Automated Transformer and Active MM-wave Circuit Design

Aarush Goradia

28th April 2025

First Reader/Advisor: Professor Kaushik Sengupta

Second Reader: Professor Minjie Chen

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Electrical and Computer Engineering

Princeton University

I hereby declare that this Independent Work report represents my own work in accordance with university regulations.

I hereby declare that this Independent Work report does not include regulated human subjects research.

I hereby declare that this Independent Work report does not include regulated animal subjects research.

Aarush Goradia

Automated Transformer and Active MM-wave Circuit Design

Aarush Goradia

This paper introduces a neural network (NN)-based framework for automatically designing on-chip transformers that meet specific S-parameter targets. The approach builds on previous work by improving the scripting used to generate random geometries for bridged transformers, ensuring that all transformer design rules are followed. The NNs are then retrained with more data across different transformer types to boost prediction accuracy and make the tool more reliable. Then, this machine learning framework is implemented to streamline the design of on-chip transformers in low-noise amplifiers (LNAs), particularly to help with impedance matching challenges. The process involves converting S-parameters into lumped parameters, which are then used to predict the optimal transformer geometry. This kind of automation can save engineers significant time, allowing them to focus on higher-level design problems instead of manually iterating through transformer layouts.

Acknowledgements

I would like to thank Professor Sengupta for helping me come up with the project and for his guidance. I would like to thank Emir Ali Karahan for helping me realize this project and for all the help.

Table of Contents

1. Introduction

The design of on-chip transformers plays a critical role in the performance of radio frequency (RF) integrated circuits, especially in applications such as low-noise amplifiers (LNAs) where impedance matching is essential. However, designing transformers that meet specific performance criteria—typically expressed through S-parameters—remains a time-consuming and iterative process. This work improves upon earlier work done on training neural networks (NN) to predict ideal transformer geometries based on inputted lumped parameters. It then takes that machine learning design and applies it to the design of LNAs to optimize impedance matching and other factors. The aim of this work is to reduce the manual effort involved in the design process of LNAs.

The design methodology used in the original work, called top-down design, is an alternate form of electromagnetic structure design that is being explored by Professor Kaushik Sengupta and the Integrated Microsystems Research Lab (IMRL) at Princeton University [1]. Sengupta's lab has recently been working on the use of this design methodology in generating geometries for impedance matching networks in power amplifiers. This research will use this approach to try to automate the generation of transformers for LNAs in MM-wave circuits. By using a neural network to generate the geometry for the transformer, impedance matching can be optimized to create an efficient LNA that does not require significant human designing, simplifying the circuit design greatly.

Currently, when synthesizing LNAs, designers follow a design methodology that starts with the ideal input and output S-parameters:

1. Given ideal input and output parameters, designers extract ideal lumped parameters for the transformer.

2. Translate lumped model results into physical dimensions.
3. Combine electromagnetic simulation results with LNA circuit models to validate and fine-tune the design [2].

The goal of this project is to remove the human element in this entire process, creating an algorithm that procedurally runs through these steps with defined S-params. The data for training the neural network is taken from Cadence EMX, a simulation software that gives the lumped parameters for a specific geometric configuration. Then, an algorithm is used to convert S-parameters into ideal lumped parameters. This will then take advantage of the neural network to predict an ideal geometry, and so the design process is completely automated.

2. Background

LNAs traditionally act as the first active stage in a receiver. When signals are sent noise and other factors in the environment attenuate the signal as it travels through the air (especially in cases where there is a large distance between sender and receiver). Upon receiving this attenuated signal, LNAs are required in order to amplify the signal to a level where it can be interpreted properly. Other active components in the receiver will add noise to the signal, so an LNA needs to be able to amplify the signal while adding a minimal amount of noise to the signal, the gain also needs to be large enough in order to minimize the noise additions from the other stage, so that they do not corrupt the signal [3]. One vital application of LNAs is in wireless communications to satellite systems, where preserving signal integrity directly impacts sensitivity and dynamic range.

Two important metrics of LNAs is the noise figure (NF), and the gain. According to Razavi, the noise figure directly adds to the receiver, with the LNA contributing some amount to the total noise of the system [3]. Moreover, since LNAs require low noise, it means that circuit topologies with only one transistor, which is usually the input device, can be the "dominant contributor" to the NF [3]. Gain is the factor of which the signal is amplified by, usually calculated as the amplitude of the final signal divided by the amplitude of the input signal. Since a LNA is an amplifier, a larger gain is more desirable, since that ensures that the noise contributions of the other components are minimized. But focusing solely on a larger gain while overlooking the NF can be detrimental for the operation of an LNA.

