen–Key BPF Components and Frequency Response

Circuit Components						
R_1	R_2	R_a	R_b	R_f	C_1	C_2
1.5kΩ	1.5kΩ	1.5kΩ	3kΩ	1.5kΩ	100nF	100nF
At −3dB, Center Frequency in rad/s						
ω_0		ω_{c1}		ω_{c2}		β
9396.62 rad/s		6650		13277.69		6627.69
At −3dB, Center Frequency in Hz						
f_0		f_{c1}		f_{c2}		β
1500.5 Hz		1058.38 Hz		2113.21 Hz		1054.82

The inner gain G = 3, and the quality factor Q = 1.42 (See: Figure 3.7). The peak gain $G_{dB} = 9.540$. The actual gain will result in the following,

$$G_{dB} = 20\log_{10}(G) \xrightarrow{\text{yields}} G = 10^{\left(\frac{9.540}{20}\right)} \approx 3 \qquad (\mathbf{3.53})$$

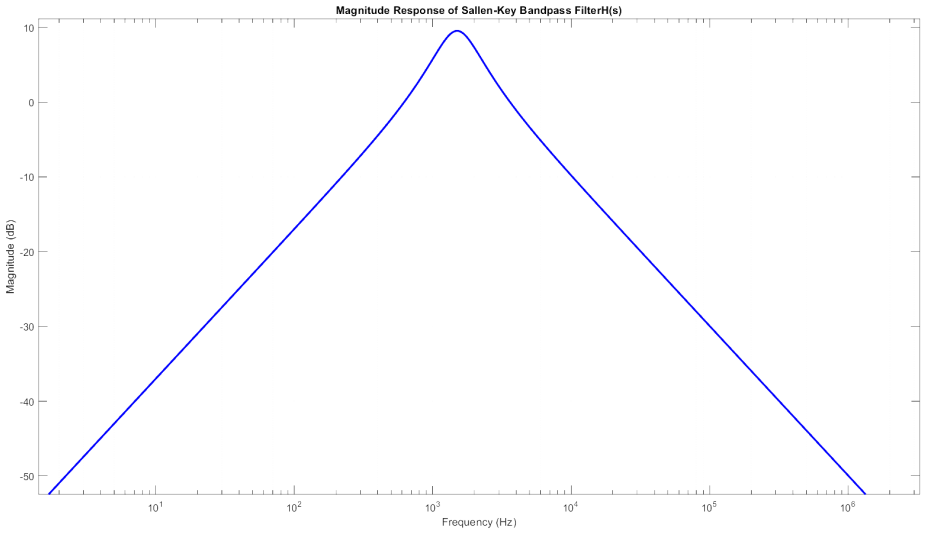


Figure 3.7 Sallen-Key BPF Magnitude Response

To verify the accuracy of the derived transfer function, the magnitude response was compared with simulation results (See: Figure 3.8). The theoretical transfer function of the Sallen-Key bandpass filter was checked by comparing its magnitude response, calculated using MATLAB, with the simulated response from Proteus. Both methods gave the same magnitude response.

This match shows that the transfer function is correct and dependable. This agreement between theoretical and simulated results confirms that the analytical model accurately represents the circuit's frequency behavior. Minor differences observed at higher frequencies can be attributed to simulation tolerances and component non-idealities. Overall, the comparison validates the correctness of the derived transfer function and demonstrates the reliability of the design. The equations can be designed to give the most suitable values for each component depending on the needs. There will be 8 variables and 4 equations.

- Choose the gain K, and set $R_b = (K-1)R_a$
- Choose C_1, and calculate $R_1 = \frac{1}{C_1\omega_0}$

- Choose C_2, and calculate $R_2 = \frac{R_1+R_f}{C_1C_2R_1R_f(\omega_0)^2}$
- Bandwidth that varies Q, $\frac{\omega_0}{Q} = \frac{1}{R_1C_1} + \frac{1}{C_2R_2} + \frac{1}{R_2C_1} - \frac{R_b}{R_aR_fC_1}$

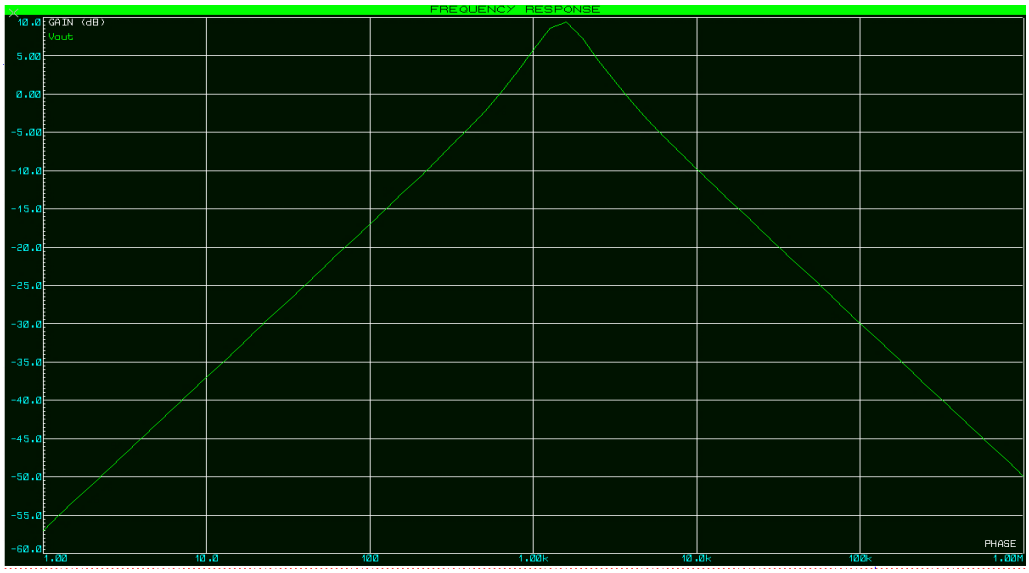


Figure 3.8 Sallen-Key BPF Proteus Magnitude Response

3.3.3 Parameter Sweeping of C_1 and R_f

To construct a comprehensive dataset for the Sallen–Key band-pass filter, the parameters $\boldsymbol{C_1}$ and $\boldsymbol{R_f}$ are swept using the EST method within user-specified limits. The customer selects the number of testing points (N_t), which defines the density of the sweep. For instance, selecting $N_t = 3$ results in $3 \times 3 = 9$ distinct parameter combinations.

The capacitor and feedback resistor values are uniformly distributed over their respective ranges:

$$C_{1,i} = \in [C_{1,min}, C_{1,max}], \quad R_{f,i} = \in [R_{f,min}, R_{f,max}] \qquad (\mathbf{3.54})$$

where i = (1,2,3……N_t)

For each pair $(C_{1,i}, R_{f,i})$, a frequency vector is generated and used to

compute the magnitude response of the filter **(EM)**. This approach ensures uniform coverage of the design space and produces data suitable for building a reliable catalogue.

3.3.4 Generation of Frequency Response Catalogue

Each $(C_{1,i}, R_{f,i})$ combination generates a distinct magnitude response over the specified frequency range. All computed responses are compiled into a catalogue, forming a structured matrix whose dimensions are given by:

$$\text{Catalogue Size} = N_t^2 \times N_f \qquad (\mathbf{3.55})$$

Where N_f represents the number of sampled frequency points used to evaluate the filter response. This catalogue provides a complete set of reference curves corresponding to every swept parameter combination.

3.3.5 User Interface Workflow

The developed MATLAB GUI integrates the complete workflow for the Sallen–Key bandpass filter as follows:

- The customer inputs the nominal component values along with the spreading percentage and selects the number of training points.
- The program performs parameter sweeping on the selected components and computes the corresponding magnitude responses over the defined frequency range.
- An Element Matrix and a full Catalogue of frequency responses are automatically generated based on all possible combinations.
- The customer may choose to view a single curve, view all curves, or compare selected curves.

3.3.6 Curve Selection and Modulation Options

After generating the catalogue of frequency responses, the user may select any curve for closer examination. Once a curve is chosen, an optional

modulation stage can be applied through the MATLAB environment. The modulation feature introduces controlled distortions to the selected frequency response, enabling the analysis of non-ideal or perturbed behaviors. The available modulation options include (Original, Cosine, Sine and Triangle Modulation).

This flexibility allows the designer to model practical imperfections and generates more diverse training data for the neural network. Consequently, the prediction of component values remains accurate even when the input response exhibits deviations from the ideal Sallen–Key behavior.

3.3.7 Neural Networks Training Methodology

For the Sallen-Key BPF, a trained artificial neural network (ANN) is utilized to predict the component values $\boldsymbol{C_1}$ and $\boldsymbol{R_f}$ corresponding to the frequency response selected by the user. In the standard scenario, the ANN is trained using the magnitude response of the chosen curve, enabling it to accurately estimate the optimal inductor and capacitor values that reproduce the desired response.

If the user applies any modulation or alteration to the selected curve, the network is instead trained on the modulated response. This ensures that the ANN accounts for the adjusted characteristics, allowing it to reliably predict component values that correspond to both the original and the modified frequency responses (See: Figure 3.9).

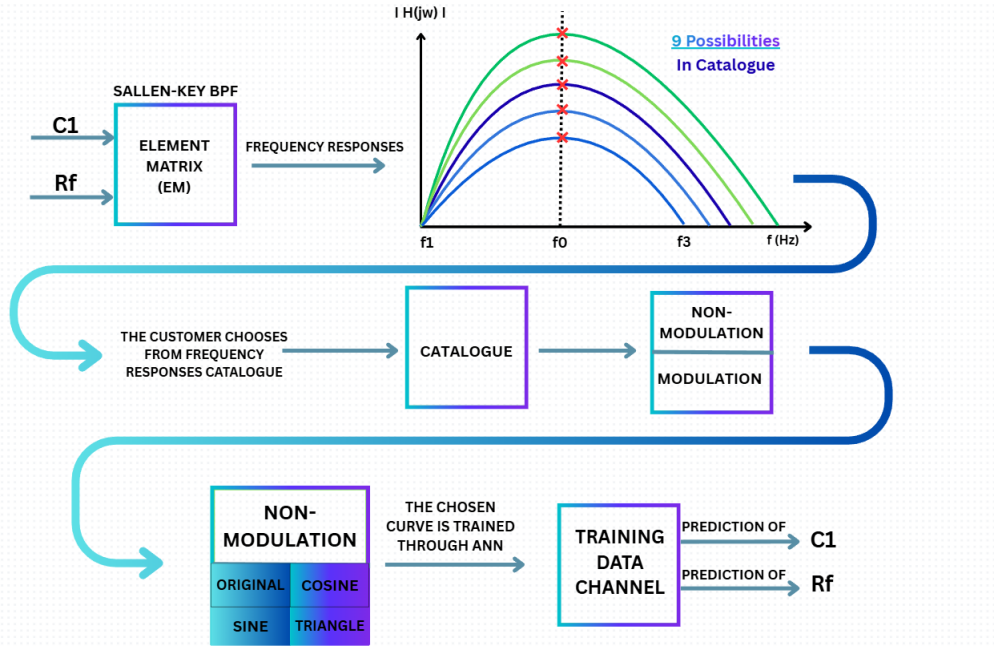


Figure 3.9 Sallen-Key BPF ANN-Model

3.3.8 ANN Prediction of C_1 and R_f

Only the nominal component values are provided by the user through the GUI; all remaining computations, including frequency responses, catalogue generation, and ANN predictions, are produced directly by MATLAB.

When the user provides an experimental or simulated magnitude response, the trained neural network predicts the corresponding values of $\mathbf{C_1}$ **and** $\mathbf{R_f}$. The predicted outputs are displayed in MATLAB alongside the best-matched curve obtained from the catalogue.

NOTE 1: For original responses,

$$0 < MSE < 1 \qquad (3.56)$$

NOTE 2: When cosine, sine or triangle ripple modulation are applied,

$$MSE > 1\ (dB^2) \qquad (3.57)$$

3.4 SINGLE STAGE LC BANDPASS FILTER

3.4.1 Filter Structure, Transfer Function and Magnitude Response

Inductor-Capacitor (LC) is a simple electronics circuit which allows a specific range of frequencies to pass and attenuates the frequencies outside that range (See: Figure 3.10). They are utilized for tuning radio frequencies, or for adjusting the bass and treble sound elements of mid-range speakers. Passive circuits cannot amplify the signal, unlike the active circuits due to its power. For an LC design, a resistor at the input can be added to reduce the quality factor Q of the circuit. When the Q is high, it results in sharper signal (Narrow β). The resistor controls the quality factor, bandwidth and reduces the peak gain.

When the bandwidth β is large, the quality factor Q becomes small, resulting in a smoother and wider in the signal response. Conversely, when the bandwidth is narrow (small β), the quality factor increases, producing a sharper and more selective frequency response (Panasonic Industry, 2018).

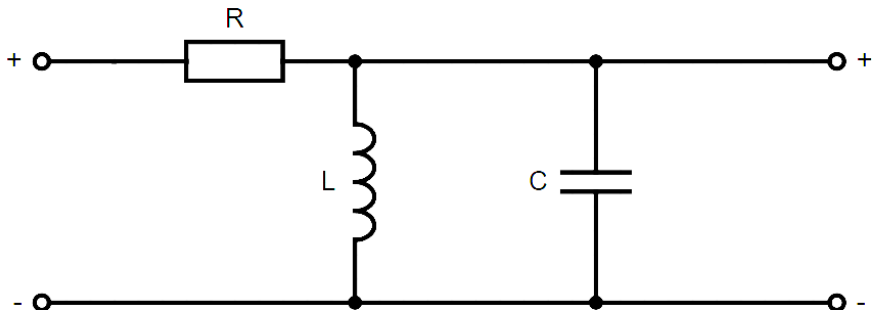


Figure 3.10 Single-Stage LC BPF Circuit

The second order bandpass filter transfer function general form written as,

$$H(s) = \frac{\beta s}{s^2 + \beta s + {\omega_0}^2} \qquad (\mathbf{3.58})$$

The transfer function H(s) must be derived by finding the output and input voltages,

$$V_{out} = Z_{Out} \times I_{in} \tag{3.59}$$

The output impedance Z_{Out} = (C//L). such that,

$$V_{out} = \left(\frac{1}{sC} // (sL)\right) I_{in} = \left(\frac{\frac{sL}{sC}}{\frac{1}{sC} + sL}\right) I_{in}$$

$$= \left(\frac{\frac{sL}{sC}}{\frac{s^2LC + 1}{sC}}\right) I_{in} = \frac{sL}{s^2LC + 1}(I_{in}) \tag{3.60}$$

Summing the resistor with output impedance gives the input impedance,

$$Z_{Input} = R + Z_{out} = R + \frac{sL}{s^2LC + 1} = \frac{s^2RLC + sL + R}{s^2LC + 1} \tag{3.61}$$

Dividing the input voltage over the input impedance determines the input current.

$$I_{in} = \frac{V_{in}}{Z_{Input}} = \frac{V_{in}}{\frac{s^2RLC + sL + R}{s^2LC + 1}} = \frac{V_{in}(s^2LC + 1)}{s^2RLC + sL + R} \tag{3.62}$$

Substituting equation (**3.62**) in the V_{out} equation (**3.59**) results,

$$V_{out} = \frac{sL}{s^2LC + 1} \cdot \left(\frac{V_{in}(s^2LC + 1)}{s^2RLC + sL + R}\right) \tag{3.63}$$

Rearrange the equation (**3.63**) to find the transfer function H(s).

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{sL}{s^2RLC + sL + R} = \frac{s\left(\frac{1}{RC}\right)}{s^2 + s\left(\frac{1}{RC}\right) + \left(\frac{1}{LC}\right)} \tag{3.64}$$

In order to obtain the magnitude response, H(s) is converted to the frequency domain H(jω) as,

$$H(j\omega) = \frac{V_o(j\omega)}{V_{in}(j\omega)} = \frac{(j\omega)(\beta)}{(j\omega)^2 + (j\omega)(\beta) + ({\omega_0}^2)} \quad (3.65)$$

$$H(j\omega) = \frac{(j\omega\beta)}{(-\omega)^2 + (j\omega\beta) + ({\omega_0}^2)} \quad (3.66)$$

$$|H(j\omega)| = \frac{\omega\beta}{\sqrt{({\omega_0}^2 - \omega^2)^2 + (\omega\beta)^2}} \quad (3.67)$$

where center frequency ω_0, bandwidth β and quality factor Q as,

$$\omega_0 = \frac{1}{\sqrt{LC}}, \qquad \beta = \frac{1}{RC}, \qquad Q = \frac{\omega_0}{\beta} \quad (3.68)$$

3.4.2 MATLAB and Proteus Simulation Analysis

For instance, assume R = 50Ω , L = 10mH and C = 1μF. The components are shown in a table (See: Table 3.2).

Table 3.2 Single-Stage LC BPF Components and Frequency Response

Circuit Components			
R	L	C	
50Ω	10mH	1μF	
At −3dB, Center Frequency in rad/s			
ω_0	ω_{c1}	ω_{c2}	β
10000 rad/s	4155.46 rad/s	24064.8 rad/s	≅ 20000
At −3dB, Center Frequency in Hz			
f_0	f_{c1}	f_{c2}	β
1591.75 Hz	658.47 Hz	3848.48 Hz	3190.01

The characteristics of the magnitude response will be displayed (See: Figure 3.11). f_0 = 1602.59 Hz. At −3dB, the cut-off frequencies f_{c1} = 661.36

Hz, f_{c2} =3830.03 Hz, bandwidth = 3183.09 Hz and Q = 0.505.

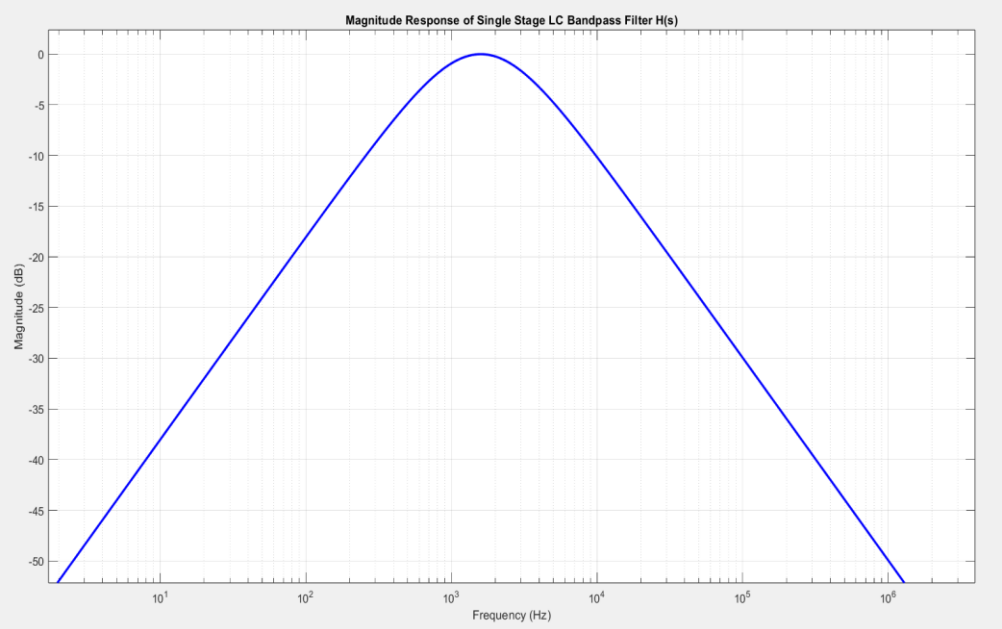


Figure 3.11 Single-Stage LC BPF Magnitude Response

This quality factor looks appropriate, and shows a smooth bandpass shape, but it changes depending on the application. For instance, FM radios and RF filters. Then, the Q must be higher, (e. g Q = 10, 20, 40,…). In general, when,

- Quality Factor between $(1 - 5)$, Low
- Quality Factor between $(5 - 20)$, Medium
- Quality Factor (> 20), High

In order to check the validity of the analytical results, the circuit was modeled in Proteus. The theoretical transfer function of the Single Stage LC bandpass filter was checked by comparing its magnitude response, calculated using MATLAB, with the simulated response from Proteus. Both methods gave the same magnitude response (See: Figure 3.12).

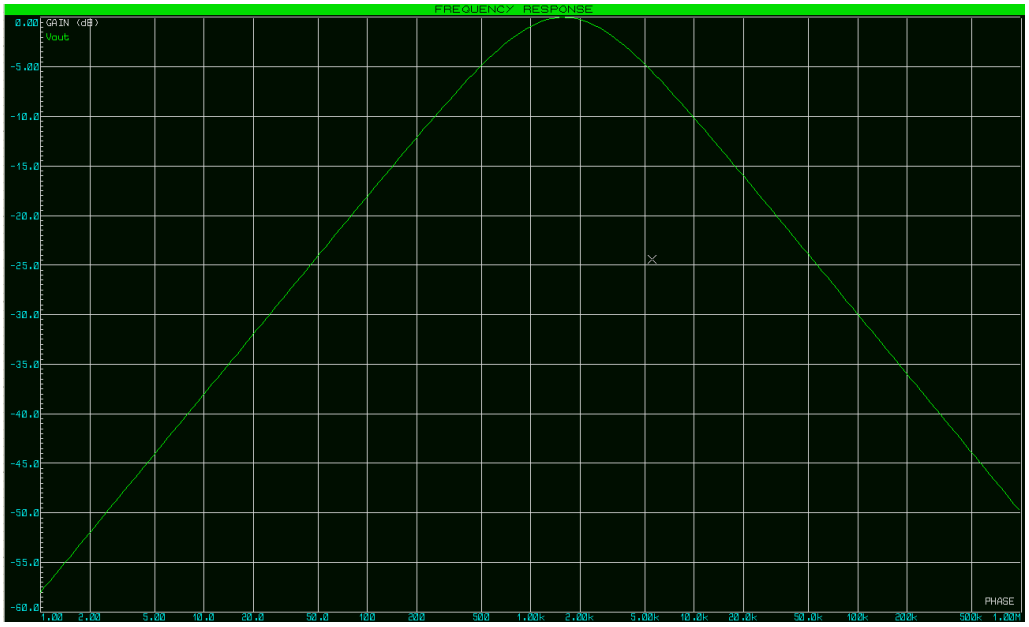


Figure 3.12 Single-Stage LC BPF Proteus Magnitude Response

3.4.3 Parameter Sweeping of L and C

To construct a comprehensive dataset for **L and C** parameters are swept using the EST method within customer-specified limits. The customer selects the number of testing points (N_t), which defines the density of the sweep. For instance, selecting $N_t = 3$ results in $3 \times 3 = 9$ distinct parameter combinations.

The inductor and capacitor values are uniformly distributed over their respective ranges:

$$L_i = \in [L_{min}, L_{max}], \qquad C_i = \in [C_{min}, C_{max}] \tag{3.69}$$

where i = (1,2,3……N_t)

For each pair (L_i, C_i), a frequency vector is generated and used to compute the magnitude response of the filter **(EM)**. This approach ensures uniform coverage of the design space and produces data suitable for building a reliable catalogue.

3.4.4 Generation of Frequency Response Catalogue

Each (L_i, C_i) combination generates a distinct magnitude response over the specified frequency range. All computed responses are compiled into a catalogue, forming a structured matrix whose dimensions are given by:

$$\text{Catalogue Size} = N_t^2 \times N_f \qquad (\mathbf{3.70})$$

Where N_f represents the number of sampled frequency points used to evaluate the filter response. This catalogue provides a complete set of reference curves corresponding to every swept parameter combination.

3.4.5 User Interface Workflow

The developed MATLAB GUI integrates the complete workflow for the Single–Stage LC BPF as follows:

- The customer inputs the nominal component values along with the spreading percentage and selects the number of training points.
- The program performs parameter sweeping on the selected components and computes the corresponding magnitude responses over the defined frequency range.
- An Element Matrix and a full Catalogue of frequency responses are automatically generated based on all possible combinations.
- The customer may choose to view a single curve, view all curves, or compare selected curves.

3.4.6 Curve Selection and Modulation Options

After the catalogue of frequency responses has been generated, the user can select any specific curve for detailed inspection. Upon selection, an optional modulation stage may be applied within the MATLAB environment. This modulation capability introduces controlled variations to the chosen frequency response, facilitating the study of non-ideal or perturbed circuit behaviors. The

available modulation types are as follows (No Modulation, Cosine Modulation, Sine Modulation and Triangle Modulation).

This approach provides flexibility to simulate practical imperfections and enriches the dataset for neural network training. As a result, the network can accurately predict component values even when the input response deviates from the ideal Single–Stage LC BPF behavior.

3.3.7 Neural Networks Training Methodology

For the Single Stage LC Bandpass Filter, the artificial neural network (ANN) is employed to estimate the inductor L and capacitor C values corresponding to a frequency response selected by the user. In a standard scenario, the ANN is trained using the magnitude response of the chosen curve, enabling precise prediction of the component values required to achieve that response.

When the user introduces modulation or modifications to the selected curve, the network is trained on the altered response. This approach ensures that the ANN can accurately predict the component values for both the original and the modified frequency responses, maintaining reliable performance under varying conditions (See: Figure 3.13).

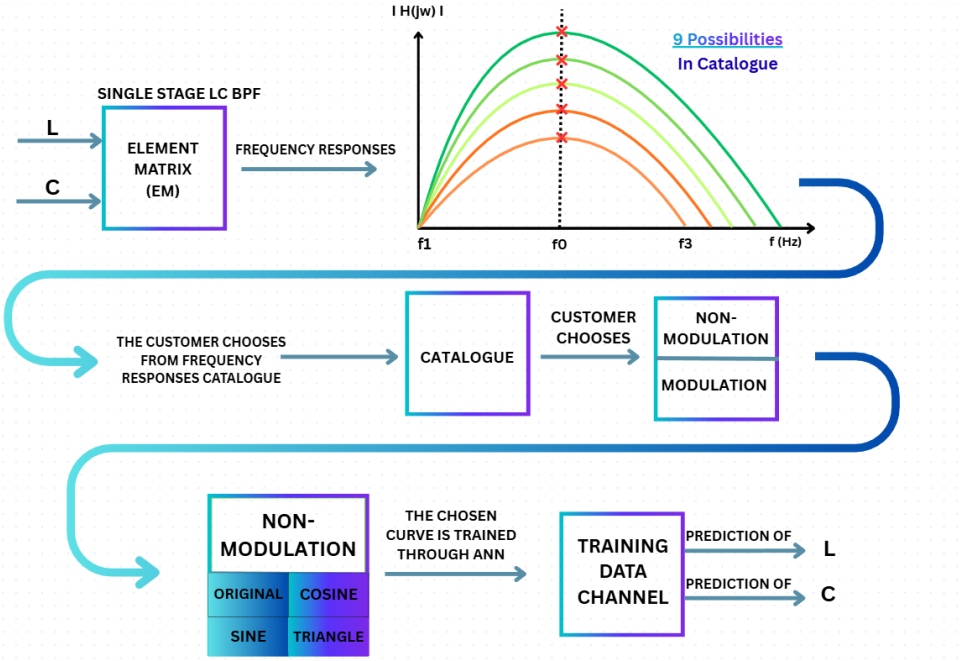


Figure 3.13 Single-Stage LC BPF ANN-Model

3.4.8 ANN Prediction of L and C

Similarly, only the nominal component values are provided by the user through the GUI, all remaining computations, including frequency responses, catalogue generation, and ANN predictions, are produced directly by MATLAB.

When the user provides an experimental or simulated magnitude response, the trained neural network predicts the corresponding values of **L and C**. The predicted outputs are displayed in MATLAB alongside the best-matched curve obtained from the catalogue.

NOTE 1: For non-modulated responses,

$$0 < MSE < 1 \qquad (\mathbf{3.71})$$

NOTE 2: When cosine ripple modulation is applied,

$$MSE > 1\ (dB^2) \qquad (\mathbf{3.72})$$

3.5 TWO STAGE LC BANDPASS FILTER

3.5.1 Filter Structure, Transfer Function and Magnitude Response

Two Stage LC bandpass filter is an electronic circuit created to permit signals within a certain frequency range to pass through while reducing the intensity of frequencies beyond that range. It is made up of two LC stages, usually arranged to increase selectivity and enhance the filter's performance compared to a single-stage design. Bandwidth is more restricted compared to single stage filter because of the cascading effect, which enhances selectivity (See: Figure 3.14).

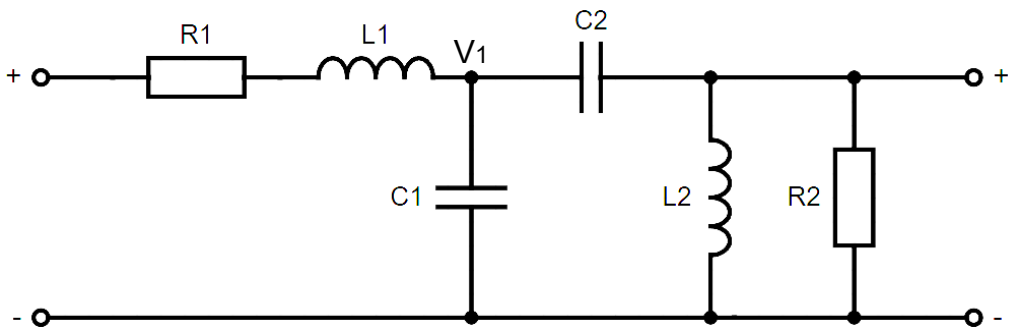


Figure 3.14 Two-Stage LC BPF Circuit

Discovering the transfer function $H(s) = \frac{V_{out}(s)}{V_{in}(s)}$, the input impedance Z_{in} and the voltage V_1 are calculated.

The output voltage using a voltage division rule expressed as,

$$V_{out} = V_1 \left(\frac{L_2 \,//\, R_2}{(L_2 // R_2) + C_2} \right) \qquad (3.73)$$

Simplifying the input impedance gives,

$$Z_{in} = R_1 + sL_1 + \left(C_1 \,//\, \left(C_2 + (R_2 \,//\, L_2) \right) \right) \qquad (3.74)$$

$$Z_{in} = \frac{s^3C_1C_2L_2R_1R_2 + s^2C_1L_2R_1 + s^2L_2C_2R_1 + sC_2R_2R_1 + sC_1R_2R_1 + s^2L_1C_2R_2}{s^3C_1C_2L_2R_2 + s^2C_1L_2 + s^2L_2C_2 + s^2C_2R_2 + sC_1R_2}$$

$$= \frac{s^3L_1L_2C_2 + s^4L_1L_2C_1C_2R_2 + s^3L_1L_2C_1 + s^2L_1C_1R_2 + s^2R_2L_2C_2 + sL_2 + R_2}{Same\ D(s)} \quad (\mathbf{3.75})$$

$$V_{in} = Z_{in} \times I_{in} \xrightarrow{\text{Solving for } I_{in}} I_{in} = \frac{V_{in}}{Z_{in}} \quad (\mathbf{3.76})$$

The input current I_{in} determined from the equation (**3.76**) above,

$$I_{in} = \frac{V_{in}(s^3C_1C_2L_2R_2 + s^2C_1L_2 + s^2L_2C_2 + s^2C_2R_2 + sC_1R_2)}{s^3C_1C_2L_2R_1R_2 + s^2C_1L_2R_1 + s^2L_2C_2R_1 + sC_2R_2R_1 + sC_1R_2R_1 + s^2L_1C_2R_2 +}$$

$$= \frac{\text{Same N(s)}}{s^3L_1L_2C_2 + s^4L_1L_2C_1C_2R_2 + s^3L_1L_2C_1 + s^2L_1C_1R_2 + s^2R_2L_2C_2 + sL_2 + R_2} \quad (\mathbf{3.77})$$

where the whole denominator written as,

$$s^3C_1C_2L_2R_1R_2 + s^2C_1L_2R_1 + s^2L_2C_2R_1 + sC_2R_2R_1 + sC_1R_2R_1 + s^2L_1C_2R_2 +$$

$$s^3L_1L_2C_2 + s^4L_1L_2C_1C_2R_2 + s^3L_1L_2C_1 + s^2L_1C_1R_2 + s^2R_2L_2C_2 + sL_2 + R_2 \quad (\mathbf{3.78})$$

Apply **KCL** to find the voltage at node V_1,

$$\frac{V_{in} - V_1}{R_1 + sL_1} = sC_1V_1 + \frac{V_1}{\frac{1}{sC_2} + \frac{sL_2R_2}{R_2 + sL_2}} \quad (\mathbf{3.79})$$

$$\frac{V_{in} - V_1}{R_1 + sL_1} = sC_1V_1 + V_1\left(\frac{s^2L_2C_2 + sC_2R_2}{s^2L_2C_2R_2 + sL_2 + R_2}\right) \quad (\mathbf{3.80})$$

$$\frac{V_{in} - V_1}{R_1 + sL_1} = V_1\left(sC_1 + \frac{s^2L_2C_2 + sC_2R_2}{s^2L_2C_2R_2 + sL_2 + R_2}\right) \quad (\mathbf{3.81})$$

$$V_{in} - V_1 = V_1\left(sC_1(R_1 + sL_1) + \frac{s^2L_2C_2 + sC_2R_2(R_1 + sL_1)}{s^2L_2C_2R_2 + sL_2 + R_2}\right) \quad (\mathbf{3.82})$$

$$V_{in} = V_1\left(1 + sC_1(R_1 + sL_1) + \frac{s^2L_2C_2 + sC_2R_2(R_1 + sL_1)}{s^2L_2C_2R_2 + sL_2 + R_2}\right) \quad (\mathbf{3.83})$$

$$V_1 = \frac{V_{in}}{\frac{s^2L_2C_2 + sC_2R_2(R_1 + sL_1)}{s^2L_2C_2R_2 + sL_2 + R_2} + sC_1(R_1 + sL_1) + 1} \quad (\mathbf{3.84})$$

First denominator term results as,

$$s^3L_2C_2L_1 + s^2L_2C_2R_1 + s^2L_1C_2R_2 + sC_2R_2R_1 \quad (\mathbf{3.85})$$

Second denominator term simplified as,

$$s^3C_1C_2L_2R_1R_2 + s^2L_2C_1R_1 + sC_1R_1R_2 + s^4C_1C_2L_1L_2R_2 + s^3L_1L_2C_1 + s^2L_1C_1R_2 \quad (\mathbf{3.86})$$

Third denominator term gives,

$$s^2L_2C_2R_2 + sL_2 + R_2 \quad (\mathbf{3.87})$$

Final form of V_1 written this way,

$$\frac{V_{in}(s^2L_2C_2R_2 + sL_2 + R_2)}{\begin{array}{c}(s^3L_2C_2L_1 + s^2L_2C_2R_1 + s^2L_1C_2R_2 + sC_2R_2R_1) + \\ (s^3C_1C_2L_2R_1R_2 + s^2L_2C_1R_1 + sC_1R_1R_2 + s^4C_1C_2L_1L_2R_2 + s^3L_1L_2C_1 + s^2L_1C_1R_2) \\ +(s^2L_2C_2R_2 + sL_2 + R_2)\end{array}} \quad (\mathbf{3.88})$$

Substituting V_1 in equation (**3.73**) to get H(s),

$$V_{out} = \frac{V_{in}(s^2C_2L_2R_2)}{\begin{array}{c}s^3L_2C_2L_1 + s^2L_2C_2R_1 + s^2L_1C_2R_2 + sC_2R_2R_1 + s^3C_1C_2L_2R_1R_2 \\ +s^2L_2C_1R_1 + sC_1R_1R_2 + s^4C_1C_2L_1L_2R_2 + \\ s^3L_1L_2C_1 + s^2L_1C_1R_2 + s^2L_2C_2R_2 + sL_2 + R_2\end{array}} \quad (\mathbf{3.89})$$

The transfer function H(s),

$$H(s) = \frac{s^2 C_2 L_2 R_2}{\begin{matrix} s^4 L_1 C_1 L_2 C_2 R_2 + s^3 L_1 L_2 C_2 + s^3 C_1 C_2 L_2 R_1 R_2 + s^3 L_1 C_1 L_2 + s^2 L_2 C_2 R_1 \\ + s^2 L_1 C_2 R_2 + s^2 L_2 C_1 R_1 + s^2 L_1 C_1 R_2 + s^2 L_2 C_2 R_2 \\ + s C_2 R_1 R_2 + s C_1 R_1 R_2 + s L_2 + R_2 \end{matrix}} \quad (3.90)$$

Normalizing the H(s) shows the final form as,

$$H(s) = \frac{s^2 \left(\frac{1}{L_1 C_1}\right)}{\begin{matrix} s^4 + s^3 \left(\frac{1}{C_1 R_2} + \frac{R_1}{L_1} + \frac{1}{C_2 R_2}\right) + s^2 \left(\frac{R_1}{L_1 C_1 R_2} + \frac{1}{L_2 C_1} + \frac{R_1}{L_1 C_2 R_2} + \frac{1}{L_2 C_2} + \frac{1}{L_1 C_1}\right) \\ + s \left(\frac{R_1}{L_1 L_2 C_1} + \frac{R_1}{L_1 L_2 C_2} + \frac{1}{L_1 C_1 C_2 R_2}\right) + \frac{1}{L_1 C_1 L_2 C_2} \end{matrix}} \quad (3.91)$$

The key role to understand Two-stage characteristics, the numerator and denominator terms are explained as,

Numerator N(s): where the term s^2 cause two zeros at s = 0 (DC and very low frequencies are blocked), and the other term comes from the first stage tank's natural frequency that defines energy exchange frequency for the first stage.

Denominator D(s): Each term corresponds to a certain physical feature phenomenon such as dynamics, stability and shape,

- s^4: Highest power of s, which is always 1 in monic polynomial which is the standard form of analyzing system behavior.
- s^3: Damping determines how the system responds to sudden changes. Controls bandwidth and peaking — larger resistors = more damping = wider bandwidth and smaller peak.
- s^2: Determines center frequency and the interaction between the two resonators.
- s: Represents energy losses that depend on both inductive and capacitive couplings. However, as an effect, it fine-tunes transition slope and phase response around the passband edges.
- Constant: Determines low-frequency attenuation.

3.5.2 MATLAB and Proteus Simulation Analysis

Assume $R_1 = R_2 = 1$ kΩ, $L_1 = L_2 = 10$mH, $C_1 = C_2 = 100\mu$F. The components' calculation is written in a table below for observation (See: Table 3.3).

Table 3.3 Two-Stage LC BPF Components and Frequency Response

Circuit Components						
L_1	L_2	C_1	C_2	R_1	R_2	
10mH	10mH	100μF	100μF	1kΩ	1kΩ	
At −3dB, Center Frequency in rad/s						
ω_0		ω_{c1}		ω_{c2}		β
1414.28 rad/s		1400.44 rad/s		1426.82 rad/s		26
At −3dB, Center Frequency in Hz						
f_0		f_{c1}		f_{c2}		β
225.92 Hz		223 Hz		227.2 Hz		4.2

To illustrate the behavior of a fourth-order system, The transfer function H(s) is calculated as,

$$H(s) = \frac{1x10^6 s^2}{s^4 + 100020s^3 + 5x10^6 s^2 + 2.0001x10^{11} s + 1x10^{12}} \qquad (3.92)$$

The obtained transfer function H(s) is put in MATLAB in order to observe the characteristics of the Fourth order filter (See: Figure 3.15).

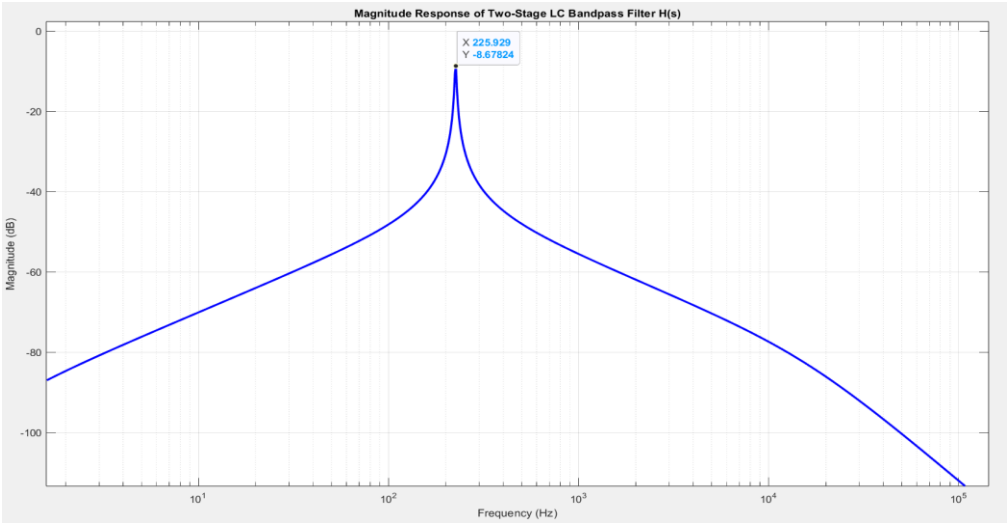


Figure 3.15 Two-Stage LC BPF Magnitude Response

The poles of the denominator are extremely critical where the center frequency is decided from. Solving the poles,

- Pole_1: $\text{s} = -9990 + \text{j}0 \cong -10000$ (Real)
- $\text{Pole}_{2,3}$: $\text{s} = -13 \pm \text{j}1414$ (Complex Conjugate Pairs)
- Pole_4: $\text{s} = -5 + \text{j}0$ (Real)

The two real poles are associated with one affects the high frequencies (-10000, Fast mode), and one affects low frequencies (-5, Slow mode) phase behavior. The complex conjugate pairs are the resonant mode that produce the peak in magnitude response around the imaginary part. The complex pole found as,

$$p = \sigma \pm j\omega_d \qquad (\mathbf{3.93})$$

where damped angular frequency $\omega_\text{d} = \text{Im(p)}$, Exponential decay rate $\sigma = \text{Re(p)}$ (σ is negative for stable systems) and Undamped natural frequency $\omega_0 = \sqrt{\sigma^2 + {\omega_\text{d}}^2}$

When damping is small, ($|\sigma| \ll \omega_\text{d}$), $\omega_0 \approx \omega_\text{d}$

However, $Q = \frac{\omega_0}{-2\sigma}$ (because σ is negative)

According to the complex conjugate pair found, $s = -13 \pm \text{j}1414$, the

imaginary part $\omega_d = 1414$ and the real part $\sigma = -13$. The center frequency = 1414.05 rad/s. Q = 54.39. Lastly, the bandwidth β as,

$$\Delta\omega = \beta \approx \frac{\omega_0}{Q} = \frac{1414.05}{54.39} \approx 26\text{rad/s} \qquad (\mathbf{3.94})$$

According to the figure 3.15, center frequency was found at 1419.55 rad/s. At -3dB, cut-off frequencies of the band were chosen to be at ω_{c1} = 1400 rad/s and ω_{c2} = 1428.6 rad/s. Furthermore, other frequency intersections are noted as,

- ω = 10 rad/s at $|H(j\omega|_{dB} = -$ 86.988 dB
- ω = 1000 rad/s at $|H(j\omega|_{dB} = -40$ dB
- ω = 1419.55 rad/s at $|H(j\omega|_{dB} = -8.678$ dB
- ω = $1x10^6$ rad/s at $|H(j\omega|_{dB} = -120$ dB

NOTE: Figure 3.15 magnitude response is represented in Hz, However, the center frequency $f_0 \cong 226$ Hz, $f_{c1} \cong 222.81$ Hz, $f_{c2} = 227.36$ Hz and $\beta =$ 4.55 Hz. Theoretically, in order to find the frequency ω = 1000 rad/s, take the real part of the denominator poles as,

$$\omega^4 - 5x10^6\omega^2 + 1x10^{12} = 0 \qquad (\mathbf{3.95})$$

Solving the quartic equation for ${\omega_1}^2$ and ${\omega_2}^2$ gives,

$$\frac{+5x10^6 \pm \sqrt{(5x10^6)^2 - 4(1x10^{12})}}{2} \qquad (\mathbf{3.96})$$

$${\omega_1}^2 = \pm 4.79125^{\,6}\ \text{rad/sec} \xrightarrow{\text{yields}} \omega_1 = 2188.89\ \text{rad/s}$$

$${\omega_2}^2 = \pm 208750\ \text{rad/sec} \xrightarrow{\text{yields}} \omega_2 = 456.89\ rad/s$$

Multiplying both frequencies defines the angular frequency at $-40dB$. Where,

$$\omega_{-40dB} = \sqrt{\omega_1\omega_2} = 1000.040\ rad/s$$

Converting H(s) to Frequency response H(jω). Each s is substituted by (jω) in equation (**3.91**). However, $(j\omega)^2 = -\omega^2$, $(j\omega)^3 = -j\omega^3$ and $(j\omega)^4 = \omega^4$

$$H(j\omega) = \frac{(j\omega)^2\left(\frac{1}{L_1C_1}\right)}{(j\omega)^4 + (j\omega)^3\left(\frac{1}{C_1R_2} + \frac{R_1}{L_1} + \frac{1}{C_2R_2}\right) + (j\omega)^2\left(\frac{R_1}{L_1C_1R_2} + \frac{1}{L_2C_1} + \frac{R_1}{L_1C_2R_2} + \frac{1}{L_2C_2} + \frac{1}{L_1C_1}\right) + (j\omega)\left(\frac{R_1}{L_1L_2C_1} + \frac{R_1}{L_1L_2C_2} + \frac{1}{L_1C_1C_2R_2}\right) + \frac{1}{L_1C_1L_2C_2}} \quad (\mathbf{3.97})$$

$$= \frac{-\omega^2\left(\frac{1}{L_1C_1}\right)}{\omega^4 - j\omega^3\left(\frac{1}{C_1R_2} + \frac{R_1}{L_1} + \frac{1}{C_2R_2}\right) - \omega^2\left(\frac{R_1}{L_1C_1R_2} + \frac{1}{L_2C_1} + \frac{R_1}{L_1C_2R_2} + \frac{1}{L_2C_2} + \frac{1}{L_1C_1}\right) + (j\omega)\left(\frac{R_1}{L_1L_2C_1} + \frac{R_1}{L_1L_2C_2} + \frac{1}{L_1C_1C_2R_2}\right) + \omega_0^4} \quad (\mathbf{3.98})$$

$$= \frac{-\left(\frac{\omega^2}{L_1C_1}\right)}{\left[\omega^4 - \omega^2\left(\frac{R_1}{L_1C_1R_2} + \frac{1}{L_2C_1} + \frac{R_1}{L_1C_2R_2} + \frac{1}{L_2C_2} + \frac{1}{L_1C_1}\right) + \omega_0^4\right] + j\left[-\omega^3\left(\frac{1}{C_1R_2} + \frac{R_1}{L_1} + \frac{1}{C_2R_2}\right) + (\omega)\left(\frac{R_1}{L_1L_2C_1} + \frac{R_1}{L_1L_2C_2} + \frac{1}{L_1C_1C_2R_2}\right)\right]} \quad (\mathbf{3.99})$$

The magnitude response |H(jω)| expressed as,

$$|H(j\omega)| = \frac{+\left(\frac{\omega^2}{L_1C_1}\right)}{\sqrt{\left(\omega^4 - \omega^2\left(\frac{R_1}{L_1C_1R_2} + \frac{1}{L_2C_1} + \frac{R_1}{L_1C_2R_2} + \frac{1}{L_2C_2} + \frac{1}{L_1C_1}\right) + \omega_0^4\right)^2 + \left(-\omega^3\left(\frac{1}{C_1R_2} + \frac{R_1}{L_1} + \frac{1}{C_2R_2}\right) + \omega\left(\frac{R_1}{L_1L_2C_1} + \frac{R_1}{L_1L_2C_2} + \frac{1}{L_1C_1C_2R_2}\right)\right)^2}} \quad (\mathbf{3.100})$$

Matching the magnitude response obtained from Proteus with that derived in MATLAB is essential for evaluating the accuracy of the filter design. (See: Figure 3.16). It can be observed that the magnitude responses obtained from MATLAB and Proteus are in close agreement.

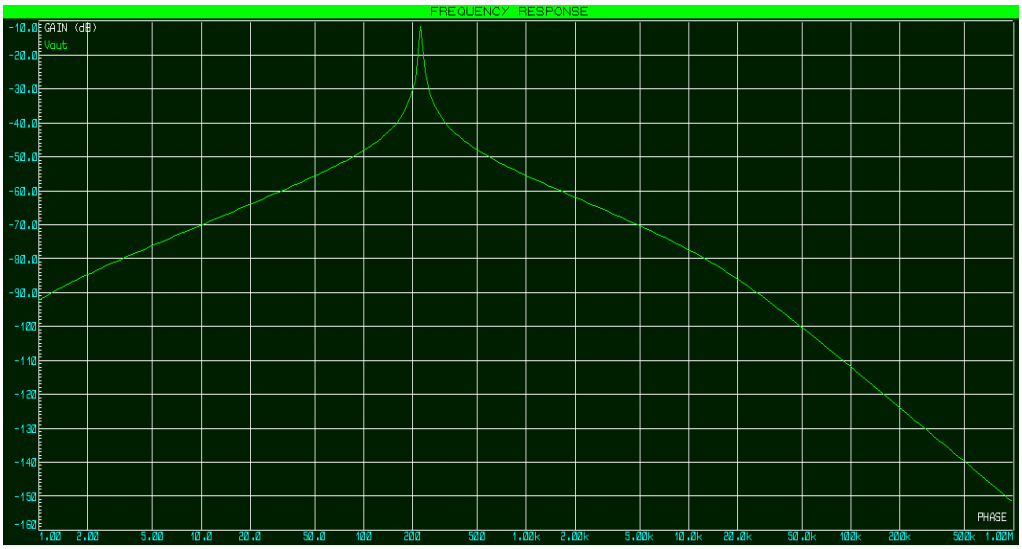


Figure 3.16 Two-Stage LC BPF Proteus Magnitude Response

3.5.3 Parameter Sweeping of L_1, L_2, C_1 and C_2

To construct a comprehensive dataset for $\mathbf{L_1}$, $\mathbf{L_2}$, $\mathbf{C_1}$ and $\mathbf{C_2}$ parameters are swept within customer-specified limits. The customer selects the number of testing points (N_t), which defines the density of the sweep. Selecting $N_t = 3$ results in $3 \times 3 \times 3 \times 3 = 81$ distinct parameter combinations. The inductor and capacitor values are uniformly distributed over their respective ranges:

$$L_{1,i \text{ and } 2,i} = \in [L_{min}, L_{max}], \quad C_{1,i \text{ and } 2,i} = \in [C_{min}, C_{max}] \qquad (\mathbf{3.101})$$

where i = (1,2,3……N_t)

For each pair $(L_{1i}, L_{2i}, C_{1i}, C_{2i})$, a frequency vector is generated to compute the magnitude response of the filter **(EM)**. This approach ensures uniform coverage of the design space and produces data suitable for building a reliable catalogue.

3.5.4 Generation of Frequency Response Catalogue

Each $(L_{1i}, L_{2i}, C_{1i}, C_{2i})$ combination generates a distinct magnitude response over the specified frequency range. All computed responses are

compiled into a catalogue, forming a structured matrix whose dimensions are given by:

$$\text{Catalogue Size} = N_t^2 \times N_f \qquad (\mathbf{3.102})$$

where N_f represents the number of sampled frequency points used to evaluate the filter response. This catalogue provides a complete set of reference curves corresponding to every swept parameter combination.

3.5.5 User Interface Workflow

The developed MATLAB GUI integrates the complete workflow for the Two–Stage LC BPF as follows:

- The customer inputs the nominal component values along with the spreading percentage and selects the number of training points.
- The program performs parameter sweeping on the selected components and computes the corresponding magnitude responses over the defined frequency range.
- An Element Matrix and a full Catalogue of frequency responses are automatically generated based on all possible combinations.
- The customer may choose to view a single curve, view all curves, or compare selected curves.

3.5.6 Curve Selection and Modulation Options

After the catalogue of frequency responses has been generated, the user can select any specific curve for detailed inspection. Upon selection, an optional modulation stage may be applied within the MATLAB environment. This modulation capability introduces controlled variations to the chosen frequency response, facilitating the study of non-ideal or perturbed circuit behaviors. The available modulation types are as follows (No Modulation, Cosine Modulation, Sine Modulation and Triangle Modulation).

This approach provides flexibility to simulate practical imperfections and enriches the dataset for neural network training. As a result, the network can accurately predict component values even when the input response deviates from the ideal Two–Stage LC BPF behavior

3.5.7 Neural Networks Training Methodology

For the Fourth-Order (Two-Stage) LC Bandpass Filter, an artificial neural network (ANN) is utilized to predict the values of the inductors and capacitors corresponding to a frequency response selected by the user. Under standard conditions, the ANN is trained on the magnitude response of the chosen curve, allowing it to accurately estimate the component values required to reproduce that response.

If the customer applies modulation or other adjustments to the selected response, the network is trained on the modified data. This ensures that the ANN can reliably determine the component values for both the original and the altered frequency responses, providing consistent and accurate predictions across varying design scenarios (See: Figure 3.17).

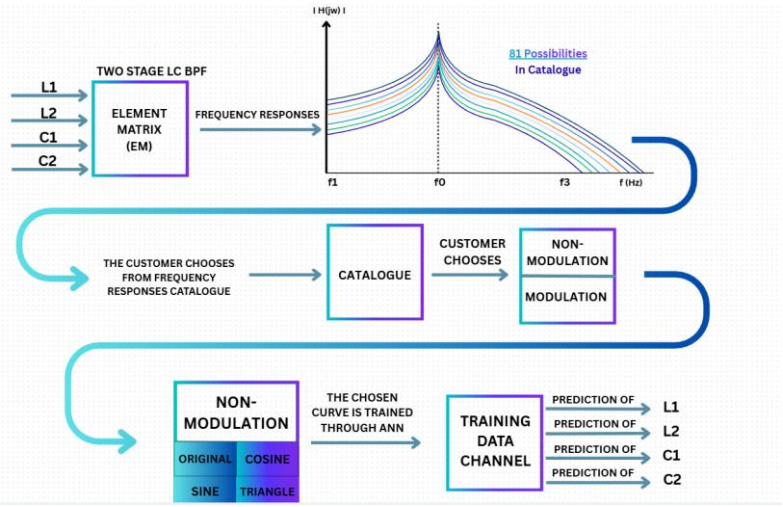


Figure 3.17 Two-Stage LC BPF ANN-Model

3.5.8 ANN Prediction of L_1, L_2, C_1 and C_2

Similarly, only the nominal component values are provided by the user through the GUI, all remaining computations, including frequency responses, catalogue generation, and ANN predictions, are produced directly by MATLAB.

When the user provides an experimental or simulated magnitude response, the trained neural network predicts the corresponding values of $\mathbf{L_1}$, $\mathbf{L_2}$, $\mathbf{C_1}$ and $\mathbf{C_2}$. The predicted outputs are displayed in MATLAB alongside the best-matched curve obtained from the catalogue.

NOTE 1: For non-modulated responses, MSE satisfies,

$$0 < MSE < 1\ (dB^2) \qquad (\mathbf{3.103})$$

NOTE 2: When cosine ripple modulation is applied,

$$MSE > 1\ (dB^2) \qquad (\mathbf{3.104})$$

3.6 GM-C BANDPASS FILTER

3.6.1 Filter Structure, Transfer Function and Magnitude Response

The operational transconductance amplifier is a function that converts an input voltage to an output current. An OTA can be used in many different electronic systems like filters, analog-to-digital converters, and oscillators. It's also the main amplifier in an operational amplifier. The OTA is a key part of many analog circuits (Geiger et. al. 1985).

An ideal operational transconductance amplifier acts like a voltage-controlled current source. It has very high input and output resistance, and its ability to convert voltage into current stays the same. The OTA has two good qualities: its current output can be adjusted by changing an external direct current voltage or current, and it can operate at high frequencies. The OTA is commonly made using CMOS, bipolar, BiCMOS, and GaAs technologies.

In CMOS, the transconductance usually ranges from tens to hundreds of micro-siemens, while in bipolar technology, it can go up to milli siemens. For example, a CMOS OTA usually works in the frequency range of 50 MHz to several hundred MHz Special techniques help the OTA work well with input voltages of several volts (Deliyannis, 1999). The symbol of a single OTA is shown in Figure 3.18.

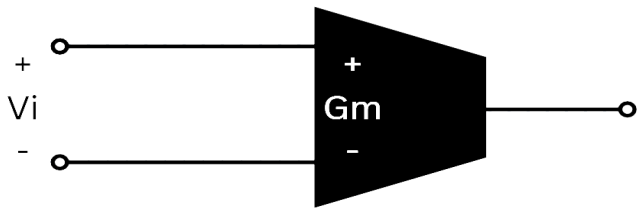


Figure 3.18 OTA Symbol

The output current from an OTA depends on the difference between the two input voltages. The way the input voltages relate to the output current is described by,

$$I_0 = G_m\,(V_i^+ - V_i^-) \qquad (3.105)$$

However, Gm-C filter represents a kind of continuous-time analog filter that employs transconductance amplifiers Gm along with capacitors C to perform filtering operations such as low-pass, high-pass, **bandpass**, or band-stop filters.

Transconductance (expressed in siemens, S) represents the gain of a transconductance amplifier, and capacitors employed to integrate the current, affecting the frequency response of the circuit. The Gm-C filter employs voltage-controlled current sources (the Gm blocks) in place of conventional resistors.

These current sources supply current to capacitors, and the relationship between current and voltage (through the capacitors) produces the intended filter response. If the Gm is linked with the capacitor, then, it's called an integrator, and if not, then, it is called a resistor. The general form of the Gm-C diagram is shown below (See: Figure 3.19).

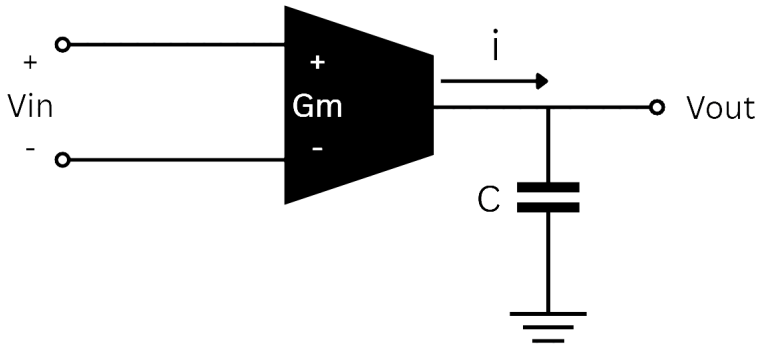


Figure 3.19 Gm-C Circuit

The output current of the OTA is the transconductance multiplied with the input voltage.

$$I_0 = G_m V_1 \qquad (3.106)$$

$$V_0 = ZI_0 = \frac{G_m}{sC}\,(V_1) \qquad (3.107)$$

Specifically, observe the BiCMOS second-order bandpass filter design (See: Figure 3.20), whose circuit topology and design methodology are consistent with previously reported BiCMOS implementations (Gaspariano et al., 2003).

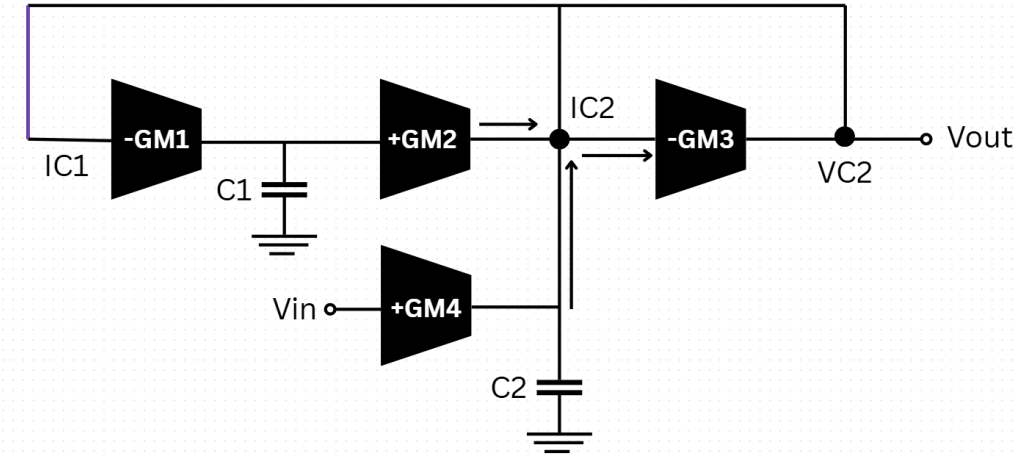


Figure 3.20 Second Order Gm-C BPF Circuit

This configuration illustrates that Gm_1 and Gm_2 form an integrator with C_1. However, Gm_4 with C_2 controls the shape of the output. Furthermore, Gm_3 controls the bandwidth β. In order to derive the transfer function H(s), the current equations at the capacitors and their voltages must be identified as,

$$\mathrm{I_{C1}} = -\mathrm{G_{m1}V_{C2}} \tag{3.108}$$

$$\mathrm{V_{C1}} = \mathrm{ZI} = \frac{1}{\mathrm{sC_1}}(\mathrm{I_{C1}}) = \frac{-\mathrm{G_{m1}V_{C2}}}{\mathrm{sC_1}} \tag{3.109}$$

$$\mathrm{I_{C2}} = \mathrm{G_{m2}V_{C1}} - \mathrm{G_{m3}V_{C2}} + \mathrm{G_{m4}V_{in}} \tag{3.110}$$

$$\mathrm{V_{C2}} = \mathrm{ZI} = \frac{1}{\mathrm{sC_2}}(\mathrm{I_{C2}}) = \frac{\mathrm{G_{m2}V_{C1}} - \mathrm{G_{m3}V_{C2}} + \mathrm{G_{m4}V_{in}}}{\mathrm{sC_2}} \tag{3.111}$$

As $\mathrm{V_{C2}} = \mathrm{V_{out}}$, substituting in equation (**3.111**) results in,

$$V_{out} = \frac{G_{m2}\left(\frac{-G_{m1}}{sC_1}\right)V_{out} - G_{m3}V_{out} + G_{m4}V_{in}}{sC_2} \quad (\mathbf{3.112})$$

Rearrange equation (**3.112**) in order to reveal H(s),

$$V_{out}(s^2C_1C_2 + sC_1G_{m3} + G_{m1}G_{m2}) = sC_1G_{m4}V_{in} \quad (\mathbf{3.113})$$

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{sC_1G_{m4}}{s^2C_1C_2 + sC_1G_{m3} + G_{m1}G_{m2}} \quad (\mathbf{3.114})$$

Normalizing H(s),

$$H(s) = \frac{Ks\left(\frac{G_{m4}}{C_2}\right)}{s^2 + s\left(\frac{G_{m3}}{C_2}\right) + \left(\frac{G_{m1}G_{m2}}{C_1C_2}\right)} \quad (\mathbf{3.115})$$

where the center frequency ω_0, bandwidth β and the quality Q of the H(s) are,

$$\omega_0 = \sqrt{\frac{G_{m1}G_{m2}}{C_1C_2}}, \qquad \beta = \frac{G_{m3}}{C_2}, \qquad Q = \frac{\omega_0}{\beta} \quad (\mathbf{3.116})$$

Translating the H(s) to H($j\omega$),

$$H(j\omega) = \frac{(j\omega)\left(\frac{G_{m4}}{C_2}\right)}{(j\omega)^2 + (j\omega)\left(\frac{G_{m3}}{C_2}\right) + \left(\frac{G_{m1}G_{m2}}{C_1C_2}\right)} = \frac{j\omega\left(\frac{G_{m4}}{C_2}\right)}{-\omega^2 + (j\omega)(\beta) + ({\omega_0}^2)} \quad (\mathbf{3.117})$$

The magnitude response is defined as,

$$|H(j\omega)| = \frac{\omega^2\left(\frac{G_{m4}}{C_2}\right)}{\sqrt{(\omega_0^2 - \omega^2)^2 + (\beta\omega)^2}} \quad (\mathbf{3.118})$$

3.6.2 MATLAB and Proteus Simulation Analysis

Assume the capacitor values to be $C_1 = C_2 = 10nF$, the transconductance values $Gm_1 = Gm_2 = 0.628mS$, $Gm_3 = 0.126mS$, and $Gm_4 = 39.6mS$. The table below arranges all components, and their calculations accurately (See: Table 3.4).

Table 3.4 Gm-C BPF Components and Frequency Response

Circuit Components						
Gm_1	Gm_2	Gm_3	Gm_4	C_1	C_1	
$0.628mS$	$0.628mS$	$0.126mS$	$39.6mS$	100nF	100nF	
At −3dB, Center Frequency in rad/s						
ω_0		ω_{c1}		ω_{c2}		β
62800 rad/s		56722.27 rad/s		69317.35 rad/s		12595.08
At −3dB, Center Frequency in Hz						
f_0		f_{c1}		f_{c2}		β
9994.93 Hz		9027.63 Hz		11032.2 Hz		2004.57

The center frequency f_0 = 10kHz, and bandwidth β turns to be 2.004kHz, However, the quality factor Q results ≅ 5 (See: Figure 3.21). The Gain doesn't affect the values of the center frequency and bandwidth, only increases the signal efficiency gain.

The capacitor values decide the sharpness of the signal smoothness. This circuit offers the benefit that both the center frequency and the quality factor of the bandpass filter can be modified by selecting suitable capacitors. The schematic diagram of the operational amplifier is displayed to learn the behavior and how it will act (See: Figure 3.22).

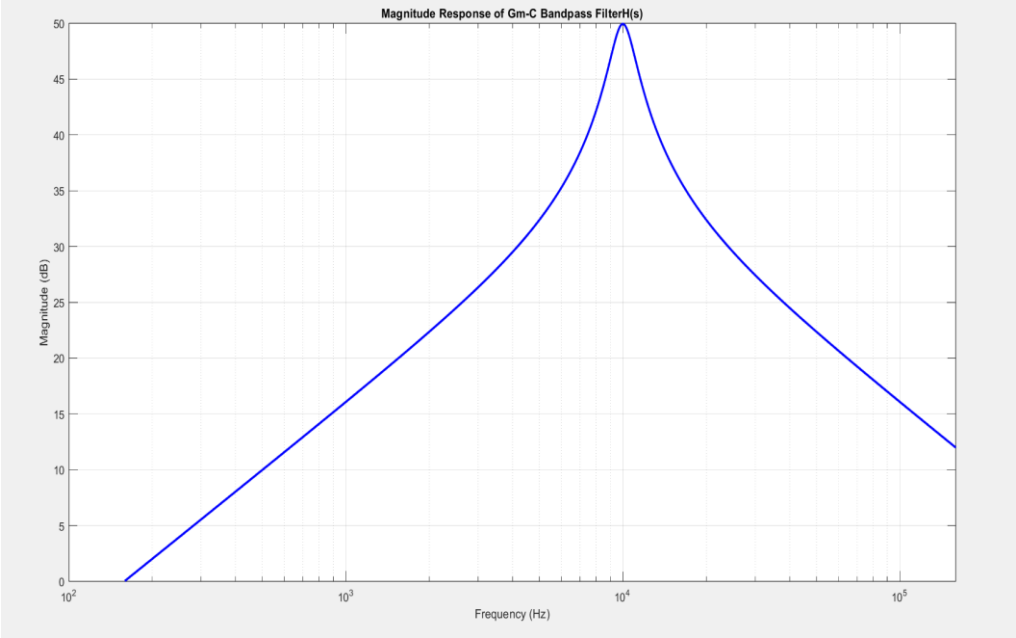


Figure 3.21 Gm-C BPF Magnitude Response

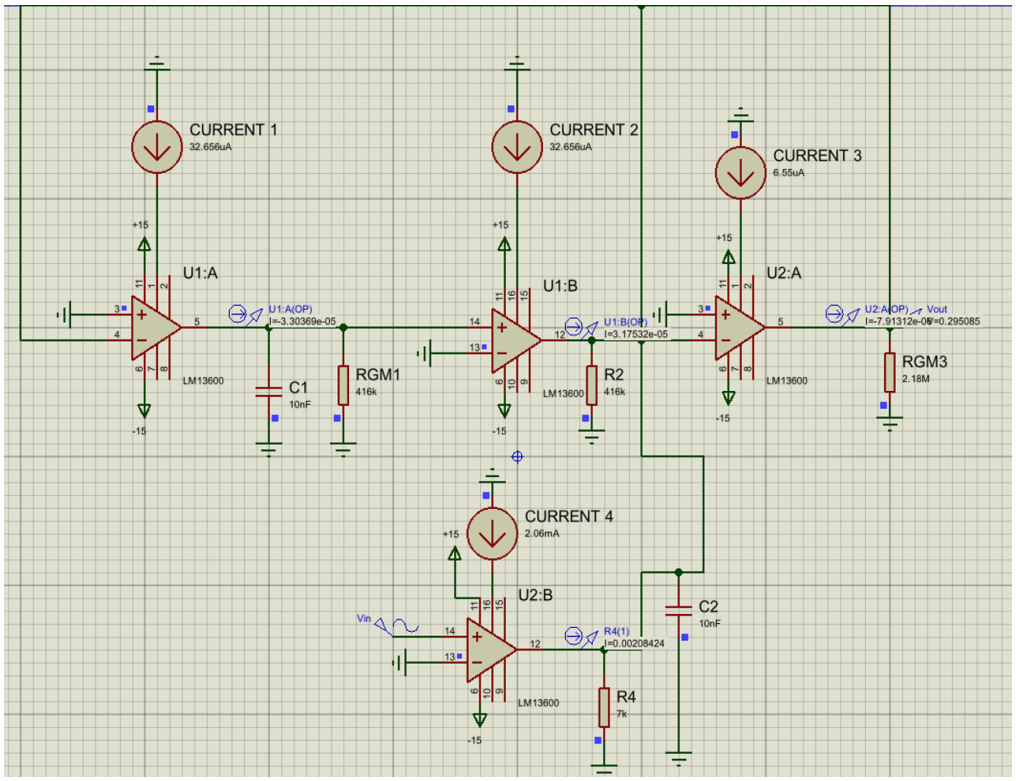


Figure 3.22 Gm-C BPF Proteus Schematic Diagram

To Prove the OTA values that used in MATLAB, two equations are used to verify calculated values,

$$G_m = \frac{I_{ABC}}{2V_T} \tag{3.119}$$

$$R_{BIAS} = \frac{V^+ - 2V_{BE}}{I_{ABC}} \tag{3.120}$$

The above equations were valid for several decades of bias current until leakage current limits the linearity of G_m (Ali and Abdaljaber, 2017).

With the use of the equations above, accurate values are implemented to give the desired bandpass filter response, which is exactly similar to the shown MATLAB, except that the gain dropped to ≅ 10% causing by non-idealities of the used OTA (See: Figure 3.23).

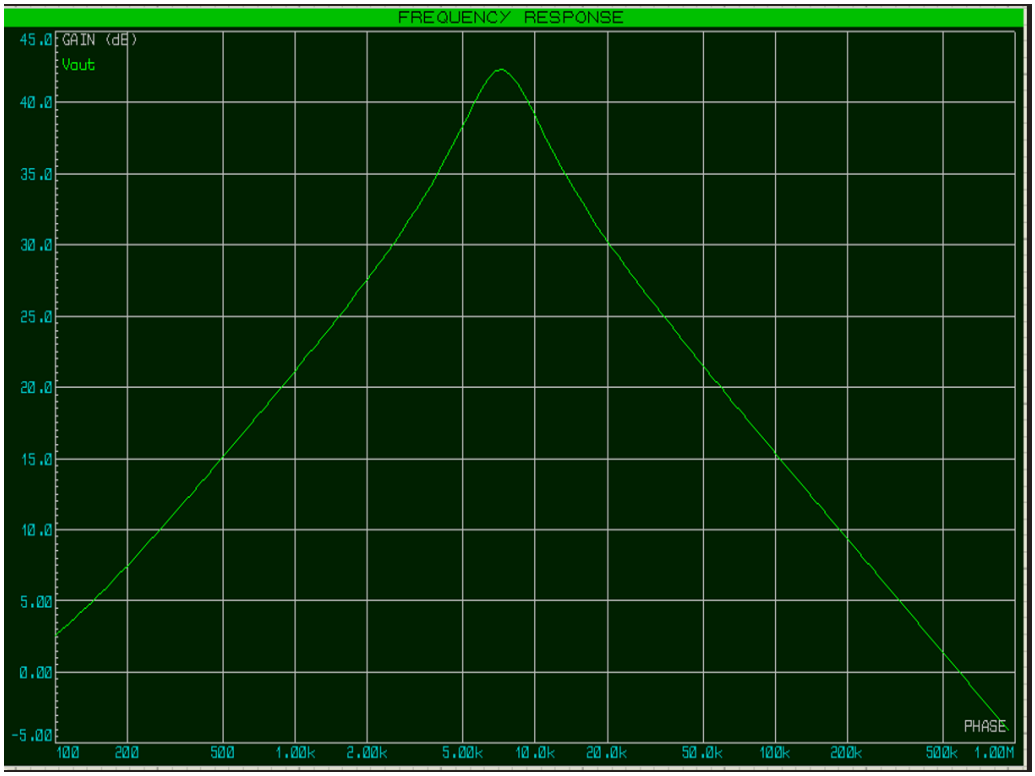


Figure 3.23 Gm-C BPF Proteus Magnitude Response

3.6.3 Parameter Sweeping of C_1 and C_2

To construct a comprehensive dataset for $\mathbf{C_1}$ **and** $\mathbf{C_2}$ parameters are swept within customer-specified limits. The customer selects the number of testing points (N_t), which defines the density of the sweep. Selecting $N_t = 3$ results in $3 \times 3 = 9$ parameter combinations. The capacitor values are uniformly distributed over their respective ranges:

$$C_{1,i} = \in [C_{min}, C_{max}], \quad C_{2,i} = \in [C_{min}, C_{max}] \qquad (\mathbf{3.121})$$

where i = (1,2,3……N_t)

For each pair ($L_{1i}, L_{2i}, C_{1i}, C_{2i}$), a frequency vector is generated to compute the magnitude response of the filter **(EM)**. This approach ensures uniform coverage of the design space and produces data suitable for building a reliable catalogue.

3.6.4 Generation of Frequency Response Catalogue

Each (C_{1i}, C_{2i}) combination generates a distinct magnitude response over the specified frequency range. All computed responses are compiled into a catalogue, forming a structured matrix whose dimensions are given by:

$$\text{Catalogue Size} = N_t^2 \times N_f \qquad (\mathbf{3.122})$$

where N_f represents the number of sampled frequency points used to evaluate the filter response. This catalogue provides a complete set of reference curves corresponding to every swept parameter combination.

3.6.5 User Interface Workflow

The developed MATLAB GUI integrates the complete workflow as follows:

- The customer inputs the nominal component values along with the spreading percentage and selects the number of training points.

- The program performs parameter sweeping on the selected components and computes the corresponding magnitude responses over the defined frequency range.
- An Element Matrix and a full Catalogue of frequency responses are automatically generated based on all possible combinations.
- The customer may choose to view a single curve, view all curves, or compare selected curves.

3.6.6 Curve Selection and Modulation Options

After the catalogue of frequency responses has been generated, the user can select any specific curve for detailed inspection. Upon selection, an optional modulation stage may be applied within the MATLAB environment. This modulation capability introduces controlled variations to the chosen frequency response, facilitating the study of non-ideal or perturbed circuit behaviors.

This approach provides flexibility to simulate practical imperfections and enriches the dataset for neural network training. As a result, the network can accurately predict component values even when the input response deviates from the ideal Two–Stage LC BPF behavior

3.6.7 Neural Networks Training Methodology

For the Gm-C BPF, ANN is utilized to predict the values of the inductors and capacitors corresponding to a frequency response selected by the user. Under standard conditions, the ANN is trained on the magnitude response of the chosen curve, allowing it to accurately estimate the component values required to reproduce that response.

If the customer applies modulation or other adjustments to the selected response, the network is trained on the modified data. This ensures that the ANN can reliably determine the component values for both the original and the altered frequency responses, providing consistent and accurate predictions across varying design scenarios (See: Figure 3.24).

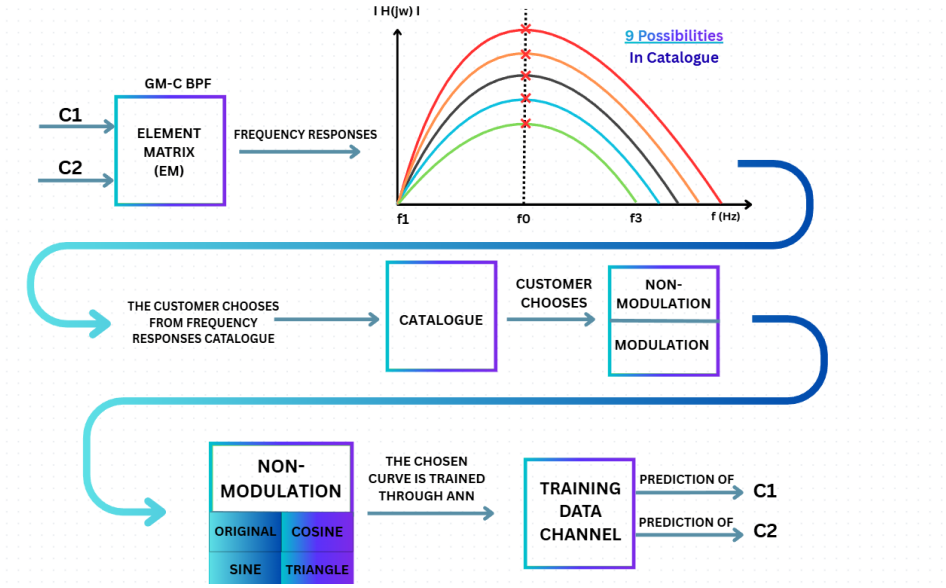


Figure 3.24 Gm-C BPF ANN-Model

3.6.8 ANN Prediction of C_1 and C_2

Similarly, only the nominal component values are provided by the user through the GUI, all remaining computations, including frequency responses, catalogue generation, and ANN predictions, are produced directly by MATLAB.

When the user provides an experimental or simulated magnitude response, the trained neural network predicts the corresponding values of $\mathbf{C_1}$ **and** $\mathbf{C_2}$**.** The predicted outputs are displayed in MATLAB alongside the best-matched curve obtained from the catalogue.

NOTE 1: For non-modulated responses, MSE satisfies,

$$0 < MSE < 1\ (dB^2) \qquad (\mathbf{3.123})$$

NOTE 2: When cosine ripple modulation is applied,

$$MSE > 1\ (dB^2) \qquad (\mathbf{3.124})$$

3.7 MOSFET N-DEPLETION TYPE

MOSFET designs have many applications and types. For instance, as analog circuit design, it can be common-source, common-drain and common-gate. Each design type can be either N-Channel or P-Channel (See: Figure 3.25). The table below will show the difference between the channels (See: Table 3.5)

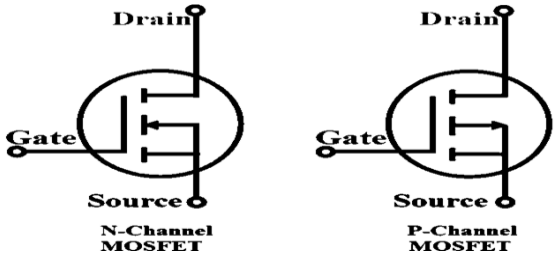


Figure 3.25 MOSFET Types

Table 3.5 N-Channel and P-Channel Comparison

N-Channel	**P-Channel**
Negative electrons charge	Positive Charge
$\mathbf{V_{GS} > V_{th}}$	$V_{GS} < V_{th}$
Faster	Slower
Arrow out of Source	Arrow into Source
Current from Drain to Source	Current from Source to Drain

A depletion-mode N-channel MOSFET is a field-effect transistor that is inherently conductive, it permits current to pass between the drain and source terminals even when no voltage is applied to the gate. This intrinsic "on" condition distinguishes it from enhancement-mode MOSFETs, which need a positive gate-to-source voltage to allow conduction. In a depletion-mode NMOS, applying a negative voltage to the gate decreases the number of free electrons in the conductive channel, effectively constricting (Harrison).

The DC calculation comes first. Then, the circuit will be translated into AC. DC values depend on AC in order to get a solution. In DC, capacitors

function as open circuit (See: Figure 3.26).

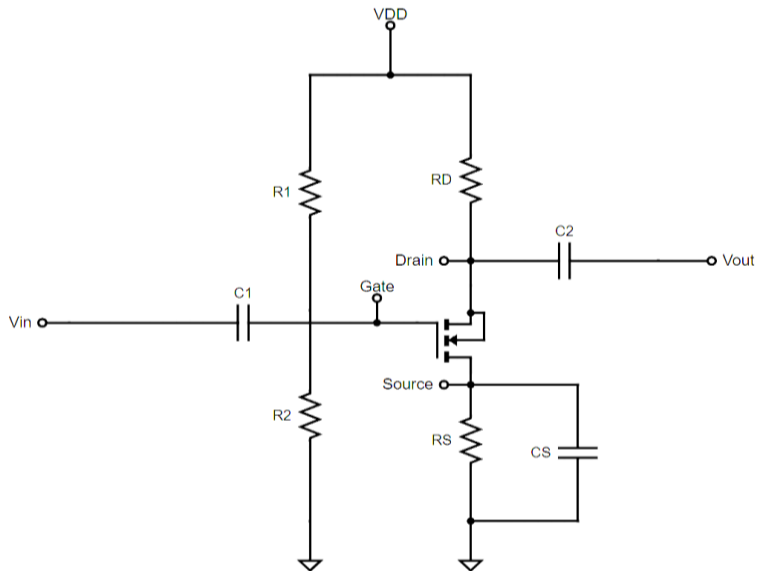


Figure 3.26 DC MOSFET Circuit

As solving for DC, despite the parameter values. V_{GS} , I_D and V_{DS} are important. After finding out, a Load-line equation is used to make voltage-current relation. However, V_{GSQ} and I_{DQ} are detected using Load-line graph (See: Figure 3.27). The load-line equations are written this way,

$$V_{GS} = V_G - R_S I_D \qquad (\mathbf{3.125})$$

The drain-to-source voltage loop is written as,

$$V_{DS} = V_{DD} - R_D I_D - R_s I_D \qquad (\mathbf{3.126})$$

By solving the circuit parameters, their corresponding values can be plotted to visualize their relationships. At a specific operating point, the voltage intersects the load-line, defining the quiescent point (Q-point), which represents the transistor's operating condition in the saturation region.

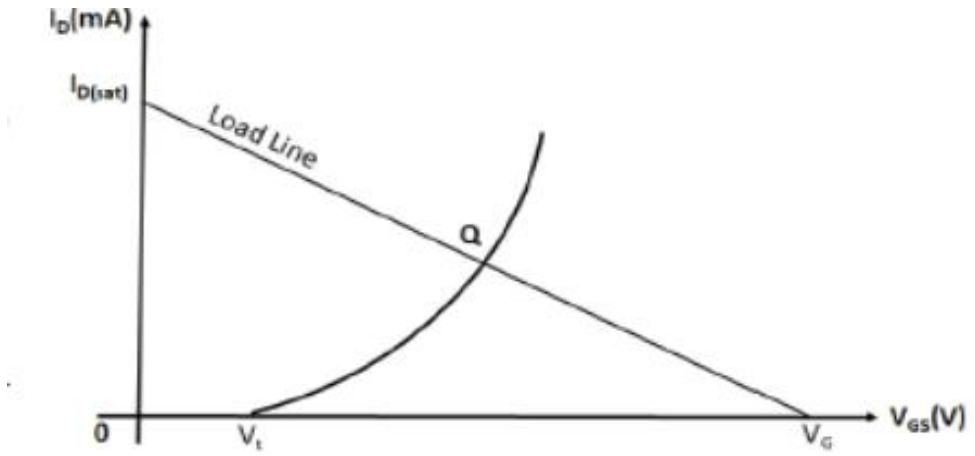


Figure 3.27 Load-line Relation

When V_{GSQ} and I_{DQ} are determined, AC starts due to DC dependency. This time all capacitors act as a short circuit. However, the circuit will be converted to AC equivalent model. This transformation adds an extra resistor r_d and a dependent current source. The value of the added resistor depends on the given (g_{os}),

$$r_d = \frac{1}{(g_{os})} \tag{3.127}$$

$$g_{m0} = \frac{2I_{DSS}}{|V_P|} \tag{3.128}$$

where g_{m0} is the zero-bias transconductance, I_{DSS} is the drain-source saturation current, V_P is the Pinch-off voltage.

The transconductance is expressed as,

$$g_m = g_{m0}\left(1 - \frac{V_{GSQ}}{V_P}\right) \tag{3.129}$$

Furthermore, the impedance characteristics significantly influence the overall circuit behavior. Each circuit configuration requires distinct analytical expressions to determine its input and output impedances. In this particular case,

the corresponding equations for the input and output impedances are expressed as follows (See: Figure 3.28).

$$Z_{Input} = \frac{R_1 R_2}{R_1 + R_2} \quad (\mathbf{3.130})$$

$$Z_{Output} = \frac{R_D\, r_r}{R_D + r_d} \quad (\mathbf{3.131})$$

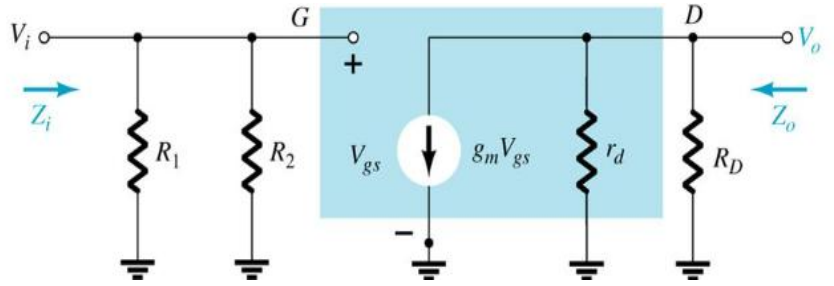


Figure 3.28 AC Equivalent Model

Lastly, the gain of the N-Channel MOSFET design can be easily found at this step. In The MOSFET design, for instance, predicting the current I_D is easy for behaviors, but it is much harder for physical features.

$$A_v = -G_m R_D \quad (\mathbf{3.132})$$

Using Figure 3.26, assume $R_1 = 110\text{M}\Omega, R_2 = 10\text{M}\Omega, R_D = 1.8\text{k}\Omega, R_S = 750\Omega$, $I_{DSS} = 6\text{mA}, V_P = -3\text{V}, V_{CC} = 18\text{V}$. Firstly in DC part, the I_{DQ}, and V_{GSQ} point are obtained to continue to AC (See: Figure 3.29).

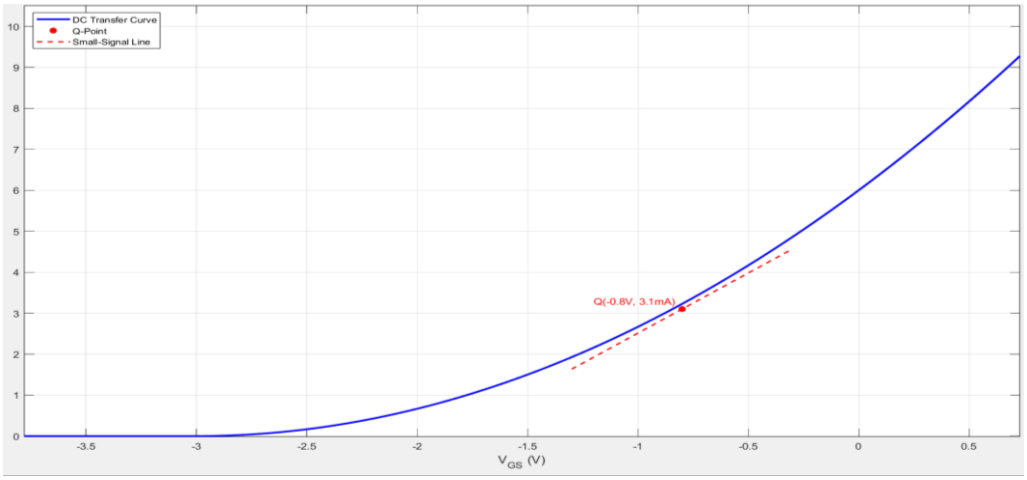


Figure 3.29 Q-Points Line

I_{DQ} and V_{GSQ} points are chosen to be 3.1mA and $-0.8V$ from the graph. $g_{m0} = 4mS$, $g_m = 2.933mS$, $r_d = 100k\Omega$. Consequently, the input and output impedance are calculated, $Z_{Input} = 9.17M\Omega$ and $Z_{Output} \cong 1.8k\Omega$. Lastly, the gain $A_v = 5.22$. The oscilloscope in proteus program shows the input and output voltages signals (See: Figure 3.30).

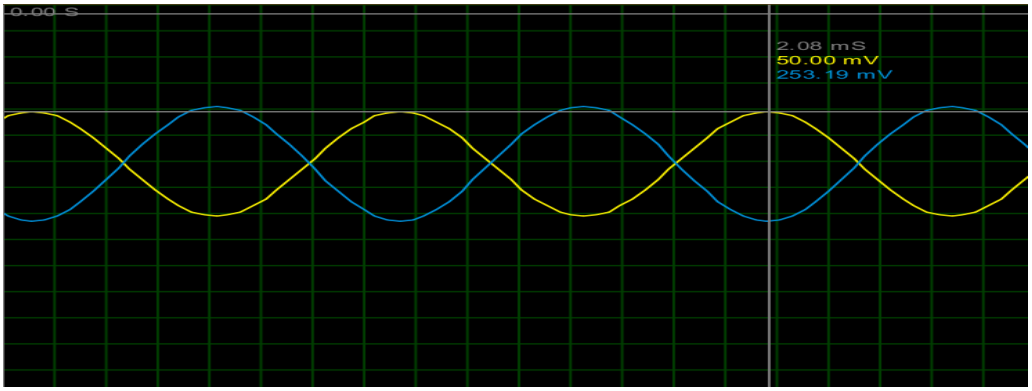


Figure 3.30 Oscilloscope N-Depletion Type Signal

With the help of neural networks, the decision of the inputs and the outputs of the electrical parameters are keen. In order to predict the drain current I_D, then, the inputs must contain V_{GS} and V_{DS}. Those voltages will predict the drain

current using Feedforward Propagation. (See: Figure 3.31 and Figure 3.32).

This methodology enables the network to capture complex, nonlinear behaviors that are often difficult to model using conventional analytical techniques. By training on a comprehensive dataset, the neural network can generalize across various biasing conditions and device operating regions.

Moreover, neural networks demonstrate the capability to model second-order effects, such as channel length modulation and velocity saturation, which are typically neglected in first-order models. Their adaptability makes them particularly useful for modeling devices across different technological nodes without requiring substantial modifications to the model structure. The resulting models are not only computationally efficient but also suitable for integration into modern circuit simulators. Consequently, neural network-based modeling has emerged as a promising tool for enhancing accuracy in device characterization and circuit-level simulation.

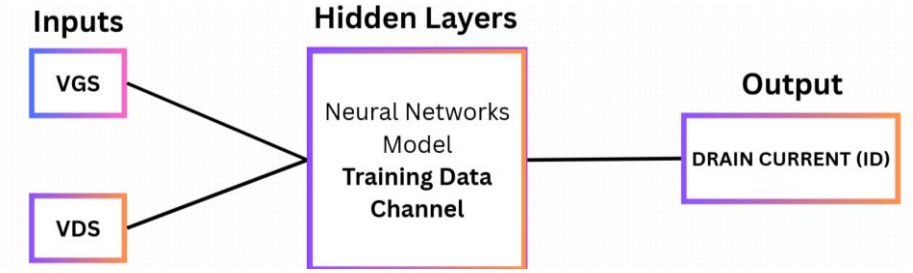


Figure 3.31 N-Channel MOSFET ANN-Model

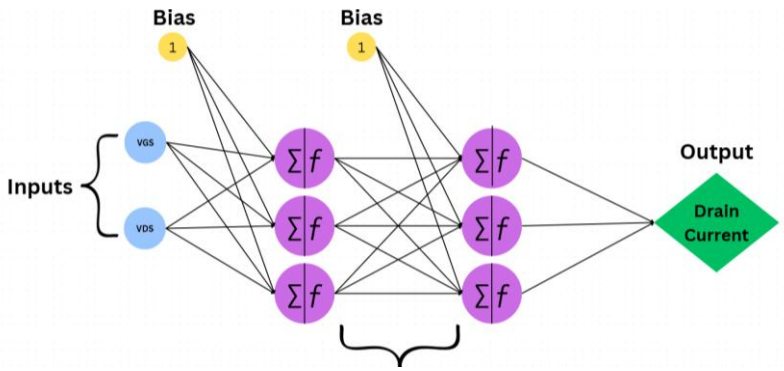


Figure 3.32 N-Channel MOSFET Hidden Layers Observation

3.8 RC Lowpass Filters Analysis Using ERT

3.8.1 Monte Carlo Randomization

Monte Carlo randomization is employed to generate the training dataset for the RC lowpass filter model. Instead of using deterministic element spreading, the resistor and capacitor values were randomly sampled from uniform distributions bounded by their respective tolerance limits. For each Monte Carlo trial, a unique (R,C) pair was drawn and used to compute the corresponding frequency response of the filter (Thakur, 2016).

This process was repeated for thousands of trials, producing a statistically diverse dataset that reflects realistic component variations. The resulting dataset was used to train an ANN to predict the component values from the observed magnitude response. The Monte Carlo approach significantly enhances dataset diversity, improves neural network generalization, and captures real-world tolerance behavior (Raychaudhuri, 2010).

Circuit parameters are randomly sampled from uniform distributions constrained by specified tolerance bounds, and the resulting data are utilized to train a supervised neural network for **Inverse Modeling Estimation**.

$$x_i = |H_i(f)| \rightarrow y_i = |R_i, C_i| \tag{3.133}$$

$$|H(f)| \rightarrow |R, C| \tag{3.134}$$

Inverse modeling is a data-driven or analytical approach in which the objective is to estimate unknown system parameters or inputs by observing the system outputs, effectively learning or computing the inverse relationship of the underlying forward model (Rajesh et al., 2015).

3.8.2 Monte Carlo Methodology

This presents the methodology adopted for developing the RC lowpass filter modeling framework based on Monte Carlo randomization ANN prediction (See: Figure 3.33). The procedure consists of four main stages:

- Defining the circuit and its nominal parameters.
- Generating randomized component values through a Monte Carlo process.
- Computing the corresponding frequency responses for each trial.
- Training the neural network to predict the resistor and capacitor values from the measured or simulated magnitude response.
- The entire workflow is implemented in MATLAB.

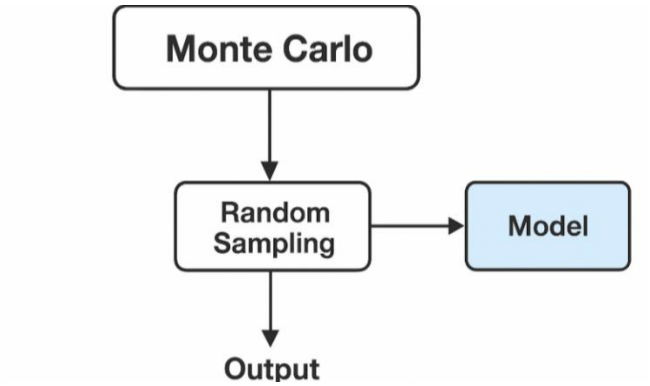


Figure 3.33 Monte Carlo–Based Random Parameter Sampling Framework

3.8.3 User Interface Workflow

The complete workflow implemented in MATLAB is summarized as follows:

- ✓ User selects nominal R and C values.
- ✓ User selects tolerance percentage(s).
- ✓ The program computes lower and upper bounds.
- ✓ Monte Carlo random values are generated for both components.
- ✓ For each randomized RC pair, the magnitude response is computed.
- ✓ The database is assembled from all Monte Carlo trials.
- ✓ A neural network is trained on this dataset.
- ✓ When the user provides an experimental or simulated response, the ANN predicts the corresponding R and C.

CHAPTER 4

4. RESULTS AND DISCUSSION

4.1 GRAPHICAL USER-INTERFACE (GUI) FRAMEWORK

To support practical use of the proposed approach, a dedicated User Interface (UI) was developed. UI allows users to explore available circuit topologies, input their desired specifications, preview system responses, and obtain optimized component values predicted by the trained Neural Networks.

Figure 4.1 illustrates the overall structure of the UI using flow chart, including the main operational layers and the supported circuit categories such as low-pass filters, band-pass filters, and MOSFET-based designs.

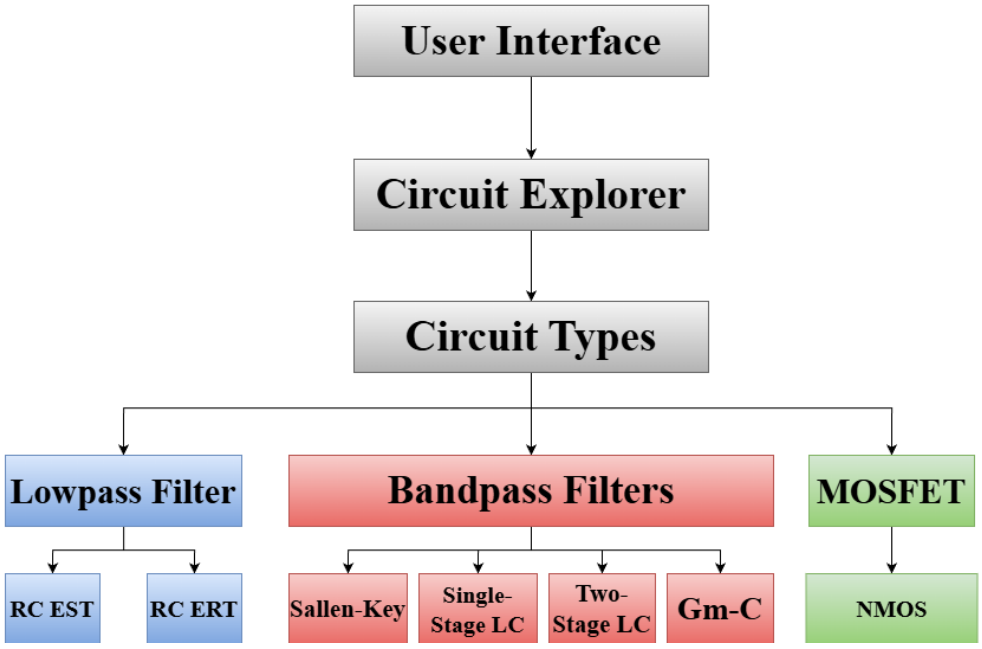


Figure 4.1 User Interface Chart

Once the user selects a circuit type and enters the required specifications, the UI generates all valid element combinations and displays their corresponding frequency responses. The user may then inspect these responses individually or compare them through the catalogue feature. Figure 4.2 provides a detailed flowchart of this process—from the initial design selection to the final stage where NN training begins and the predicted characteristics, error matrices, and optimized circuit values are produced.

In the following sections, numerical examples are provided for each circuit category. These include:

- Input specifications used for training and testing,
- Input specifications used for training and testing,
- Predicted circuit responses generated by the UI and the Neural Networks,
- Comparison between predicted and reference (target) responses,
- Performance metrics such as Mean Squared Error (MSE), prediction accuracy, and computational cost.

These results collectively demonstrate the effectiveness of integrating machine-learning-based optimization with an interactive user interface for automated circuit design. Furthermore, the numerical evaluations confirm that the proposed Neural Network models are capable of generalizing across a wide range of design specifications.

User Interface does not only accelerates the design process, but also provides users with a transparent view of intermediate computations and model predictions. This combination of automation and interpretability enhances the reliability of the design workflow, particularly for complex circuit topologies. Overall, the presented results validate the practicality of the system and establish a foundation for future improvements and potential real-time hardware implementation.

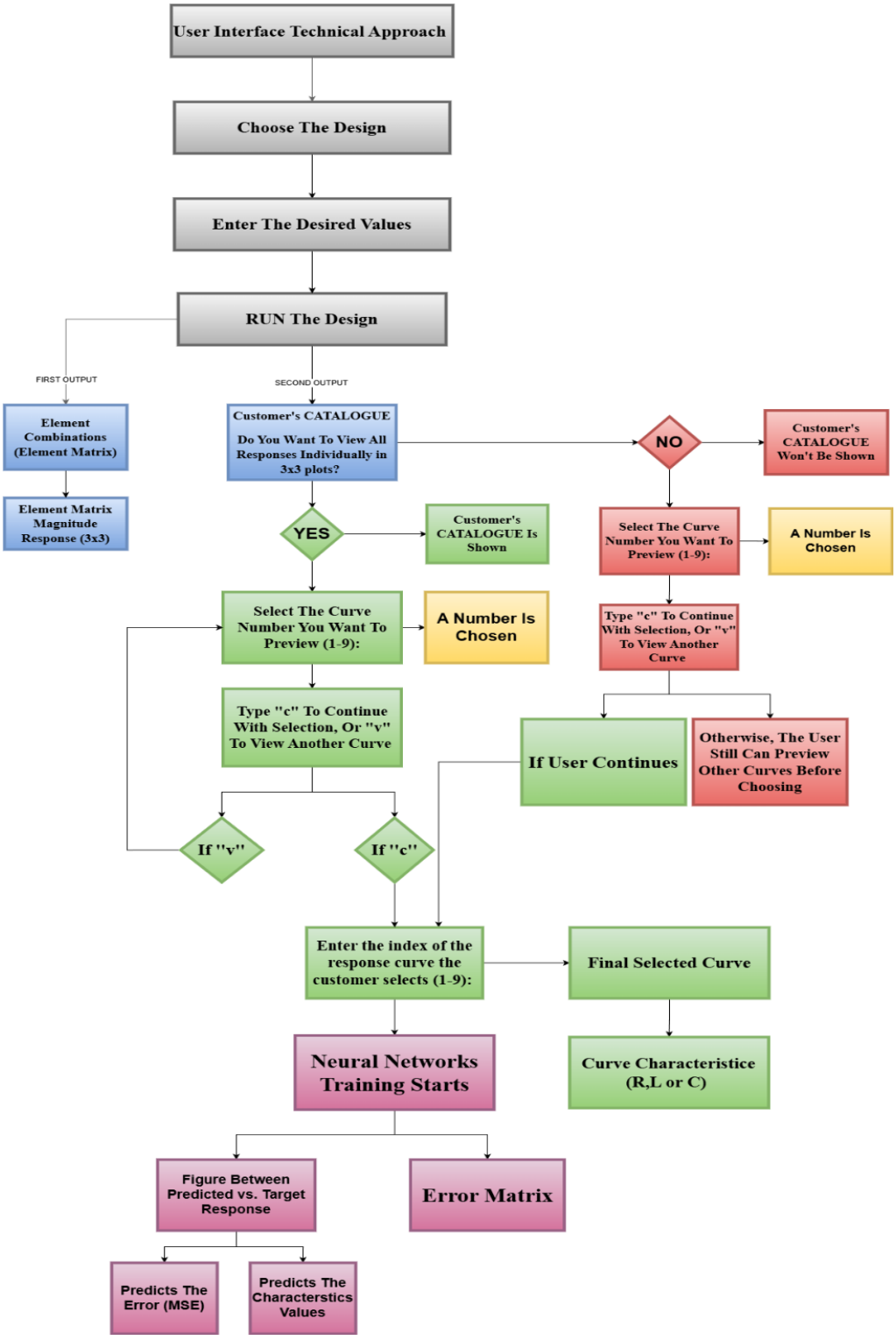


Figure 4.2 User Flow Chart

In addition to the core workflow described earlier, an extended post-processing stage was implemented in MATLAB to introduce controlled modulation into the selected frequency response. While the UI is responsible for collecting user inputs. The modulation procedure is executed entirely within the MATLAB environment after the user finalizes a specific curve. This stage allows the selected response to be altered using predefined ripple functions, including cosine-, sine-, and triangle-based variations, with user-specified amplitudes and phase ranges. The purpose of this modulation layer is to simulate non-idealities that commonly appear in practical circuits, such as minor oscillatory disturbances, component imperfections, or shaping effects used in signal conditioning.

By enriching the target response with mathematically controlled distortions, the resulting data set provides a more challenging and diverse training environment for the Neural Network models. The inclusion of modulation results in updated performance analyses, where the NN predictions are compared not only against ideal curves but also against these modified targets. This contributes to more comprehensive error matrices, improved understanding of generalization behavior, and a clearer evaluation of model robustness when handling perturbed frequency characteristics. Ultimately, this MATLAB-based modulation mechanism strengthens the realism and adaptability of the proposed design framework, ensuring that the system remains effective even when applied to circuit responses that deviate from textbook ideal behavior (See: Figure 4.3).

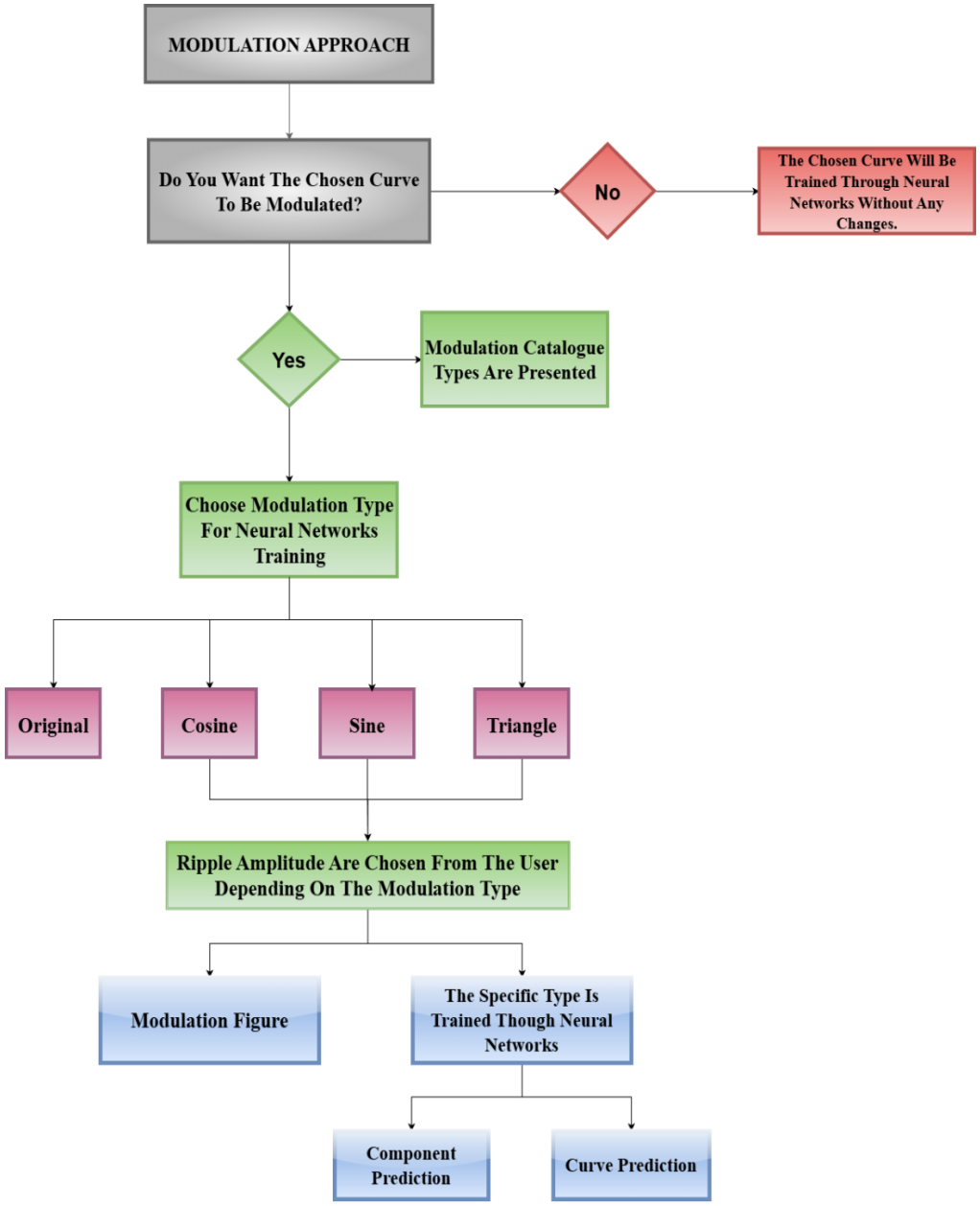


Figure 4.3 Modulation Flow Chart

4.2 NUMERICAL RESULTS

This chapter presents the numerical results obtained from the developed circuit-design system. Using the modeling framework described in Chapter 3, all circuit parameters, design variables, and response data were compiled to form the datasets used for ANN training and performance evaluation.

4.2.1 RC Lowpass Filter

The circuit parameters are input through the user interface, allowing the user to freely specify any desired values (See: Figure 4.4).

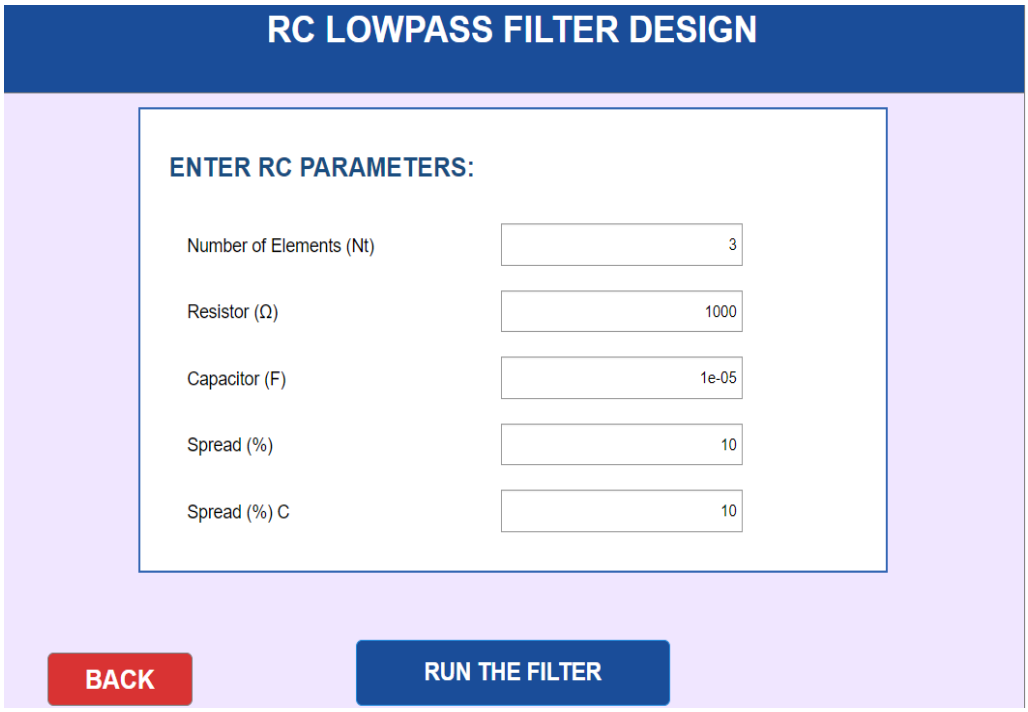


Figure 4.4 RC LPF GUI

Continuing with the specified values of the circuit, RC LPF frequency response provides **9 lines**. The table below shows the sweeping parameters with their cut-off frequencies as **EM** (See: Table 4.1).

Table 4.1 RC LPF Element Matrix

R (kΩ)	C (µF)	$f_0\ rad/s$
0.9	9µF	19.649
0.9	10µF	17.684
0.9	11µF	16.076
1	9µF	17.684
1	10µF	15.915
1	11µF	14.469
1.1	9µF	16.076
1.1	10µF	14.469
1.1	11µF	13.153

When the EM is shown, the magnitude responses of the filter are shown (See: Figure 4.5). However, the customer needs to see the catalogue for this filter before choosing his suitable curve (See: Figure 4.6).

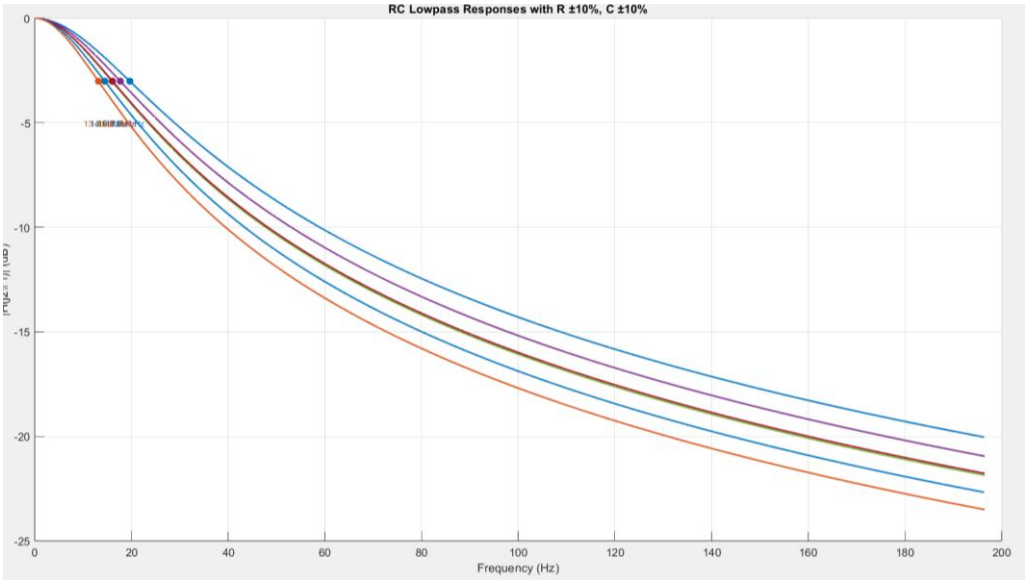


Figure 4.5 RC LPF Magnitude Responses

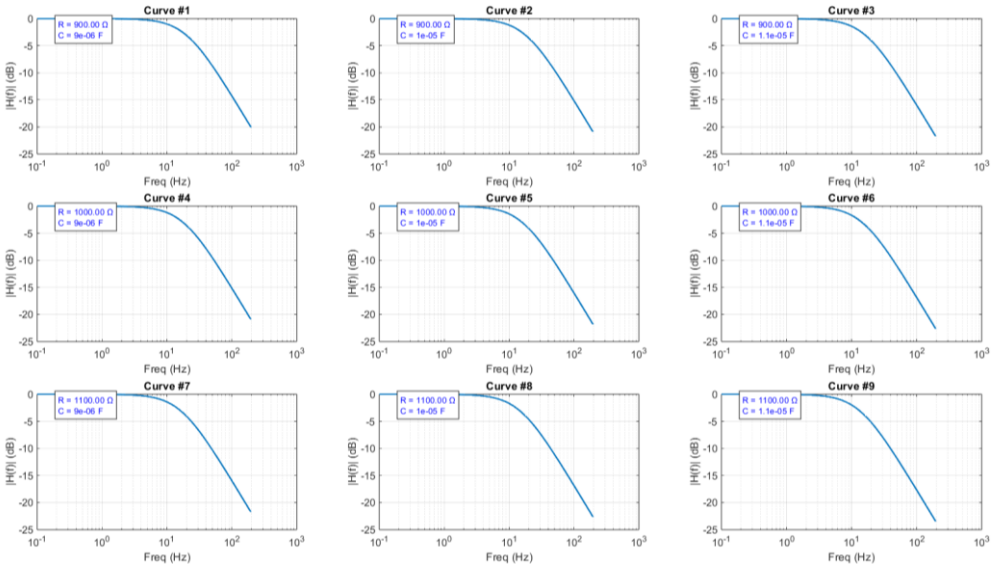
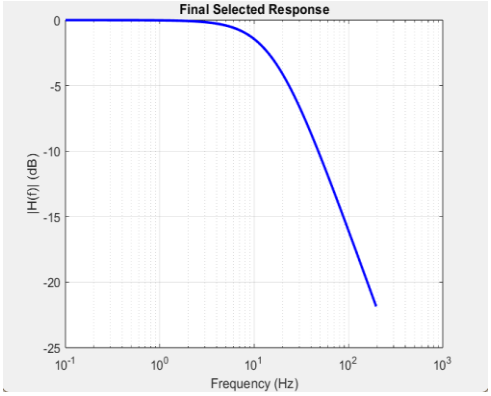


Figure 4.6 RC LPF Catalogue

At this point, the customer may preview and observe all the curves, and after previewing the curves, the customer finally selects suitable curve. When the customer selects final curve, curve's characteristics are given. Assume the customer chose the fifth (5^{th}) curve (See: Figure 4.7).



Customer Selected Curve #5:

R = 1.000000 kΩ

C = 10.000000 µF

Figure 4.7 Customer's Final Selection

After the customer chooses the curve, the modulation option plays a role right now. If the customer doesn't need any modulation of the chosen curve, then, the chosen curve will be trained through Neural Networks to obtain best

values of the curve, but if the customer wants modulation, then, the modulated curve is trained though NN to obtain optimized values of the modulated curve.

Case 1: If the customer chooses No Modulation, predicted values, predicted curve and error matrix are shown (See: Figure 4.8).

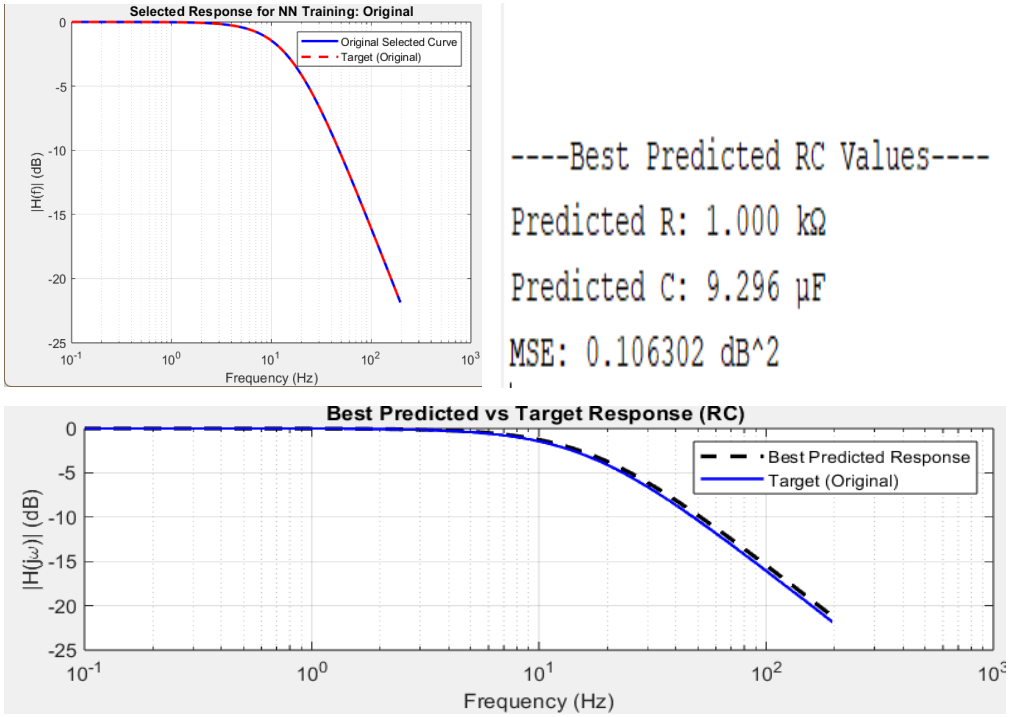


Figure 4.8 RC LPF No Modulation Result

The Error Matrix of R and C can be shown also be shown below (See: Figure 4.9).

R_True	R_Pred	R_Error	R_Error_percent	C_True	C_Pred	C_Error	C_Error_percent
900	969.04	69.038	7.6709	9e-06	7.4733e-06	-1.5267e-06	-16.963
900	931.88	31.876	3.5418	1e-05	7.9562e-06	-2.0438e-06	-20.438
900	992.84	92.835	10.315	1.1e-05	9.0631e-06	-1.9369e-06	-17.608
1000	931.88	-68.124	-6.8124	9e-06	7.9562e-06	-1.0438e-06	-11.598
1000	999.65	-0.34945	-0.034945	1e-05	9.2962e-06	-7.0381e-07	-7.0381
1000	1024.9	24.928	2.4928	1.1e-05	1.1667e-05	6.6705e-07	6.0641
1100	992.84	-107.16	-9.7423	9e-06	9.0631e-06	6.3143e-08	0.70159
1100	1024.9	-75.072	-6.8247	1e-05	1.1667e-05	1.667e-06	16.67
1100	1063	-37.034	-3.3668	1.1e-05	1.2772e-05	1.7718e-06	16.107

Figure 4.9 No Modulation Error Matrix

Case 2: If the customer wants to modulate the curve, the modulations are made through Sine, Cosine and Triangle (See: Figure 4.10). When modulation occurs, the phase values must be chosen between 25 π and 100π. **<u>25π is chosen</u>.** The ripple amplitude is chosen to be **<u>1dB</u>.**

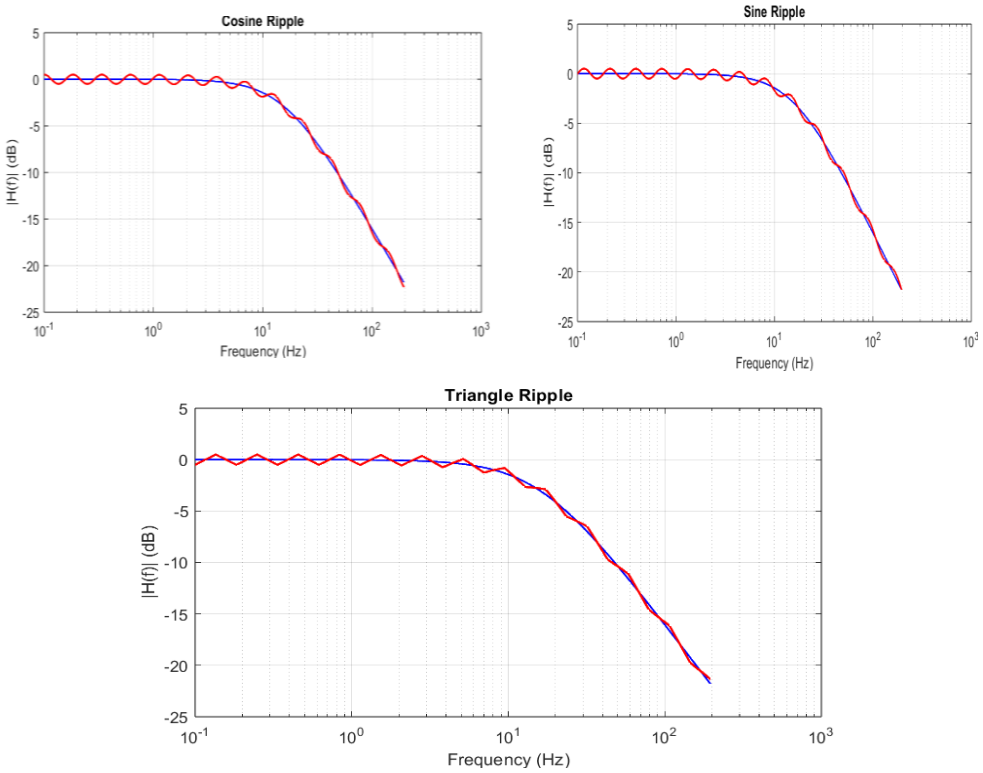


Figure 4.10 RC LPF Modulations Catalogue At 25π

When the customer finally looks at the modulation catalogue, the customer chooses Cosine modulation. The Neural networks will train the modulated signal (See: Figure 4.11).

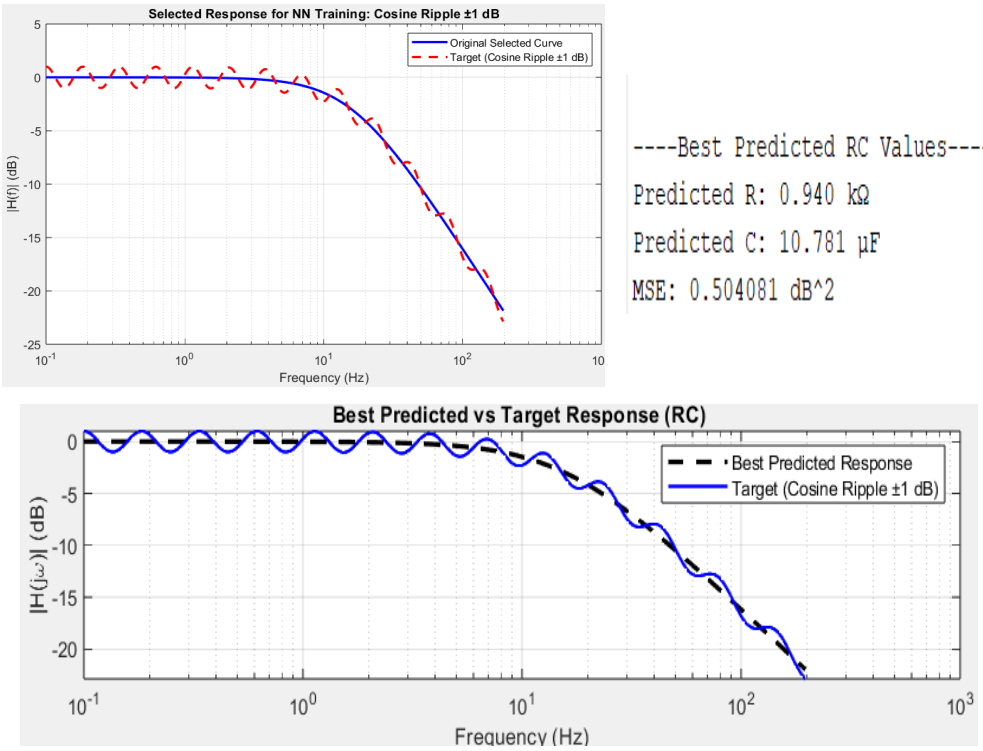


Figure 4.11 RC LPF Prediction Using Cosine Modulation

The modulated curve's Error Matrix of R and C can be shown also be shown below (See: Figure 4.12).

R_True	R_Pred	R_Error	R_Error_percent	C_True	C_Pred	C_Error	C_Error_percent
900	858.78	-41.218	-4.5798	9e-06	1.0896e-05	1.8958e-06	21.064
900	939.5	39.503	4.3892	1e-05	1.0781e-05	7.8098e-07	7.8098
900	1051.2	151.25	16.805	1.1e-05	9.8932e-06	-1.1068e-06	-10.062
1000	939.5	-60.497	-6.0497	9e-06	1.0781e-05	1.781e-06	19.789
1000	1040.9	40.907	4.0907	1e-05	9.881e-06	-1.1903e-07	-1.1903
1000	1010.7	10.723	1.0723	1.1e-05	9.3038e-06	-1.6962e-06	-15.42
1100	1051.2	-48.753	-4.4321	9e-06	9.8932e-06	8.9316e-07	9.924
1100	1010.7	-89.277	-8.1161	1e-05	9.3038e-06	-6.9616e-07	-6.9616
1100	1021.1	-78.858	-7.169	1.1e-05	9.4966e-06	-1.5034e-06	-13.667

Figure 4.12 Error Matrix of Cosine Modulation

4.2.2 Sallen-Key Bandpass Filter

Sallen-Key circuit parameters are input through the user interface, allowing the user to freely specify any desired values (See: Figure 4.13).

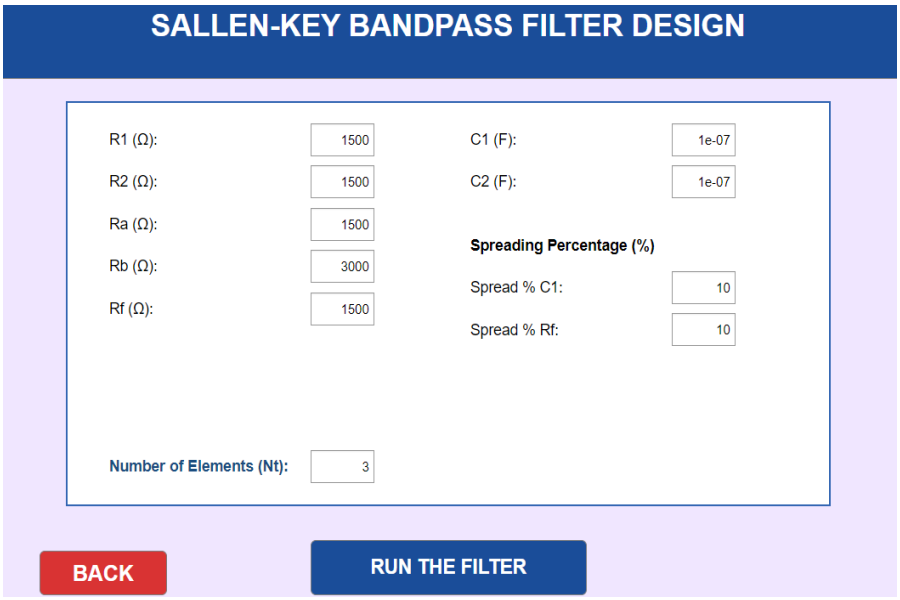


Figure 4.13 Sallen-Key BPF GUI

When the user enters the parameter values, the frequency response provides **9 lines**. The table below shows the sweeping parameters with their cut-off frequencies as **EM** (See: Table 4.2).

Table 4.2 Sallen-Key BPF Element Matrix

C_1 (nF)	R_f (Ω)	$f_0\ (rad/s)$
90 nF	1350	1625
90 nF	1500	1581.7
90 nF	1650	1545.3
100 nF	1350	1541.6
100 nF	1500	1500.5
100 nF	1650	1466
110 nF	1350	1469.9
110 nF	1500	1430.7
110 nF	1650	1397.8

When the EM is shown, the magnitude responses of the filter are shown (See: Figure 4.14). However, the customer needs to see the catalogue for this filter before choosing his suitable curve (See: Figure 4.15).

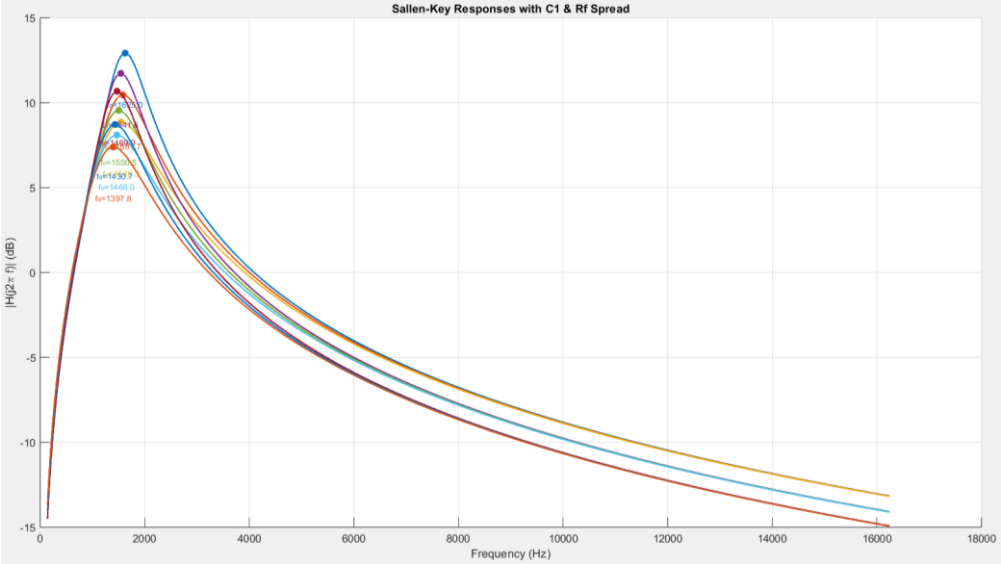


Figure 4.14 Sallen-Key BPF Magnitude Responses

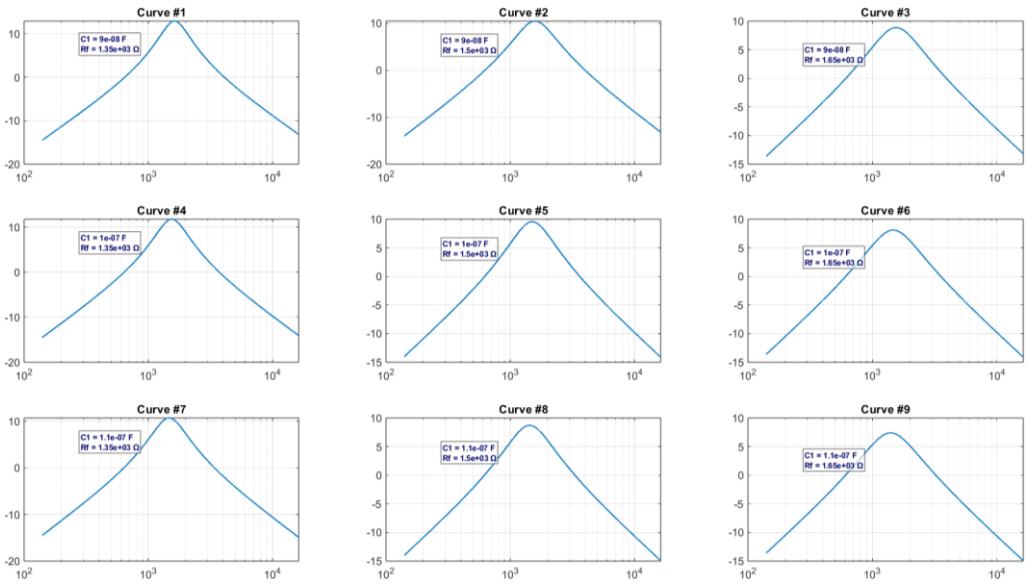


Figure 4.15 Sallen-Key BPF Catalogue

At this stage, the user can review and examine all available curves, and after previewing them, select the most appropriate one. Once the final curve is chosen, its corresponding characteristics are displayed. For example, assume the user selects the fifth curve (See: Figure 4.16).

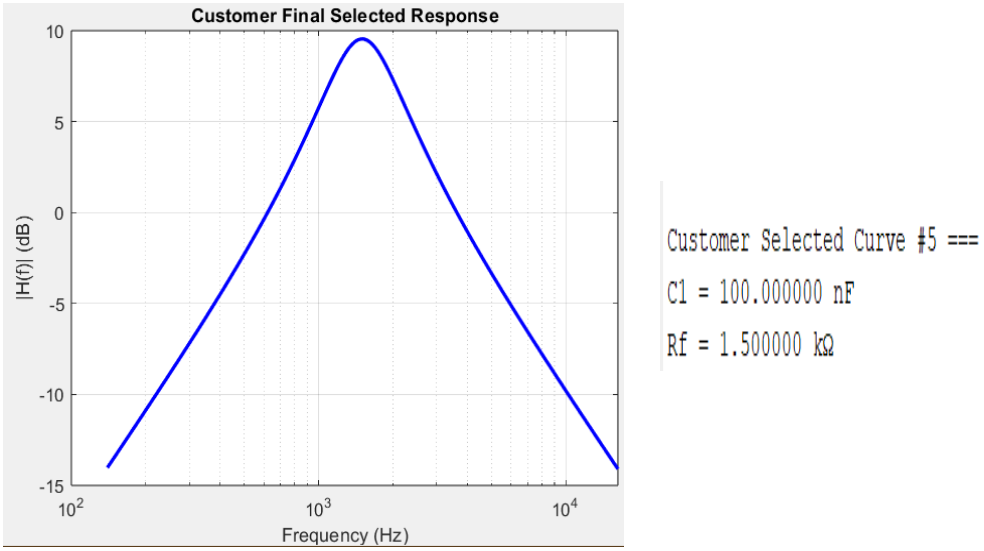


Figure 4.16 Customer's Final Selection

Once the user selects a curve, the modulation option becomes relevant. If no modulation is required, the selected curve is directly processed by the neural network to determine its optimal parameter values. However, if modulation is desired, the modulated version of the curve is trained through the neural network to obtain the optimized parameters for the modulated curve.

Case 1: If the user selects "No Modulation", the predicted values, the corresponding predicted curve, and the error matrix are displayed (See: Figure 4.17).

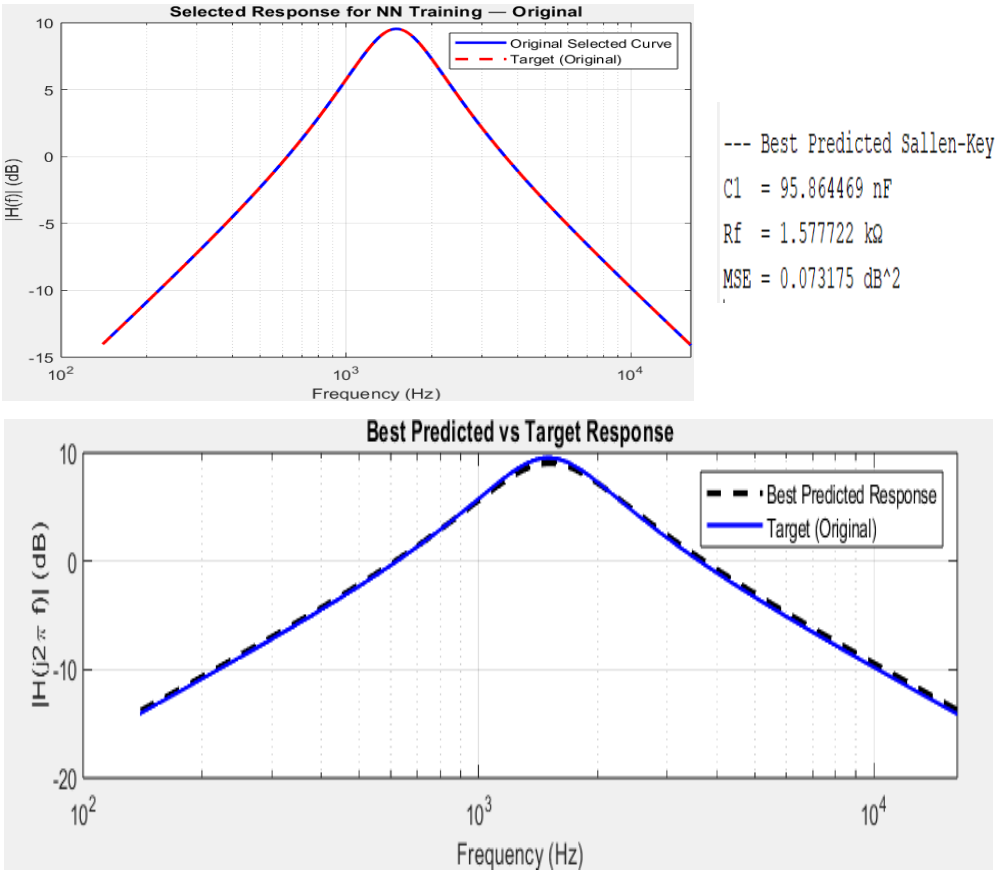


Figure 4.17 Sallen-Key BPF No Modulation Result

The Error Matrix of R_f and C_1 can be shown also be shown below (See: Figure 4.18).

C1_True	C1_Pred	C1_Error	C1_Error_percent	Rf_True	Rf_Pred	Rf_Error	Rf_Error_percent
9e-08	9.5903e-08	5.9026e-09	6.5584	1350	1458	107.97	7.9981
9e-08	8.9733e-08	-2.6707e-10	-0.29675	1500	1568.4	68.412	4.5608
9e-08	8.6342e-08	-3.6578e-09	-4.0643	1650	1648.9	-1.0573	-0.064076
1e-07	9.5952e-08	-4.0479e-09	-4.0479	1350	1415.6	65.619	4.8606
1e-07	8.4976e-08	-1.5024e-08	-15.024	1500	1498.4	-1.5778	-0.10519
1e-07	9.0209e-08	-9.7905e-09	-9.7905	1650	1677.6	27.58	1.6715
1.1e-07	9.0254e-08	-1.9746e-08	-17.951	1350	1361.5	11.481	0.85044
1.1e-07	9.5236e-08	-1.4764e-08	-13.421	1500	1549.4	49.408	3.2939
1.1e-07	9.5864e-08	-1.4136e-08	-12.85	1650	1577.7	-72.278	-4.3805

Figure 4.18 No Modulation Error Matrix

Case 2: If the user chooses to apply modulation, the curve can be modulated using Sine, Cosine, or Triangle functions (See: Figure 4.19). During modulation, the phase value must be selected within the range of 25π to 100π. A phase of **25π** is used. The ripple amplitude is set to **1 dB**.

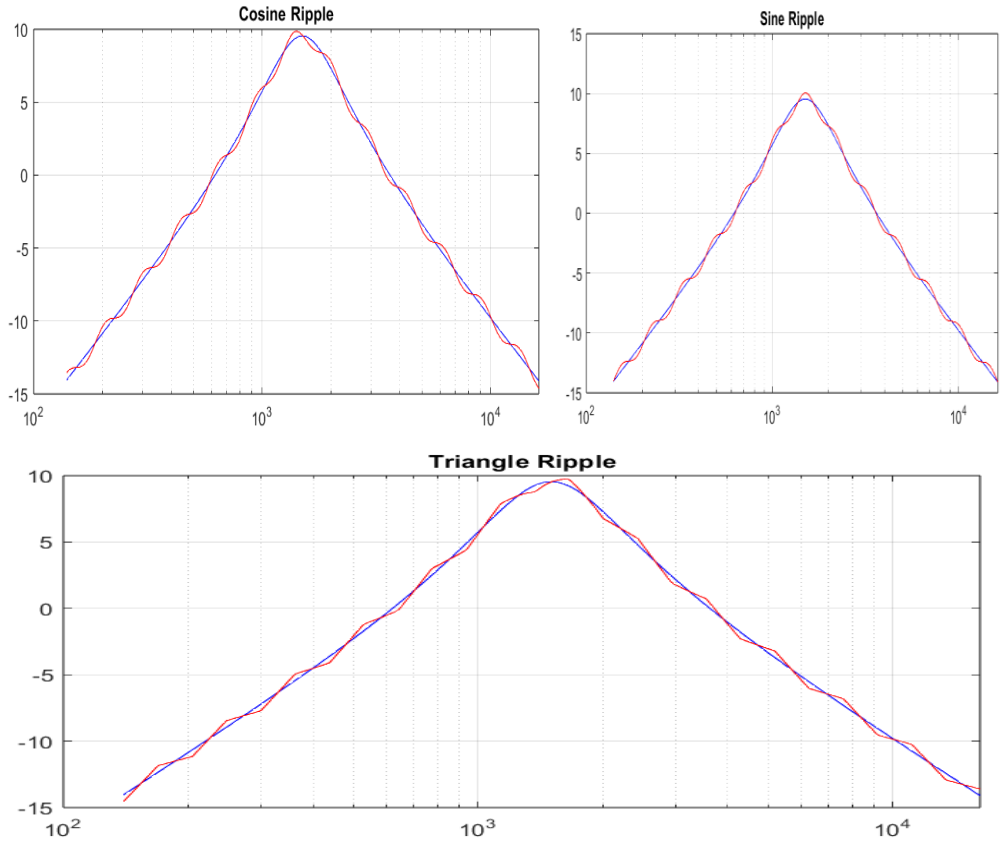
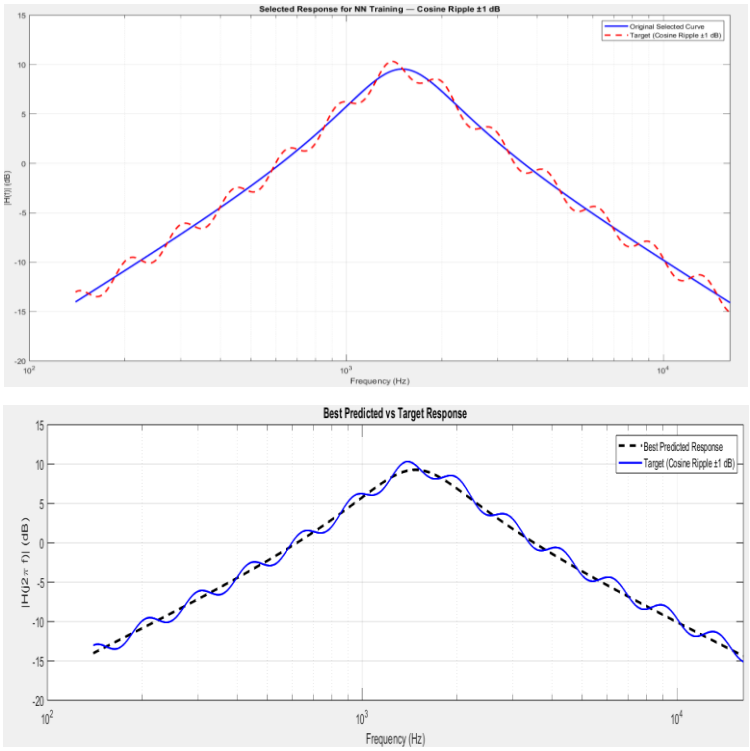


Figure 4.19 Sallen-Key BPF Modulations Catalogue At 25$\boldsymbol{\pi}$

After reviewing the modulation catalogue, the user selects the Cosine modulation option. The neural network then trains the corresponding modulated signal (See: Figure 4.20).




```
--- Best Predicted Sallen-Key Bandpass Filter Values ---
C1  = 103.184389 nF
Rf  = 1.500000 kΩ
MSE = 0.553069 dB^2
```

Figure 4.20 Sallen-Key Prediction Using Cosine Modulation

The modulated curve's Error Matrix of R_f and C_1 can be shown also be shown below (See: Figure 4.21).

C1_True	C1_Pred	C1_Error	C1_Error_percent	Rf_True	Rf_Pred	Rf_Error	Rf_Error_percent
9e-08	9.8205e-08	8.2049e-09	9.1166	1350	1412.1	62.13	4.6023
9e-08	1.0318e-07	1.3184e-08	14.649	1500	1500	-2.2737e-13	-1.5158e-14
9e-08	1.0581e-07	1.5814e-08	17.571	1650	1650	0	0
1e-07	1.0812e-07	8.1208e-09	8.1208	1350	1350	0	0
1e-07	1.1638e-07	1.6375e-08	16.375	1500	1500	0	0
1e-07	1.188e-07	1.8802e-08	18.802	1650	1650	0	0
1.1e-07	1.1128e-07	1.2765e-09	1.1605	1350	1363.6	13.607	1.0079
1.1e-07	1.2406e-07	1.4058e-08	12.78	1500	1500	-2.2737e-13	-1.5158e-14
1.1e-07	1.2416e-07	1.4157e-08	12.87	1650	1650	0	0

Figure 4.21 Error Matrix of Cosine Modulation

4.2.3 Single Stage LC Bandpass Filter

Single-Stage LC circuit parameters are entered via the user interface, enabling the user to assign any preferred values (See: Figure 4.22).

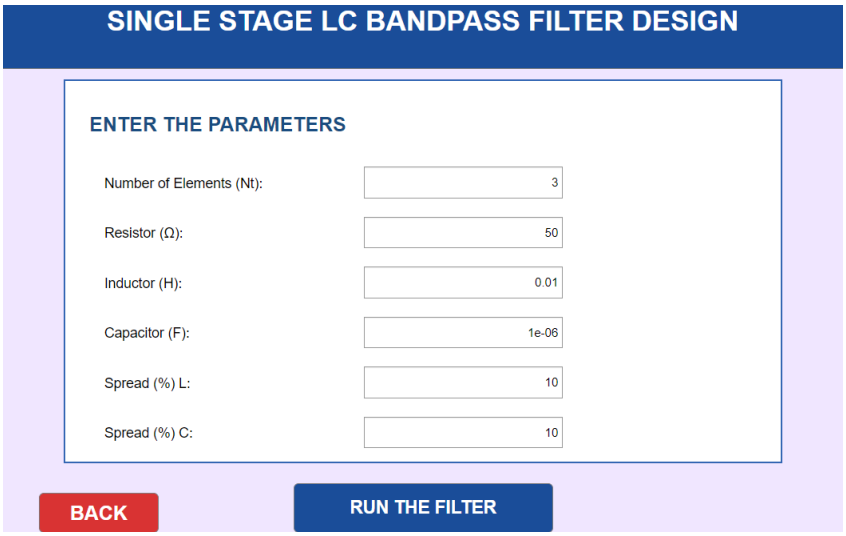


Figure 4.22 Single-Stage LC BPF GUI

After the user inputs the parameter values, the resulting frequency response generates nine curves. The table below presents the swept parameters along with their corresponding cut-off frequencies as EM (See: Table 4.3).

Table 4.3 Single-Stage LC BPF Element Matrix

L (mH)	C (μF)	$f_0\ (rad/s)$
9mH	0.9 μF	1768.4
9mH	1 μF	1677.6
9mH	1.1 μF	1599.6
10mH	0.9 μF	1677.6
10mH	1 μF	1591.5
10mH	1.1 μF	1517.6
11mH	0.9 μF	1599.6
11mH	1 μF	1517.6
11mH	1.1 μF	1446.9

Once the EM is displayed, the filter's magnitude responses are presented (See: Figure 4.23). However, before selecting the appropriate curve, the user must first review the filter's catalogue (See: Figure 4.24).

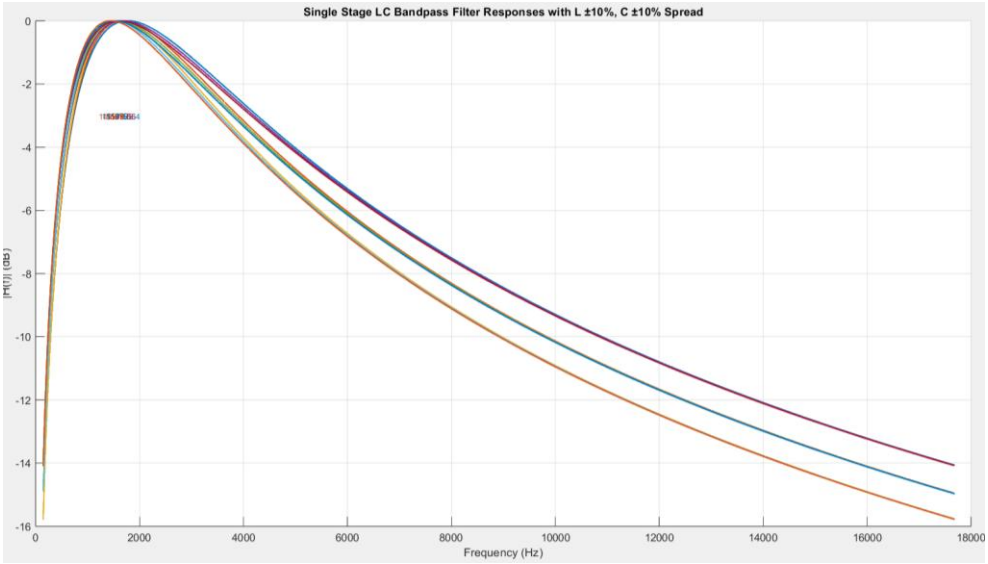


Figure 4.23 Single-Stage LC BPF Magnitude Responses

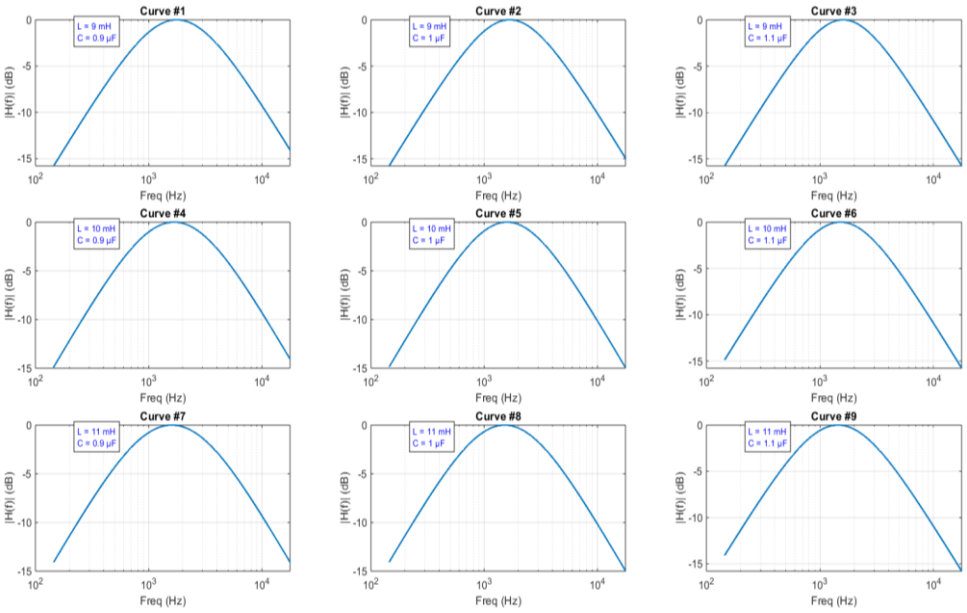


Figure 4.24 Single-Stage LC BPF Catalogue

At this phase, the user is able to inspect all available curves and, after reviewing them, select the most suitable option. Once the final curve is chosen, its corresponding characteristics are presented. For instance, assume the user selects the fifth curve (See: Figure 4.25).

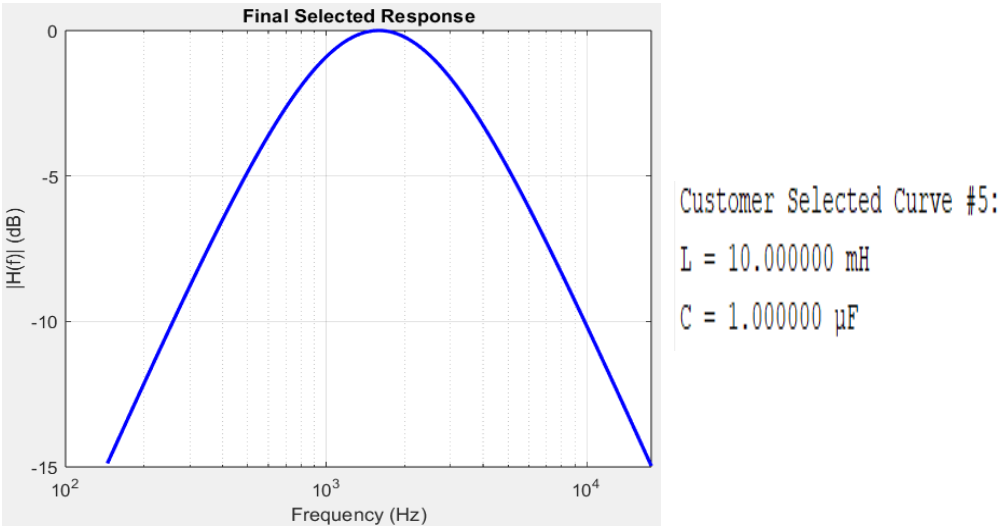


Figure 4.25 Customer's Final Selection

Once a curve is selected, the modulation option becomes active. If the user does not require modulation, the chosen curve is processed directly by the neural network to estimate its optimal parameter values. In contrast, if modulation is requested, the neural network trains the modulated curve to determine its optimized parameters.

Case 1: When the user chooses "No Modulation," the system presents the predicted values, the corresponding predicted curve, and the error matrix (See: Figure 4.26).

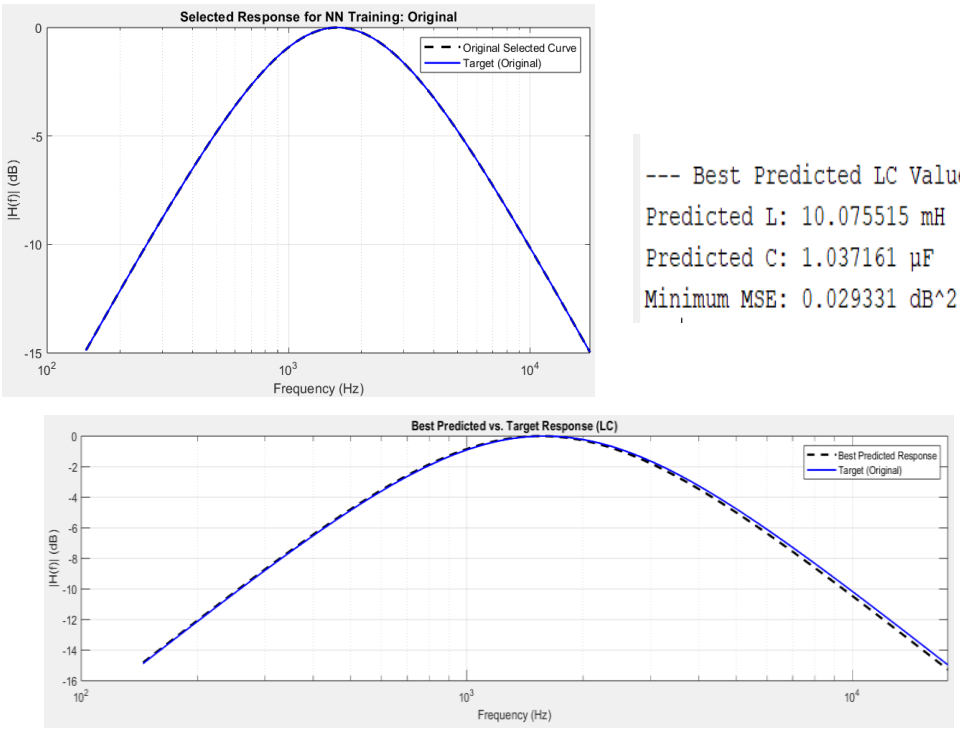


Figure 4.26 Single-Stage LC BPF No Modulation Prediction

The error matrix for L and C is also presented below (See: Figure 4.27).

L_True	L_Pred	L_Error	L_Error_percent	C_True	C_Pred	C_Error	C_Error_percent
0.009	0.010209	0.0012091	13.434	9e-07	9.2506e-07	2.5057e-08	2.7841
0.009	0.010076	0.0010755	11.95	1e-06	1.0372e-06	3.7161e-08	3.7161
0.009	0.010036	0.0010357	11.508	1.1e-06	1.1517e-06	5.1738e-08	4.7034
0.01	0.010749	0.00074935	7.4935	9e-07	9.5163e-07	5.1633e-08	5.737
0.01	0.010353	0.00035261	3.5261	1e-06	1.0683e-06	6.8337e-08	6.8337
0.01	0.010417	0.0004173	4.173	1.1e-06	1.1428e-06	4.2757e-08	3.887
0.011	0.010793	-0.00020736	-1.8851	9e-07	9.8193e-07	8.1926e-08	9.1029
0.011	0.010614	-0.00038594	-3.5086	1e-06	1.0666e-06	6.6594e-08	6.6594
0.011	0.010938	-6.2073e-05	-0.5643	1.1e-06	1.0926e-06	-7.3528e-09	-0.66844

Figure 4.27 No Modulation Error Matrix

Case 2: If the user opts for modulation, the curve may be modified using Sine, Cosine, or Triangle functions (See: Figure 4.28). During this process, the phase must be chosen within the interval of 25π to 100π; in this example, a phase of **25π** is selected. The ripple amplitude is specified as **1 dB**.

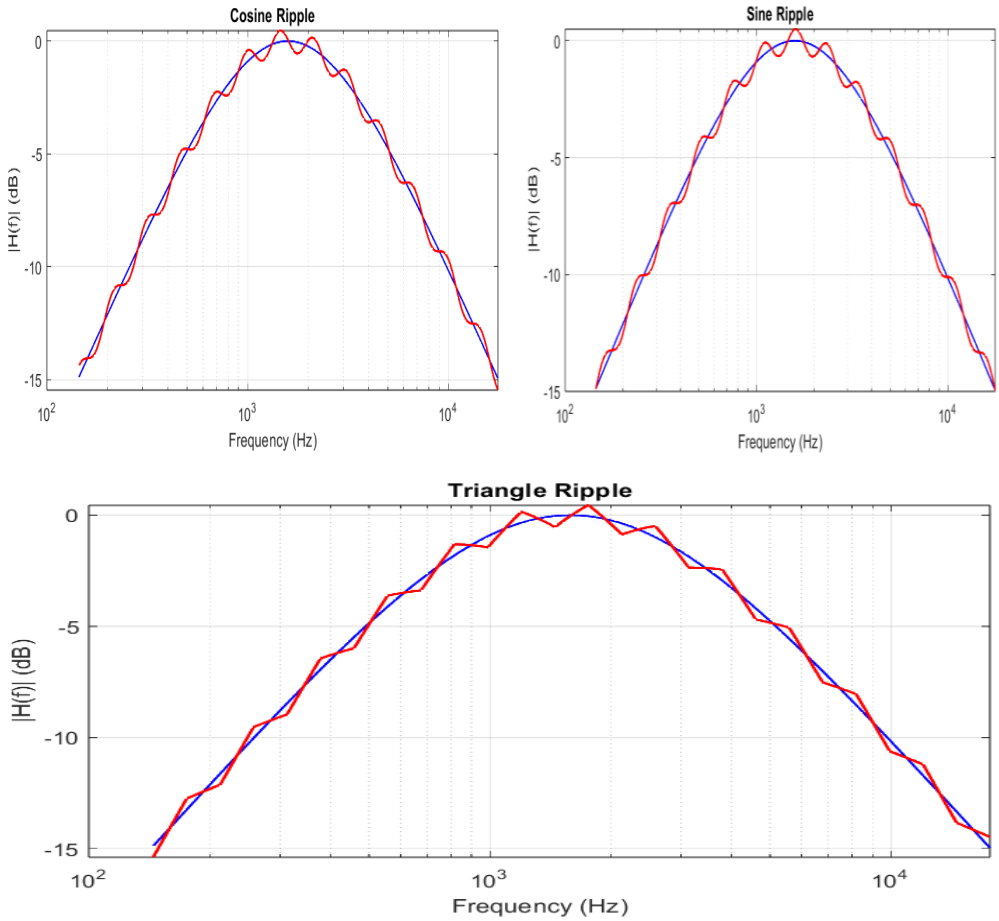
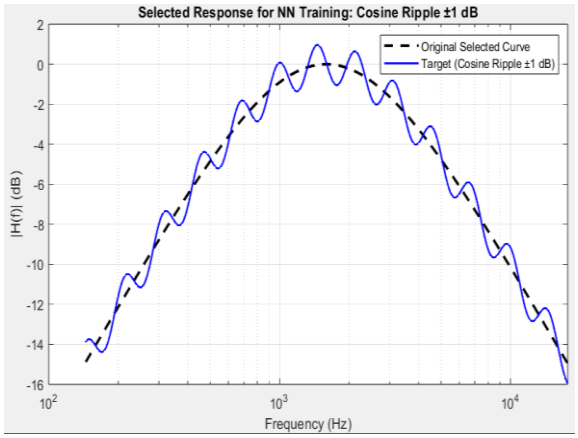


Figure 4.28 Single-Stage LC BPF Modulations Catalogue At 25$\boldsymbol{\pi}$

After examining the modulation catalogue, the user chooses the Cosine modulation option. The neural network subsequently trains the associated modulated signal (See: Figure 4.29).




```
--- Best Predicted LC Values
Predicted L: 10.079843 mH
Predicted C: 1.034014 µF
Minimum MSE: 0.525236 dB^2
```

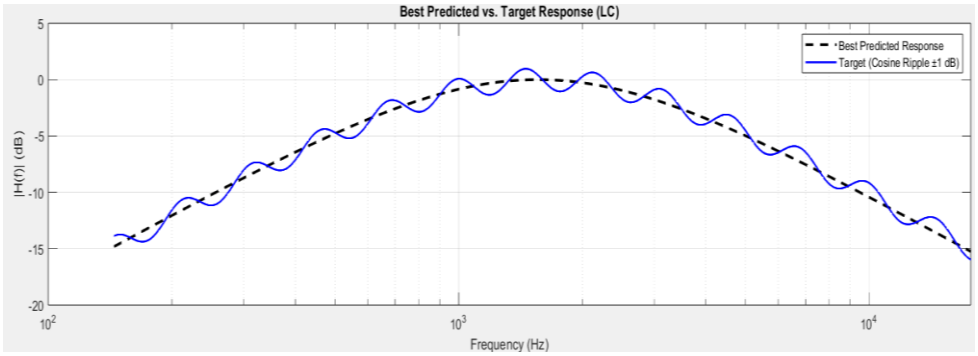


Figure 4.29 Single-Stage LC BPF Prediction Using Cosine Modulation

The error matrix for the modulated curve's L and C values is also presented below (See: Figure 4.30).

L_True	L_Pred	L_Error	L_Error_percent	C_True	C_Pred	C_Error	C_Error_percent
0.009	0.0090571	5.7075e-05	0.63416	9e-07	1.2772e-06	3.7721e-07	41.913
0.009	0.0090575	5.7522e-05	0.63913	1e-06	1.3691e-06	3.6906e-07	36.906
0.009	0.0089776	-2.2357e-05	-0.24841	1.1e-06	1.2933e-06	1.933e-07	17.573
0.01	0.0099655	-3.4509e-05	-0.34509	9e-07	9.1863e-07	1.8633e-08	2.0703
0.01	0.01008	7.9843e-05	0.79843	1e-06	1.034e-06	3.4014e-08	3.4014
0.01	0.0097447	-0.00025535	-2.5535	1.1e-06	1.0687e-06	-3.1335e-08	-2.8486
0.011	0.01092	-7.9821e-05	-0.72565	9e-07	7.7692e-07	-1.2308e-07	-13.676
0.011	0.010897	-0.0001034	-0.94	1e-06	8.7238e-07	-1.2762e-07	-12.762
0.011	0.011013	1.3231e-05	0.12028	1.1e-06	9.7422e-07	-1.2578e-07	-11.435

Figure 4.30 Error Matrix of Cosine Modulation

4.2.4 Two Stage LC Bandpass Filter

The parameters of the Two-Stage LC circuit are entered through the user interface, allowing the user to specify any desired values (See: Figure 4.31).

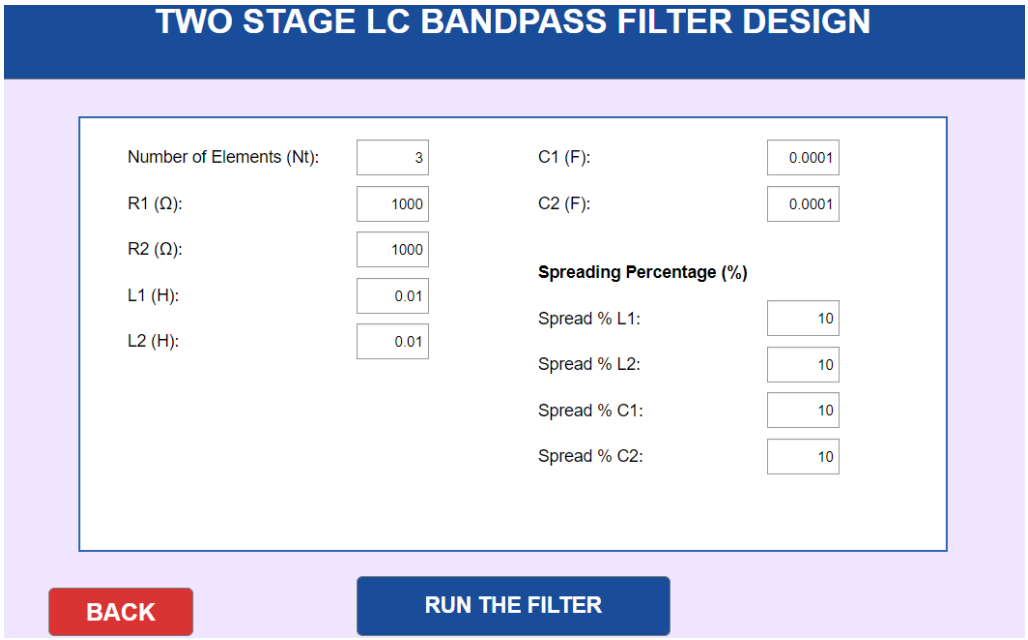


Figure 4.31 Two-Stage LC BPF GUI

Once the user enters the parameter values, the generated frequency response produces nine curves. Table 4.4 below lists the swept parameters together with their associated cut-off frequencies as EM.

$$\mathbf{N_T = 3^4 = 81\ Possibilities.}$$

Table 4.4 Two-Stage LC BPF Element Matrix

L_1 (mH)	L_2 ((mH))	C_1 (µF)	C_2 (µF)
9mH	9mH	90 µF	90 µF
9mH	9mH	90 µF	100 µF
9mH	9mH	90 µF	110 µF
9mH	9mH	100 µF	90 µF
9mH	9mH	100 µF	100 µF

9mH	9mH	100 µF	110 µF
9mH	9mH	110 µF	90 µF
9mH	9mH	110 µF	100 µF
9mH	9mH	110 µF	110 µF
9mH	10mH	90 µF	90 µF
9mH	10mH	90 µF	100 µF
9mH	10mH	90 µF	110 µF
9mH	10mH	100 µF	90 µF
9mH	10mH	100 µF	100 µF
9mH	10mH	100 µF	110 µF
9mH	10mH	110 µF	90 µF
9mH	10mH	110 µF	100 µF
9mH	10mH	110 µF	110 µF
9mH	11mH	90 µF	90 µF
9mH	11mH	90 µF	100 µF
9mH	11mH	90 µF	110 µF
9mH	11mH	100 µF	90 µF
9mH	11mH	100 µF	100 µF
9mH	11mH	100 µF	110 µF
9mH	11mH	110 µF	90 µF
9mH	11mH	110 µF	100 µF
9mH	11mH	110 µF	110 µF
10mH	9mH	90 µF	90 µF
10mH	9mH	90 µF	100 µF
10mH	9mH	90 µF	110 µF
10mH	9mH	100 µF	90 µF
10mH	9mH	100 µF	100 µF
10mH	9mH	100 µF	110 µF
10mH	9mH	110 µF	90 µF
10mH	9mH	110 µF	100 µF
10mH	9mH	110 µF	110 µF
10mH	10mH	90 µF	90 µF
10mH	10mH	90 µF	100 µF
10mH	10mH	90 µF	110 µF
10mH	10mH	100 µF	90 µF
10mH	10mH	100 µF	100 µF
10mH	10mH	100 µF	110 µF
10mH	10mH	110 µF	90 µF
10mH	10mH	110 µF	100 µF
10mH	10mH	110 µF	110 µF
10mH	11mH	90 µF	90 µF
10mH	11mH	90 µF	100 µF
10mH	11mH	90 µF	110 µF
10mH	11mH	100 µF	90 µF

10mH	11mH	100 µF	100 µF
10mH	11mH	100 µF	110 µF
10mH	11mH	110 µF	90 µF
10mH	11mH	110 µF	100 µF
10mH	11mH	110 µF	110 µF
11mH	9mH	90 µF	90 µF
11mH	9mH	90 µF	100 µF
11mH	9mH	90 µF	110 µF
11mH	9mH	100 µF	90 µF
11mH	9mH	100 µF	100 µF
11mH	9mH	100 µF	110 µF
11mH	9mH	110 µF	90 µF
11mH	9mH	110 µF	100 µF
11mH	9mH	110 µF	110 µF
11mH	10mH	90 µF	90 µF
11mH	10mH	90 µF	100 µF
11mH	10mH	90 µF	110 µF
11mH	10mH	100 µF	90 µF
11mH	10mH	100 µF	100 µF
11mH	10mH	100 µF	110 µF
11mH	10mH	110 µF	90 µF
11mH	10mH	110 µF	100 µF
11mH	10mH	110 µF	110 µF
11mH	11mH	90 µF	90 µF
11mH	11mH	90 µF	100 µF
11mH	11mH	90 µF	110 µF
11mH	11mH	100 µF	90 µF
11mH	11mH	100 µF	100 µF
11mH	11mH	100 µF	110 µF
11mH	11mH	110 µF	90 µF
11mH	11mH	110 µF	100 µF
11mH	11mH	110 µF	110 µF

After the EM is shown, the filter's magnitude responses are displayed (See: Figure 4.32). However, before choosing the appropriate curve, the user is required to review the filter's catalogue (See: Figure 4.33).

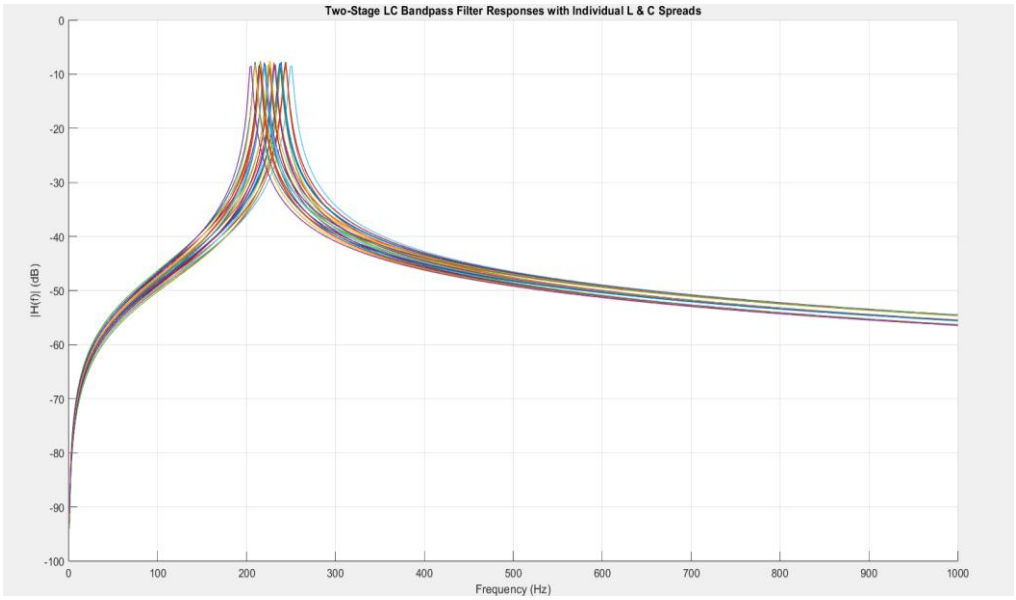


Figure 4.32 Two-Stage LC BPF Magnitude Responses

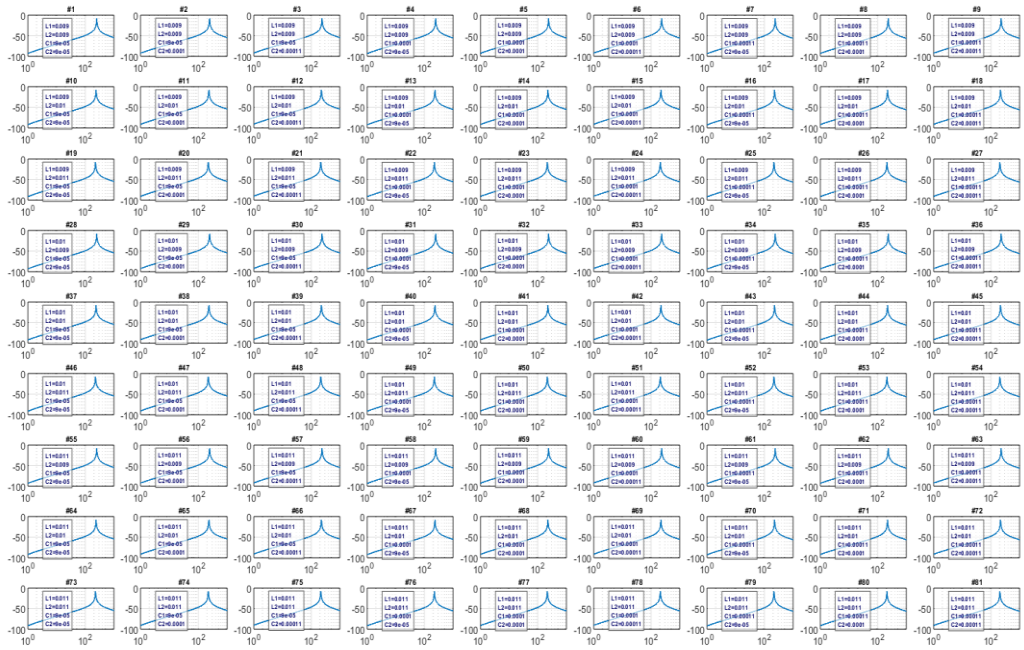


Figure 4.33 Two-Stage LC BPF Catalogue

During this phase, the user can examine all available curves and, after evaluation, choose the most appropriate one. Once the final curve is selected, its

associated characteristics are displayed. For example, consider the case where the user selects the forty-first (**41st**) curve (See: Figure 4.34).

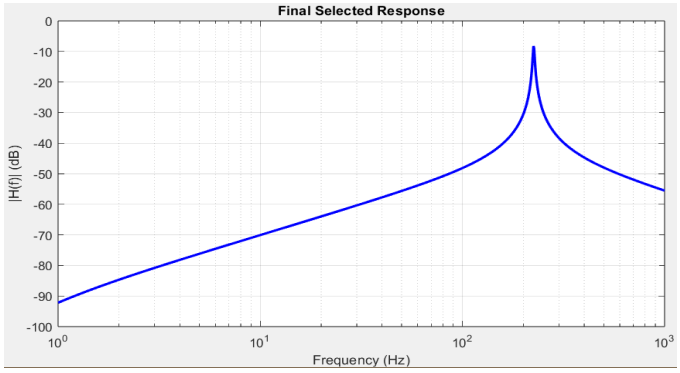
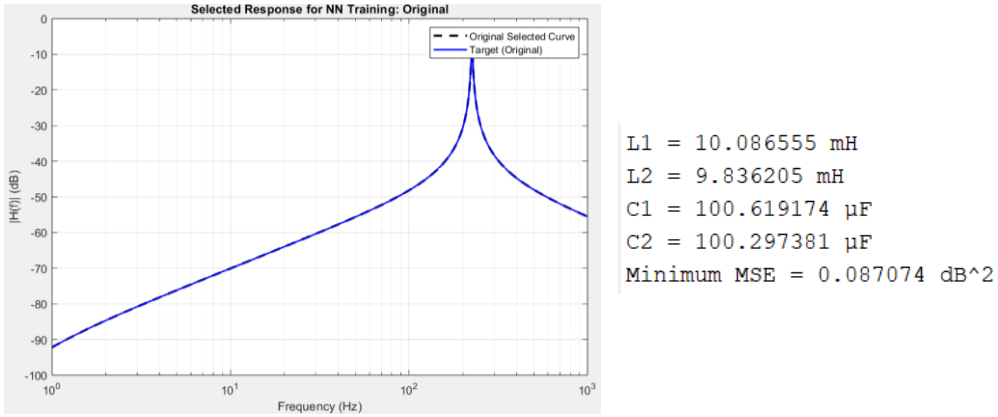


Figure 4.34 Customer's Final Selection

After the user selects a curve, the modulation feature becomes available for use. If modulation is not selected, the neural network processes the original curve to derive its optimal parameter values. However, if the user opts for modulation, the neural network instead trains the modified curve to generate its optimized parameters.

Case 1: When "No Modulation" is chosen, the system outputs the predicted parameter values, the resulting predicted curve, and the corresponding error matrix (See: Figure 4.35).



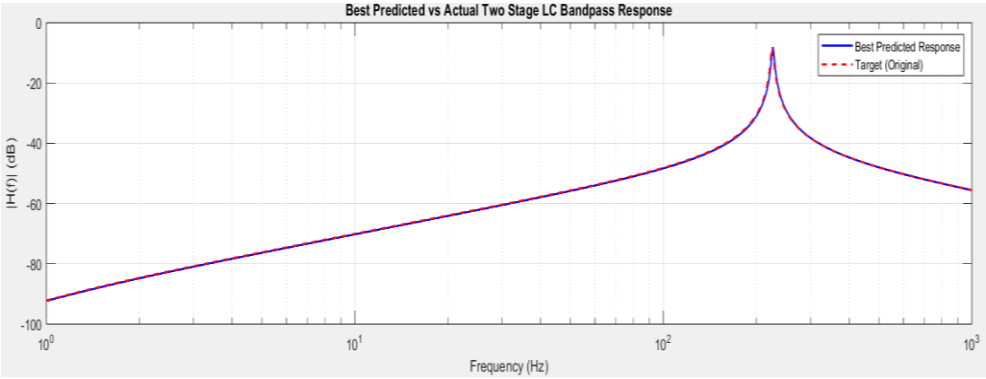


Figure 4.35 Two-Stage LC BPF No Modulation Prediction

The first 10 rows of error matrix for L_1, L_2, C_1 and L_2 are presented below (See: Figure 4.36).

L1_Pred	L2_Pred	C1_Pred	C2_Pred	L1_Error	L2_Error	C1_Error	C2_Error
0.0098407	0.0094416	0.00010065	8.7513e-05	0.00084071	0.00044163	1.0648e-05	-2.4866e-06
0.0099138	0.0094026	0.0001003	9.1933e-05	0.00091379	0.00040258	1.0305e-05	-8.0672e-06
0.010042	0.0094243	9.8934e-05	9.785e-05	0.0010417	0.00042429	8.934e-06	-1.215e-05
0.010124	0.0092921	0.00010409	9.0226e-05	0.0011241	0.00029205	4.0944e-06	2.2594e-07
0.010127	0.0094557	0.0001028	9.5335e-05	0.0011269	0.0004557	2.7968e-06	-4.6654e-06
0.010249	0.0094904	0.0001029	0.00010053	0.0012491	0.0004904	2.9031e-06	-9.4658e-06
0.010123	0.009173	0.00010781	9.0342e-05	0.0011231	0.000173	-2.1916e-06	3.4239e-07
0.009929	0.0091729	0.00011031	9.395e-05	0.00092902	0.00017288	3.0733e-07	-6.0501e-06
0.009686	0.0092884	0.00011113	9.8435e-05	0.00068599	0.00028841	1.1315e-06	-1.1565e-05
0.0098178	0.0097684	9.6042e-05	9.6996e-05	0.00081777	-0.00023155	6.0418e-06	6.9963e-06

Figure 4.36 No Modulation Error Matrix

Case 2: If the user selects modulation, the curve can be altered using Sine, Cosine, or Triangle modulation functions (See: Figure 4.37). In applying modulation, the phase value must fall within the range of 25π to 100π. A phase of **25π** is chosen. The ripple amplitude is set to **1dB.**

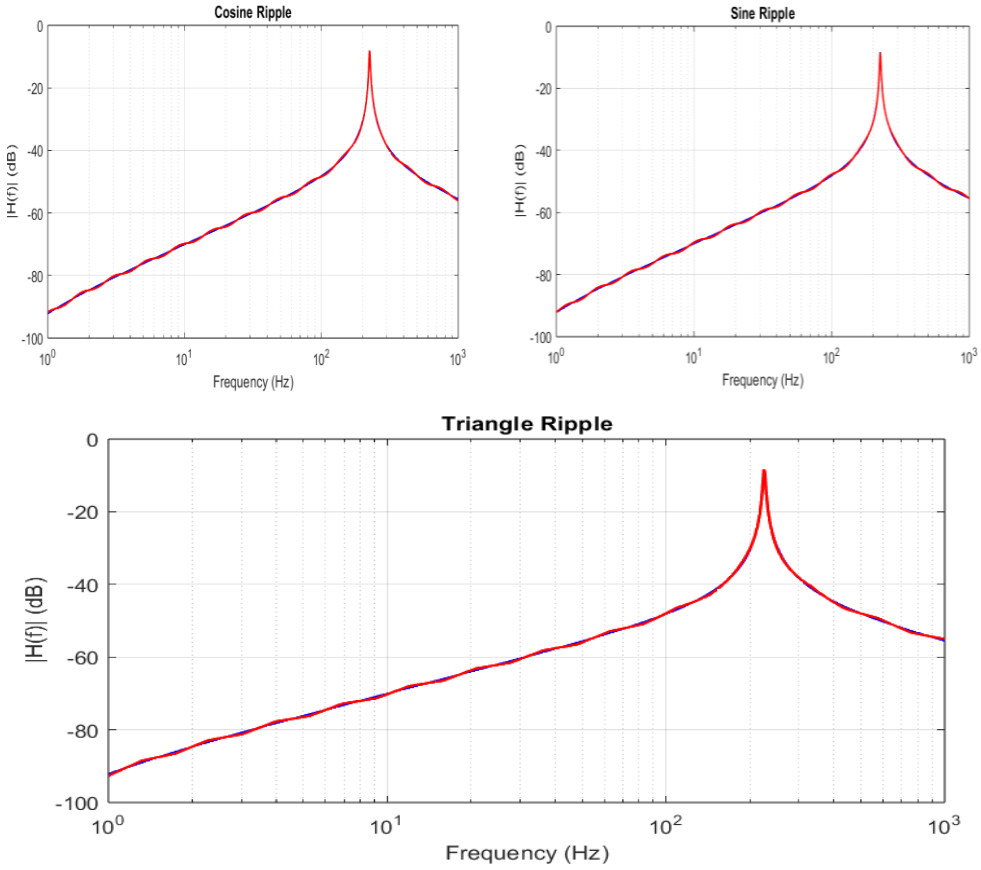
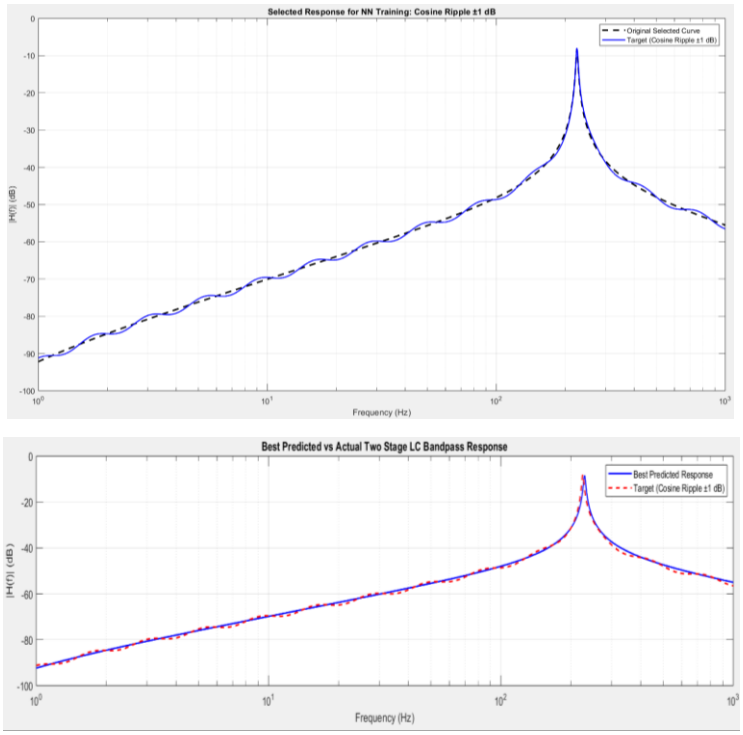


Figure 4.37 Two-Stage LC BPF Modulations Catalogue At 25$\boldsymbol{\pi}$

After reviewing the modulation catalogue, the user selects the Cosine modulation approach. The neural network then trains the resulting modulated signal (See: Figure 4.38).




```
--- Best Predicted (L1,L2,C1 and C2) Values
L1 = 9.995372 mH
L2 = 10.802977 mH
C1 = 94.012797 µF
C2 = 85.264193 µF
Minimum MSE = 0.973889 dB^2
```

Figure 4.38 Two-Stage LC BPF Prediction Using Cosine Modulation

The error matrix for the modulated curve's values is also presented below (See: Figure 4.39).

L1_Pred	L2_Pred	C1_Pred	C2_Pred	L1_Error	L2_Error	C1_Error	C2_Error
0.010057	0.0090954	8.4344e-05	0.00011275	0.001057	9.5405e-05	-5.6562e-06	2.2749e-05
0.010021	0.0091562	8.4038e-05	0.00011048	0.0010213	0.00015625	-5.9616e-06	1.0481e-05
0.010041	0.0093472	8.4745e-05	0.00010758	0.0010406	0.00034723	-5.255e-06	-2.424e-06
0.010045	0.0090669	8.2295e-05	0.00011508	0.0010445	6.685e-05	-1.7705e-05	2.5077e-05
0.010011	0.0092278	8.2237e-05	0.00011618	0.0010114	0.00022777	-1.7763e-05	1.6177e-05
0.0099033	0.0093573	8.2148e-05	0.00011665	0.00090333	0.00035731	-1.7852e-05	6.6514e-06
0.010022	0.009034	8.0105e-05	0.00011855	0.0010221	3.3957e-05	-2.9895e-05	2.8551e-05
0.0098595	0.0091705	7.9431e-05	0.00012054	0.00085947	0.00017054	-3.0569e-05	2.0536e-05
0.0097444	0.0094127	8.1763e-05	0.00012298	0.00074437	0.00041271	-2.8237e-05	1.2978e-05
0.010116	0.0096925	8.6367e-05	0.00010859	0.0011163	-0.00030748	-3.6328e-06	1.8587e-05

Figure 4.39 Error Matrix of Cosine Modulation

4.2.5 Transconductance-Capacitor Bandpass Filter

Gm–C circuit parameters are input via the user interface, enabling the user to assign any desired values (See: Figure 4.40).

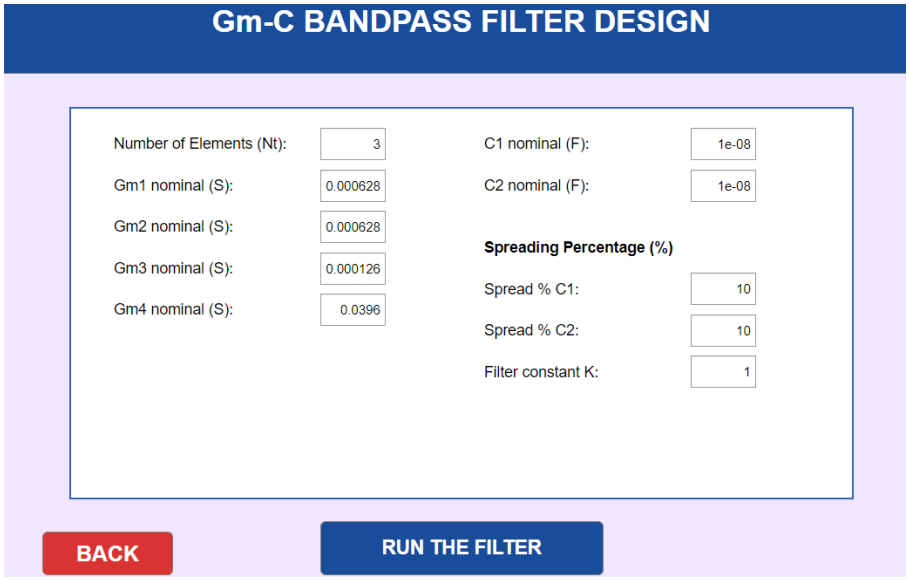


Figure 4.40 Gm-C BPF GUI

After the user inputs the parameter values, the resulting frequency response generates nine curves. Table 4.5 presents the swept parameters along with their corresponding cut-off frequencies as EM.

Table 4.5 Gm-C BPF Element matrix

C_1 (nF)	C_2 (nF)	$f_0\ (rad/s)$	Bandwidth (β)
9nF	9nF	11105	2228.2
9nF	10nF	10536	2005.4
9nF	11nF	10045	1823
10nF	9nF	10536	2228.2
10nF	10nF	9994.9	2005.4
10nF	11nF	9529.8	1823
11nF	9nF	10045	2228.2
11nF	10nF	9529.8	2005.4
11nF	11nF	9086.3	1823

Once the EM is displayed, the filter's magnitude responses are shown (See: Figure 4.41). Before selecting the most suitable curve, the user must review the filter's catalogue (See: Figure 4.42). This ensures that the chosen curve aligns with the desired filter performance and specifications.

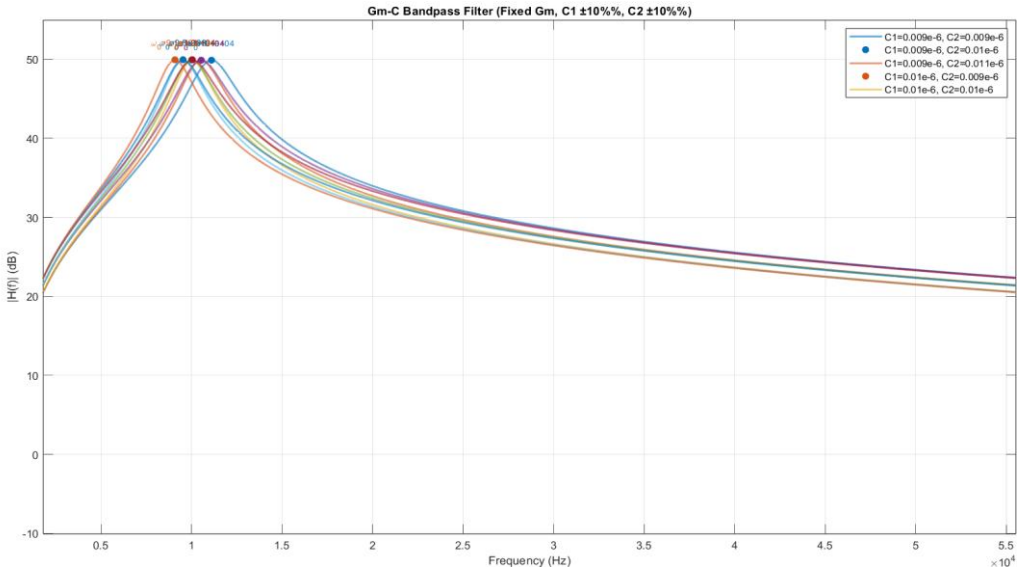


Figure 4.41 Gm-C BPF Magnitude Responses

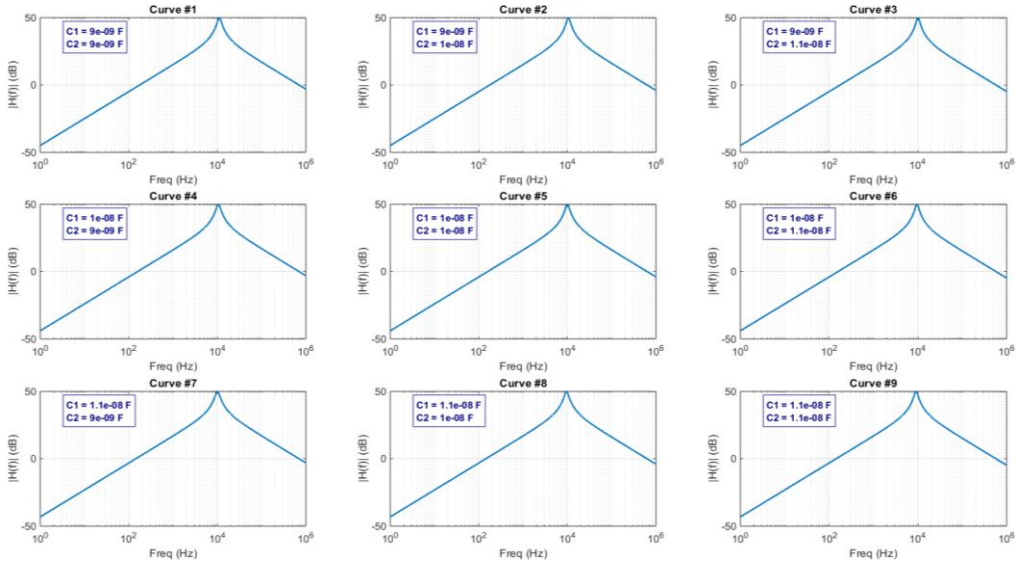


Figure 4.42 Gm-C BPF Catalogue

In this phase, the user is able to review all available curves and, after careful evaluation, select the most suitable one. Once the final curve is chosen, its corresponding characteristics are presented. For instance, assume the user selects the fifth curve (See: Figure 4.43).

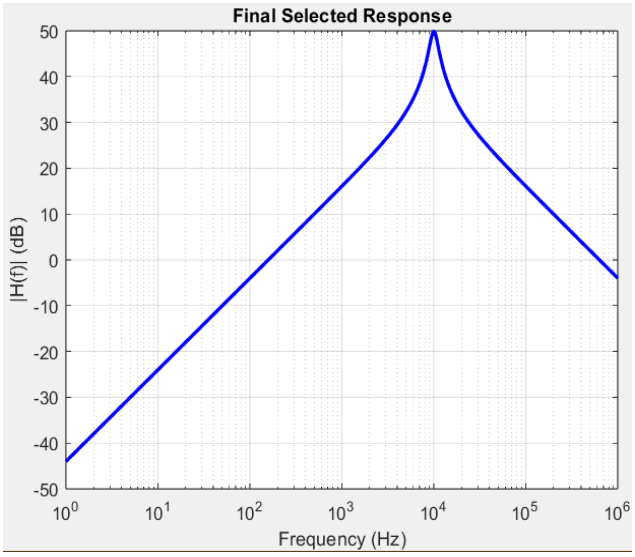


Figure 4.43 Customer's Final Selection

Once a curve has been chosen, the modulation option is activated. If the user decides not to apply modulation, the neural network analyzes the original curve to determine its optimal parameter values. Conversely, if modulation is applied, the neural network trains the altered curve to obtain its optimized parameters.

Case 1: If the user selects "No Modulation," the system displays the predicted parameters, the generated curve, and the associated error matrix (See: Figure 4.44).

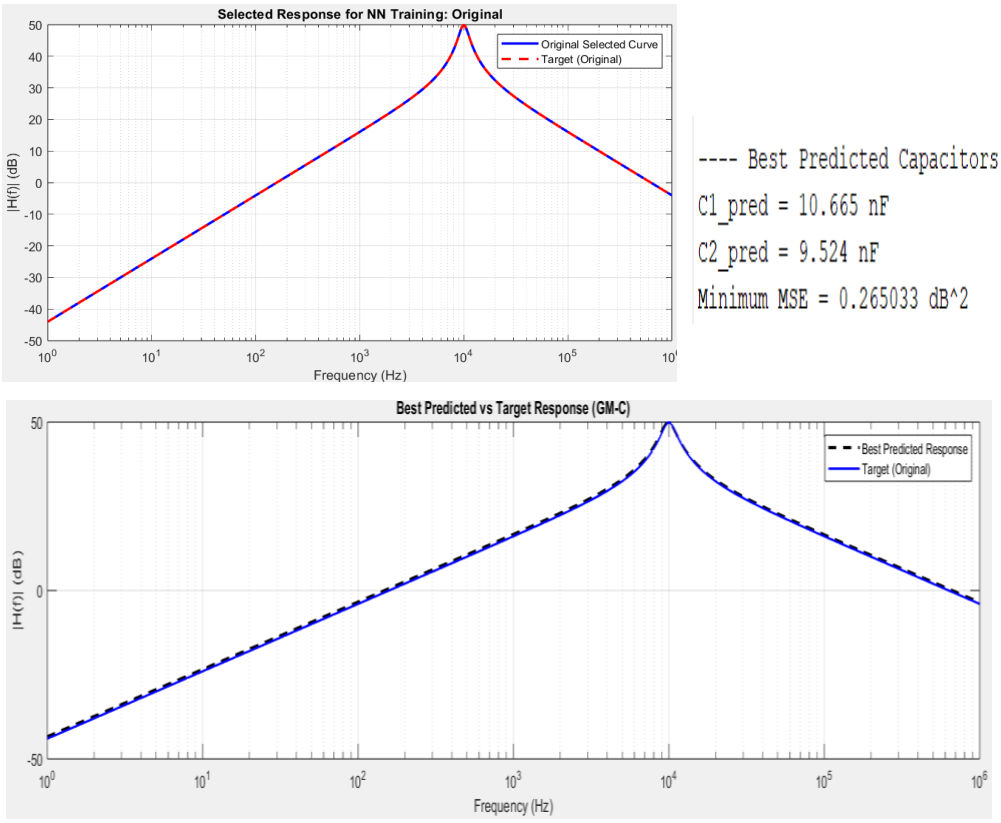


Figure 4.44 Gm-C BPF No Modulation Prediction

The error matrix for C_1 and C_2 is also presented below (See: Figure 4.45).

C1_True	C1_Pred	C1_Error	C1_Error_percent	C2_True	C2_Pred	C2_Error	C2_Error_percent
9e-09	9.437e-09	4.3701e-10	4.8556	9e-09	8.7239e-09	-2.7607e-10	-3.0674
9e-09	8.3101e-09	-6.8993e-10	-7.6659	1e-08	9.1019e-09	-8.9806e-10	-8.9806
9e-09	8.3261e-09	-6.7394e-10	-7.4882	1.1e-08	9.6301e-09	-1.3699e-09	-12.454
1e-08	9.9518e-09	-4.8155e-11	-0.48155	9e-09	8.6995e-09	-3.0048e-10	-3.3387
1e-08	1.0248e-08	2.4789e-10	2.4789	1e-08	9.1386e-09	-8.614e-10	-8.614
1e-08	9.0159e-09	-9.8413e-10	-9.8413	1.1e-08	1.0009e-08	-9.9093e-10	-9.0084
1.1e-08	1.2083e-08	1.0831e-09	9.8461	9e-09	8.7798e-09	-2.2016e-10	-2.4462
1.1e-08	1.0665e-08	-3.3458e-10	-3.0416	1e-08	9.5237e-09	-4.7626e-10	-4.7626
1.1e-08	9.3384e-09	-1.6616e-09	-15.105	1.1e-08	1.0316e-08	-6.8401e-10	-6.2182

Figure 4.45 No Modulation Error Matrix

Case 2: When the user opts for modulation, the curve may be modified using Sine, Cosine, or Triangle functions (See: Figure 4.46). During modulation, the phase value must be selected within the range of 25π to 100π; in this instance, a phase of **25π** is used. The ripple amplitude is specified as **1 dB**.

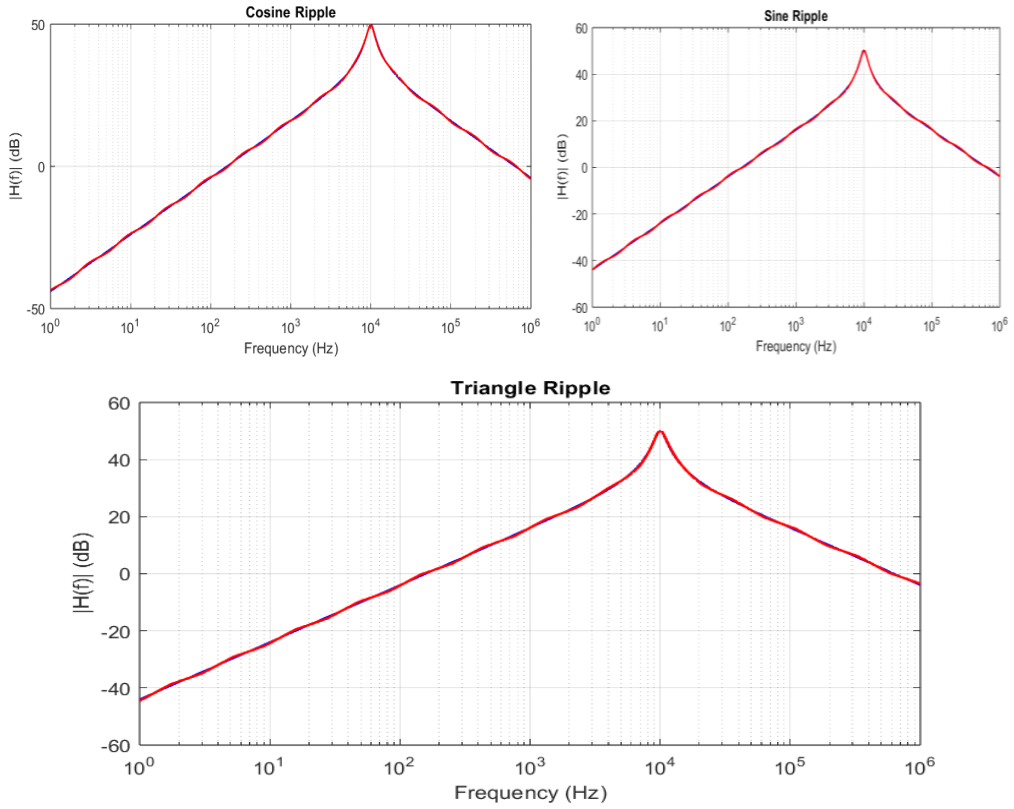
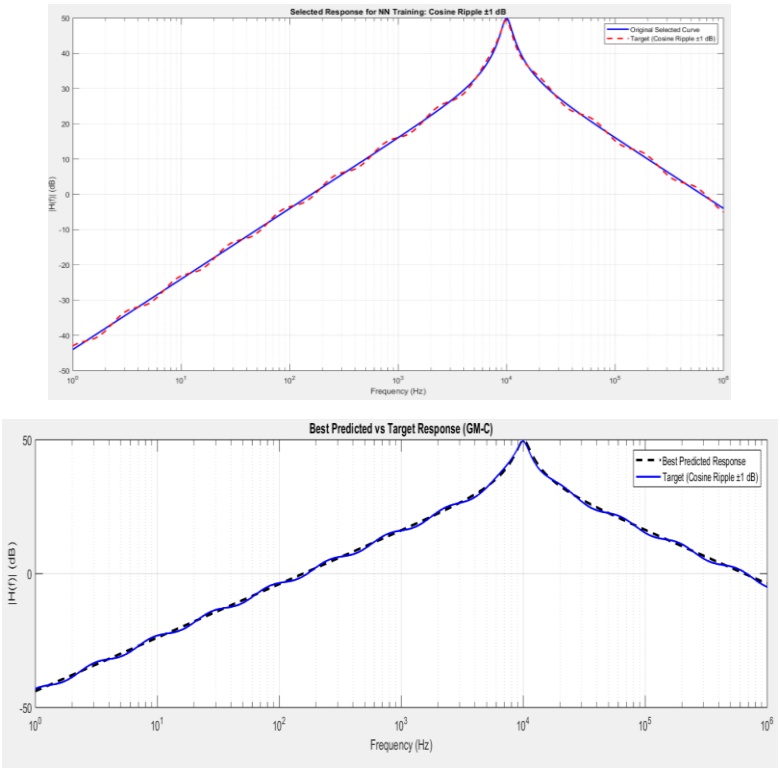


Figure 4.46 Gm-C BPF Modulations Catalogue At 25***π***

After examining the modulation catalogue, the user selects the Cosine modulation method. The neural network subsequently trains the resulting modulated signal (See: Figure 4.47). This process ensures that the modulated signal achieves optimized performance according to the selected parameters.




```
---- Best Predicted Capacitors ----
C1_pred = 10.117 nF
C2_pred = 9.734 nF
Minimum MSE = 0.542016 dB^2
```

Figure 4.47 Gm-C BPF Prediction Using Cosine Modulation

The error matrix for the modulated curve's values is also presented below (See: Figure 4.48).

C1_True	C1_Pred	C1_Error	C1_Error_percent	C2_True	C2_Pred	C2_Error	C2_Error_percent
9e-09	9.1622e-09	1.6225e-10	1.8027	9e-09	1.0655e-08	1.6545e-09	18.384
9e-09	1.0104e-08	1.1041e-09	12.268	1e-08	1.1361e-08	1.3612e-09	13.612
9e-09	1.0567e-08	1.5668e-09	17.409	1.1e-08	1.2025e-08	1.0254e-09	9.3222
1e-08	1.0117e-08	1.1669e-10	1.1669	9e-09	9.734e-09	7.3399e-10	8.1554
1e-08	1.0491e-08	4.9149e-10	4.9149	1e-08	1.0387e-08	3.8713e-10	3.8713
1e-08	1.0906e-08	9.0593e-10	9.0593	1.1e-08	1.0841e-08	-1.5939e-10	-1.449
1.1e-08	1.0892e-08	-1.0754e-10	-0.9776	9e-09	8.2776e-09	-7.2244e-10	-8.0271
1.1e-08	1.1603e-08	6.0258e-10	5.478	1e-08	9.2387e-09	-7.6135e-10	-7.6135
1.1e-08	1.1599e-08	5.9858e-10	5.4416	1.1e-08	1.0055e-08	-9.4509e-10	-8.5917

Figure 4.48 Error Matrix of Cosine Modulation

4.2.6 NMOS

In this study, Chapter 3 presented five case-study designs developed as part of the proposed methodology. Chapter 4 reported the corresponding numerical results and performance analyses. While the first four designs were fully implemented, analyzed, and validated, the fifth design was deliberately left incomplete.

This final design is therefore identified as an open direction for future work. Its detailed implementation, optimization, and evaluation are recommended for future researchers who may wish to extend the present study. This approach not only preserves the continuity of the research framework but also provides an opportunity for further refinement and exploration beyond the scope of the current thesis.

4.2.7 RC Lowpass Filter (ERT)

The circuit parameters are input via the user interface (See: Figure 4.49).

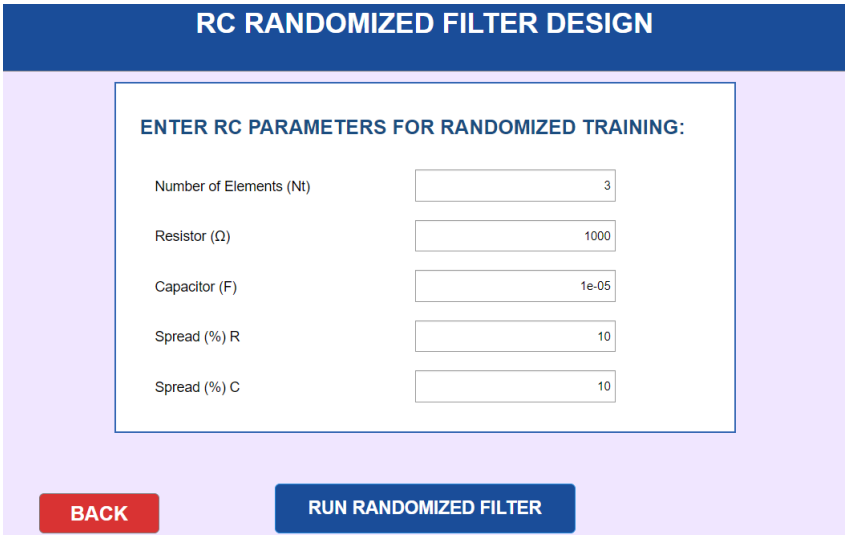


Figure 4.49 RC LPF ERT GUI

Table 4.6 presents the random sweeping parameters along with their corresponding cut-off frequencies.

Table 4.6 RC LPF ERT Element Matrix

R (kΩ)	C (μF)	$f_0\ (rad/s)$
1.0629	10.93	13.699
1.0812	9.3152	15.803
0.9254	10.941	15.719
1.0827	10.914	13.469
1.0265	9.9708	15.551
0.91951	10.601	16.328
0.9557	9.2838	17.938
1.0094	9.8435	16.018
1.0915	10.831	13.462

The magnitude responses are initially presented to enable the subsequent construction of the response catalogue (See: Figure 4.50 and Figure 4.51).

Magnitude Responses

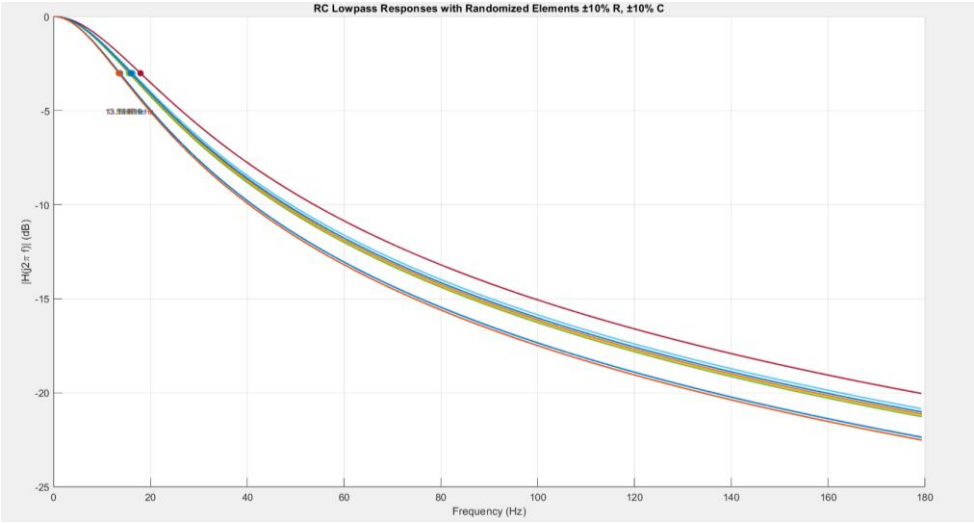


Figure 4.50 RC LPF ERT Magnitude Responses

Catalogue Construction

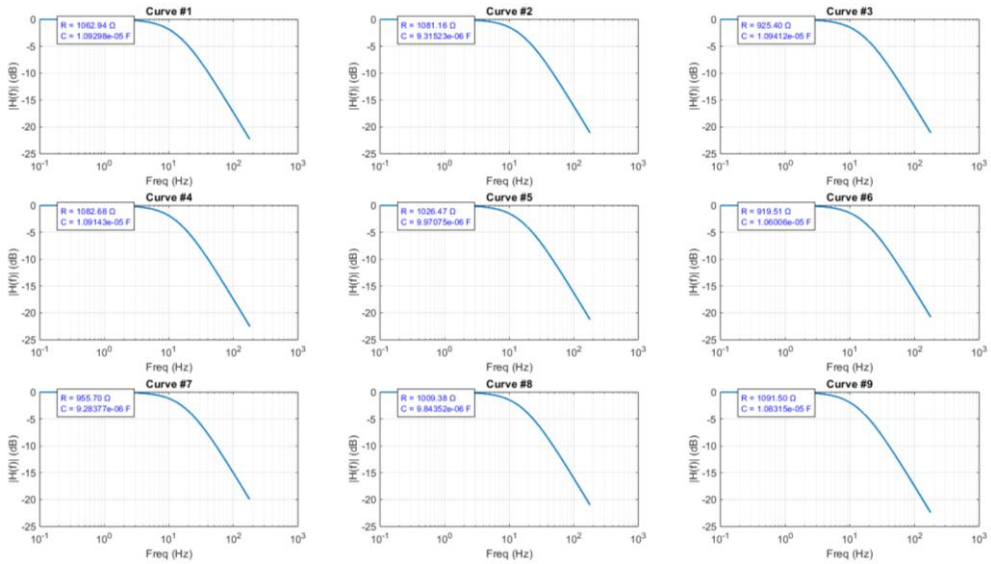
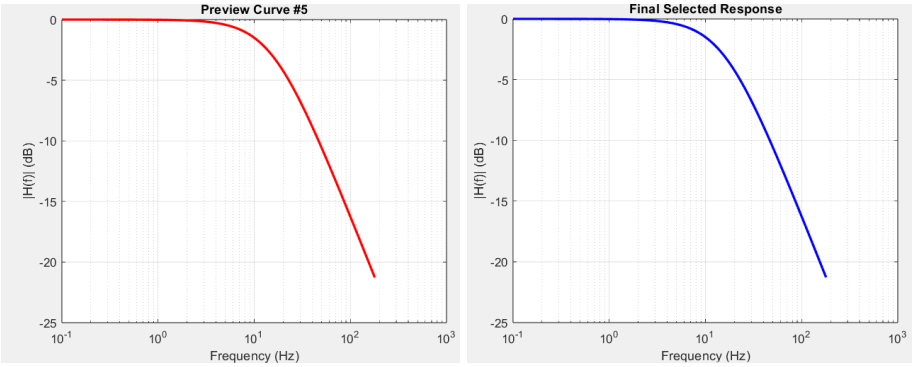


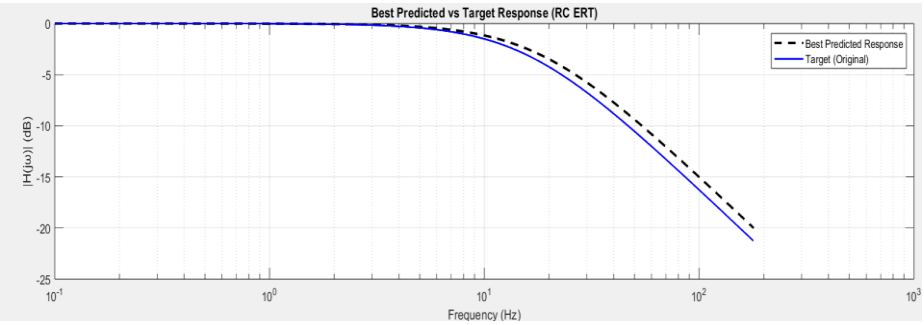
Figure 4.51 RC LPF ERT Catalogue




```
Customer Selected Curve #5:
R = 1.026472 kΩ
C = 9.970751 µF
```

Figure 4.52 Customer's Final Selection

In this phase, the user selects the most appropriate one, after which the corresponding characteristics are displayed. Then, the selected curve is trained through NN without any modulation, and the error is found (See: Figure 4.53).



R_True	R_Pred	R_Error	R_Error_percent	C_True	C_Pred	C_Error	C_Error_percent
1062.9	1235.3	172.34	16.213	1.093e-05	1.0748e-05	-1.8205e-07	-1.6656
1081.2	870.5	-210.66	-19.484	9.3152e-06	1.0082e-05	7.6662e-07	8.2298
925.4	866.58	-58.818	-6.3559	1.0941e-05	1.0183e-05	-7.5792e-07	-6.9272
1082.7	1231.5	148.78	13.742	1.0914e-05	1.076e-05	-1.5401e-07	-1.4111
1026.5	865.09	-161.38	-15.722	9.9708e-06	1.0161e-05	1.9074e-07	1.9129
919.51	893.17	-26.339	-2.8644	1.0601e-05	9.4138e-06	-1.1867e-06	-11.195
955.7	894.7	-60.996	-6.3823	9.2838e-06	9.4087e-06	1.2495e-07	1.3459
1009.4	890.42	-118.96	-11.785	9.8435e-06	9.46e-06	-3.8353e-07	-3.8962
1091.5	1231.4	139.88	12.815	1.0831e-05	1.0761e-05	-7.0905e-08	-0.65462

```
----Best Predicted RC Values----
Predicted R: 0.867 kΩ
Predicted C: 10.183 µF
MSE: 0.416672 dB^2
```

Figure 4.53 ERT LPF ANN Prediction

CONCLUSION AND SUGGESTIONS

CONCLUSION

This thesis presented a comprehensive investigation into the application of Artificial Neural Networks (ANNs) for the design and analysis of analog electronic circuits, with a particular emphasis on passive and active filter topologies. The proposed framework successfully integrates classical circuit theory with data-driven machine learning techniques to enable automated, efficient, and accurate circuit **parameter estimation** based on frequency response characteristics.

EST provided uniform and deterministic coverage of the design space, ensuring bias-free learning and stable performance for circuits with a limited number of parameters. In contrast, **ERT** implemented through Monte Carlo randomization—introduced stochastic variability that closely reflects real-world component tolerances and manufacturing variations. The results demonstrated that **ERT** enhances neural network generalization, particularly for higher-dimensional and tolerance-sensitive circuit designs, while **EST** remains highly effective for structured exploration and catalogue-based design workflows.

The proposed methodology was validated across a diverse set of circuit topologies. **MATLAB**-based analytical modeling and **Proteus** simulations confirmed the accuracy of the predicted component values and frequency responses. Additionally, the introduction of controlled modulation allowed the neural networks to be evaluated under non-ideal and perturbed conditions, providing deeper insight into model robustness and error behavior.

A key contribution of this work is the development of an interactive MATLAB Graphical User Interface (**GUI**), which unifies circuit selection, dataset generation, response visualization, neural network training, and parameter prediction into a single design environment. This interface not only accelerates the design cycle but also enhances transparency and usability, making the framework accessible to both students and practicing engineers.

Overall, the results confirm that neural-network-based circuit design is a powerful and flexible alternative to conventional analog design methodologies. The proposed system demonstrates strong accuracy, adaptability, and scalability, establishing a solid foundation for future research and practical deployment in intelligent electronic design automation.

FUTURE WORKS

One immediate extension involves the complete implementation and optimization of the MOSFET-based (**NMOS**) circuit design that was intentionally left as an open case study in this thesis. A detailed investigation of transistor-level modeling, including biasing conditions, nonlinear effects, and small-signal parameters, would further validate the applicability of the proposed methodology to advanced analog integrated circuit design.

Future studies may also extend the framework to additional analog and mixed-signal circuit topologies, such as low-noise amplifiers, oscillators, phase-locked loops, and multi-stage amplifier systems. Incorporating these circuits would allow the evaluation of the neural network approach under more complex design constraints and higher-dimensional parameter spaces.

Finally, future work may focus on enhancing the developed graphical user interface (**GUI**) by incorporating multi-objective optimization, automated tolerance analysis, and real-time interaction with commercial electronic design automation (EDA) tools. Such improvements would further increase the practicality, scalability, and industrial relevance of the proposed system.

REFERENCES

Lo, W. L., Chung, H. S. H., Hsung, R. T. C., Fu, H., & Shen, T. W. (2023). PV panel model parameter estimation by using neural network. Sensors, 23(7), 3657. https://doi.org/10.3390/s23073657

Chen, H., Li, Q., Ye, Z., & Pang, S. (2025). Neural Network-Based Parameter Estimation and Compensation control for Time-Delay servo system of Aeroengine. Aerospace, 12(1), 64. https://doi.org/10.3390/aerospace12010064

Daylak, F., & Ozoguz, S. (2025). Automated neural Network-Based optimization for enhancing dynamic range in active filter design. Electronics, 14(4), 786. https://doi.org/10.3390/electronics14040786

Boonseng, C., Nilnimitr, N., & Kularbphettong, K. (2019). An investigation on the selection of filter topologies for passive filter applications by neural network. In 2019 International Conference on Power, Energy and Innovations (ICPEI) (pp. 48–51). IEEE. https://doi.org/10.1109/ICPEI47862.2019.8944934

Taye, M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions, 11(3), 52. https://www.mdpi.com/2079-3197/11/3/52

Jafari, A., Sadri, S., & Zekri, M. (2010). Design optimization of analog integrated circuits by using artificial neural networks. In 2010 International Conference on Soft Computing and Pattern Recognition (SoCPaR) (pp. 385–388). IEEE. https://doi.org/10.1109/SOCPAR.2010.5686736

Gencer, F. B., Xhafa, X., İnam, B. B., & Yelten, M. B. (2020). Design and validation of an artificial neural network based on analog circuits. Analog Integrated Circuits and Signal Processing, 106(3), 475–483. https://doi.org/10.1007/s10470-020-01713-x

Schmidt, W. F., Kraaijveld, M. A., & Duin, R. P. W. (1992). Feedforward neural networks with random weights. In Proceedings of the 11th IAPR International Conference on Pattern Recognition: Conference B: Pattern Recognition Methodology and Systems (pp. 1–4). IEEE. https://doi.org/10.1109/ICPR.1992.201708

Hashim, N. Z. B., & Jamuar, S. S. (2014). Gm-C based band pass filter. In 2014 2nd International Conference on Electronic Design (ICED) (pp. 325–328). IEEE. https://doi.org/10.1109/ICED.2014.7015823

Kumar, A., Kumar, A., Kushwaha, S., & Singh, A. K. (2016). Analysis of an analog Sallen-Key band pass filter based on particle swarm optimization technique. International Research Journal of Engineering and Technology (IRJET), 3(6), 1229–1234. https://www.irjet.net/archives/V3/i6/IRJET-V3I6232.pdf

Berzal, F. (2025). Neural network outputs and loss functions. arXiv. https://arxiv.org/abs/2511.05131

Allakhverdiyeva, N. (2016). Application of neural network for digital recursive filter design. In 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT). IEEE. https://doi.org/10.1109/ICAICT.2016.7991720

Singh, M. A., & Thakare, V. B. V. (2015). Artificial neural network use for design low pass FIR filter: A comparison. International Journal of Electronics and Electrical Engineering, 3(3), 216–219. https://doi.org/10.12720/ijeee.3.3.216-219

Mohamed, I. S., Rovetta, S., Do, T. D., Dragicević, T., & Diab, A. A. Z. (2019). Neural network-based modeling and control of dynamic systems: A review. IEEE Access, 7, 124737–124749. https://doi.org/10.1109/ACCESS.2019.2938220

Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. E. (2009). Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 8(7), 579–588. https://www.researchgate.net/publication/228340819_Multilayer_perceptron_and_neural_networks

Chen, Y.-Y., & Fang, S.-Y. (2025). Optimizing analog circuit design through a machine learning-assisted evolutionary algorithm. Electronics Letters, 61(1). https://doi.org/10.1049/ell2.70331

Hathidara, A., & Pandey, L. (2025). Implementing an artificial quantum perceptron. Annals of Computational Physics and Materials Science, 2(1), 1–5. https://doi.org/10.33140/ACPMS.02.01.01

Amirian, M., & Schwenker, F. (2020). Radial basis function networks for convolutional neural networks to learn similarity distance metric and improve interpretability. IEEE Access, 8, 123087–123097. https://doi.org/10.1109/ACCESS.2020.3007337

Schmidt, R. M. (2019). Recurrent neural networks (RNNs): A gentle introduction and overview. arXiv. https://doi.org/10.48550/arXiv.1912.05911

Gan, X., Sun, J., & Li, S. (2019). Wavelet neural network fault diagnosis based on UKF algorithm for Sallen-Key bandpass filter circuit. In 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 575–578). IEEE. https://doi.org/10.1109/ICISCAE48440.2019.221699

Wei, J. H., Mao, W., Fang, H., Zhang, Z., Zhang, J. X., & Lan, B. J. (2020). Advanced MOSFET model based on artificial neural network. In 2020 China Semiconductor Technology International Conference (CSTIC). IEEE. https://doi.org/10.1109/CSTIC49141.2020.9282457

Wu, Y. (2014). Lecture 9. Department of Systems and Computational Biology, Albert Einstein College of Medicine. https://einsteinmed.edu/uploadedFiles/labs/Yaohao-Wu/Lecture%209.pdf

Thakur, S. (2016). A complete Monte Carlo and sensitivity analysis of various elements in analog VLSI circuits. International Journal of u- and e-Service Science and Technology, 9(11), 239–250. https://doi.org/10.14257/ijunesst.2016.9.11.21

Raychaudhuri, S. (2008). Introduction to Monte Carlo simulation. In Proceedings of the 2008 Winter Simulation Conference (pp. 91–100). IEEE. https://doi.org/10.1109/WSC.2008.4736059

Rajesh, R. J., Preethi, R., Mehata, P., & Pandian, B. J. (2015). Artificial neural network based inverse model control of a nonlinear process. In 2015 International Conference on Computer, Communication and Control (IC4) (pp. 91–100). IEEE. https://doi.org/10.1109/IC4.2015.7375581

Geiger, R. L., & Sánchez-Sinencio, E. (1985). Active filter design using operational transconductance amplifiers: A tutorial. IEEE Circuits and Devices Magazine, 1(2), 20–32. https://doi.org/10.1109/MCD.1985.6311946

Sánchez Gaspariano, L. A., Prieto Fuenlabrada, A., & García Delgado, J. (2003). A BiCMOS Gm-C bandpass filter with Q enhancement based on symbolic design. In Proceedings of the 2nd WSEAS International Conference on Electronics, Control and Signal Processing. WSEAS. https://www.researchgate.net/publication/262355617

Ali, H. K., & Abdaljabar, J. S. (2017). Analysis and simulation of active filters using operational transconductance amplifier (OTA). European Scientific Journal, 13(15), 170–184. https://doi.org/10.19044/esj.2017.v13n15p170

Deliyannis, T., Sun, Y., & Fidler, J. K. (1998). Single operational

transconductance amplifier (OTA) filters. In Continuous-time active filter design (1st ed., Chap. 8, p. 38). CRC Press. https://doi.org/10.1201/9781439821879-8

Denisenko, D. Y., Prokopenko, N. N., Ivanov, Y. I., & Butyrlagin, N. N. (2022). Sallen-Key band-pass filters with independent tuning of general parameters. In 2022 Moscow Workshop on Electronic and Networking Technologies (MWENT). IEEE. https://doi.org/10.1109/MWENT55238.2022.9802404

Olalekan, O. B., & Toluwani, O. V. (2017). Sallen-Key topology, MFB and Butterworth in bandpass design for audio circuit design. Asian Journal of Electrical Sciences, 6(1), 23–28. https://doi.org/10.51983/ajes-2017.6.1.1992

Zumbahlen, H., with the engineering staff of Analog Devices. (2008). Analog filters (Chap. 8). In Linear circuit design handbook (pp. 581, 583–585, 587–621, 623–679). Newnes. https://doi.org/10.1016/B978-0-7506-8703-4.00008-0

Harrison, L. (n.d.). An introduction to depletion-mode MOSFETs. Advanced Linear Devices, Inc. https://www.aldinc.com/pdf/IntroDepletionModeMOSFET.pdf

Macukow, B. (2016). Neural networks—State of the art, brief history, basic models and architecture. In IFIP International Conference on Computer Information Systems and Industrial Management (Lecture Notes in Computer Science, Vol. 9842, pp. 3–14). Springer. https://doi.org/10.1007/978-3-319-45378-1_1

McCaffrey, J. (2015, October 11). Parameter Sweeps, or How I Took My Neural Network for a Test Drive. Visual Studio Magazine. https://visualstudiomagazine.com/articles/2015/11/01/parameter-sweeps.aspx

O'Shea, K., & Nash, R. (2015, November 26). An introduction to convolutional neural networks. arXiv. https://arxiv.org/abs/1511.08458

Panasonic Industry. (2018, July 23). Basic knowledge of LC filters. https://industrial.panasonic.com/ww/ds/ss/technical/b4

Corchete, V. (2019, November). RC Filters. ResearchGate. https://doi.org/10.13140/RG.2.2.10456.85764/1

Krennrich, G. (2020, February 10). Experimental Design and Process Optimization with R. https://bookdown.org/gerhard_krennrich/doe_and_optimization/

Ivanova, M. (2021, March 8). Analysis and modelling of CMOS Gm-C filters through machine learning. AIP Conference Proceedings, 2333(1), Article 070011. https://doi.org/10.1063/5.0042232

David, A. (2021, April 22). Single Layer Perceptron and Activation Function – Medium.https://medium.com/codex/single-layer-perceptron-and-activation-function-b6b74b4aae66

Rosebrock, A. (2021, May 12). Introduction to neural networks –PyImageSearch PyImageSearch.https://pyimagesearch.com/2021/05/06/introduction-to-neural-networks/

D. Foote, K. (2021, November 9). A Brief history of Neural networks. Dataversity.https://www.dataversity.net/articles/a-brief-history-of-neural-networks/

Awan, A. A. (2023, August 30). What are neural networks. Datacamp. https://www.datacamp.com/blog/what-are-neural-networks

Oxeltech, T. (2025, August 27). Your Ultimate guide to designing analog filters. Oxeltech. https://oxeltech.de/analog-filters-design-guide/

GeeksforGeeks. (2025, September 30). MultiLayer Perceptron Learning in Tensorflow.GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/multi-layer-perceptron-learning-in-tensorflow/

APPENDICES

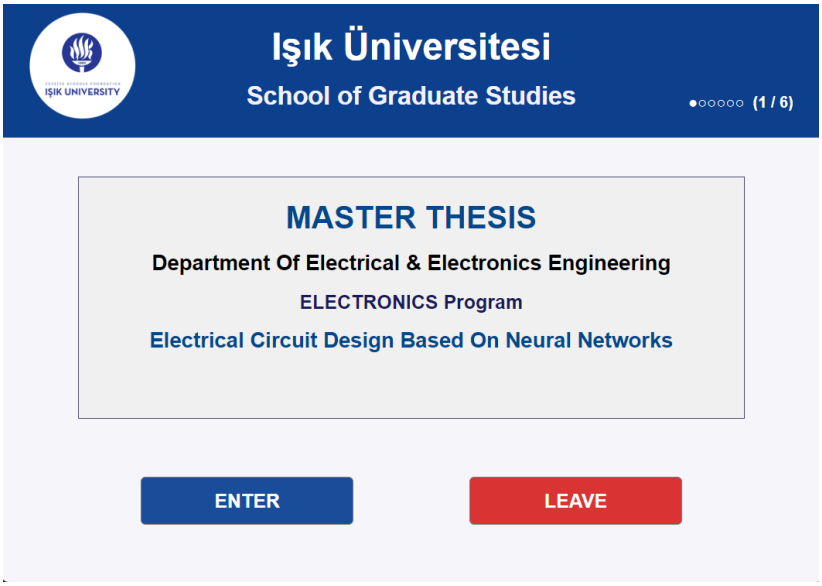


Figure A.1 GUI Main Screen

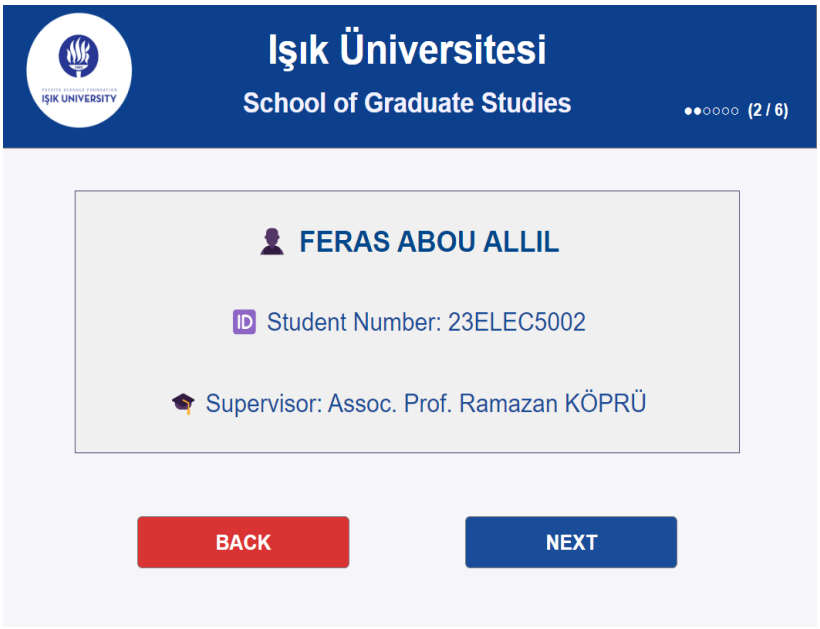


Figure A.2 Author Page

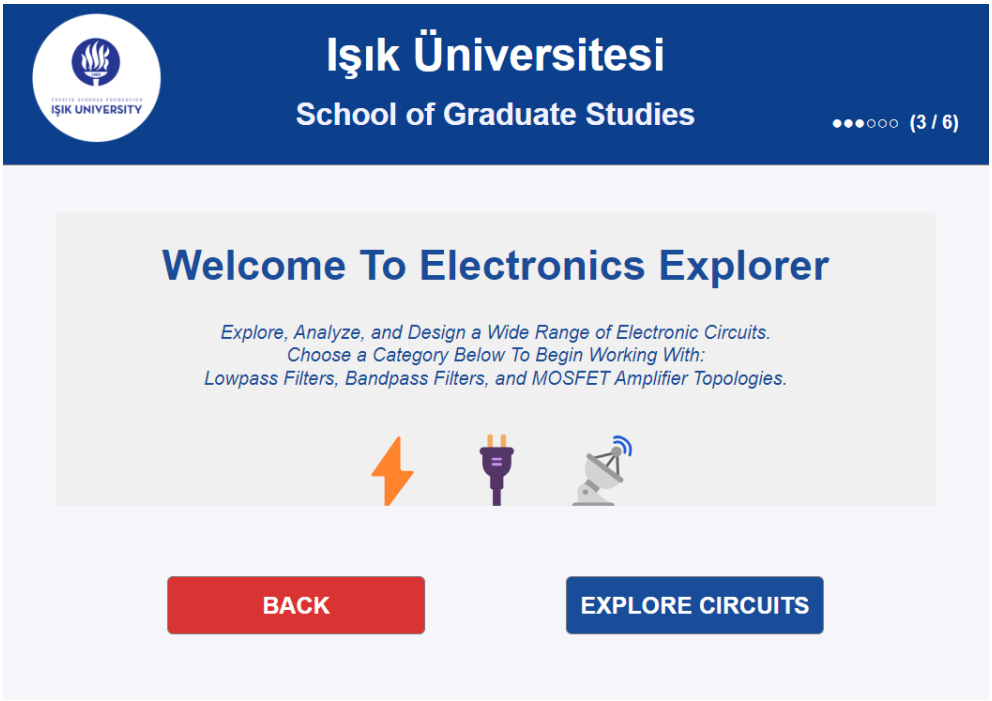


Figure A.3 Introduction Page

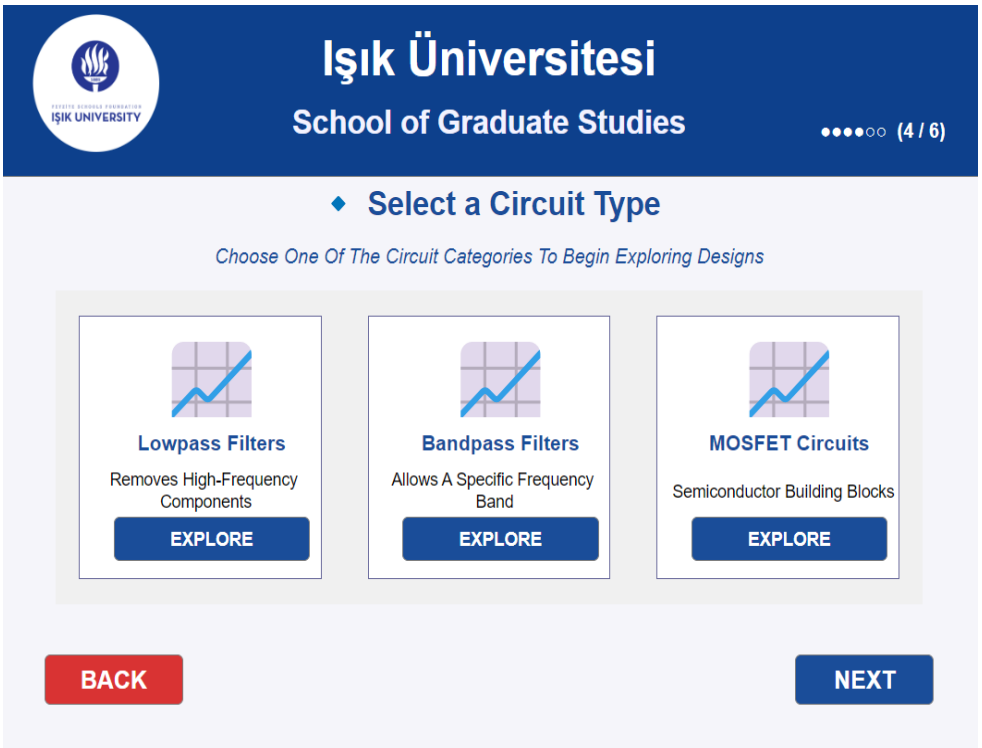


Figure A.4 Categories Page

CURRICULUM VITAE

* "6689 Sayılı Kişisel Verilerin Korunması Kanunu Hükümlerine Göre Çevrimiçi Yayın Dosyasında Bulunan Kişisel Veriler Ve Islak İmzalar Silinmiştir."