One large consideration that engineers make when designing LNAs is their input matching. There seem to be two main ways of how to make sure that the input matching is efficient. The first is conjugate matching, where:

$$Z_{\text{in}} = Z_{\text{source}}^{*}$$

This form of matching maximizes the power transfer by neutralizing the reactances that create standing waves, ensuring the maximum power absorption by the load. The other form of matching is from a noise point of view, optimum noise matching, which aims to minimize the noise figure of the LNA. This prioritizes Z_{opt}, which is a transistor specific impedance that is optimal for minimizing NF, at the cost of gain.

The quality of the input matching is measured by the term return loss, which is the reflected power divided by the incident power. For source impedance R_s, we can write:

$$\text{Return Loss} = \left|\frac{Z_{in} - R_s}{Z_{in} + R_s}\right|^2$$

where Z_{in} indicates the input impedance. Additionally, according to Razavi, a return loss of about -10 dB is considered acceptable.

The choice of matching is highly dependent on the context of the LNAs use. In cases where sensitivity matters more, like in satellite receivers and 5G, optimum noise matching seems to be the superior option. In cases where amplification matters the most, like in antennas or transmission lines, conjugate matching is the most superior choice because it ensures that the maximum power is transferred. Therefore, when designing LNAs, it seems that using optimum noise matching is a better choice. Due to the fact that standard LNAs are designed with a 50Ω input resistance, the topologies that are available for design are limited, and so cannot be designed from scratch based on certain NF and gain [3].

The LNAs that will be designed in this work, though, are for MM-wave radios in the 60 GHz band. This area has seen some recent innovation, with radios in this frequency range being well suited for “applications such as WPANs and gigabit/s point-to-point links” [4]. Recent studies have used CMOS technologies to develop LNAs, allowing them to easily be placed on chips.

Moreover, according to Yao et al., when scaling down to the nanometer size, the minimum NF seems to improve. In that paper, Yao et al. use spiral inductors rather than transmission lines in the matching network, which seems to increase the quality factor (Q) [4].

3. Methodology

This project will contain two main steps. It will first start with extending prior written code that generates random transformer geometries in Cadence Virtuoso. This prior code generates random geometries for three types of transformers: stacked, interleaved, and bridged as shown below.

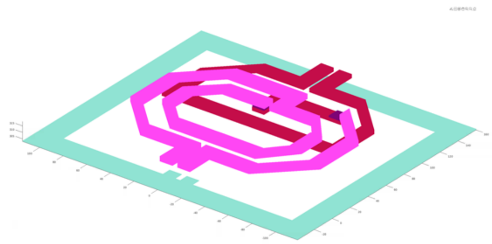

Figure 1: Bridged Transformer

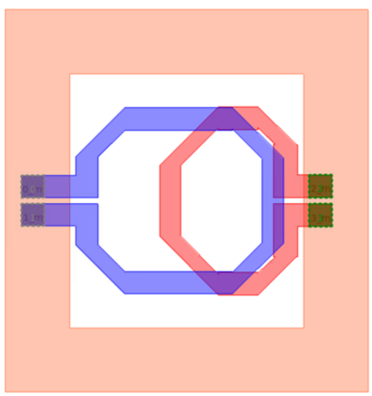

Figure 3: Stacked Transformer

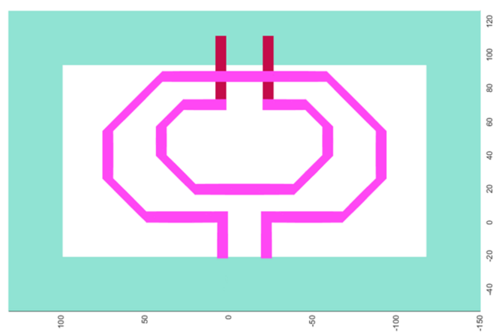

Figure 2: Interleaved Transformer

Initially, the neural network prediction values were not particularly high and so more data points (simulations at 40 GHz) will be taken for the interleaved and stacked transformers, with the neural networks then being re-trained to have better accuracy. Like the last paper, "data [will be] run through a regression model with 3 layers and 64 neurons per layer[s]", using a Leaky ReLu activation layer for the highest accuracy.

The next part of the process is creating the algorithm for the automated LNA design. The actual design of the LNA will be through a 3-stage cascaded network. The first stage will be an input transformer, that matches the source impedance (usually 50 ohms) to the load impedance of the device. The next step is the device, where simulations give S-parameters. Then, the output impedance of the device is matched to a load impedance that has a higher gain. In all of these steps, the S-parameters are optimized as to reduce power lost and minimize noise introduced. This requires a few steps. For one, it requires a python program, that optimizes for $|S_{21}|$ and $|S_{11}|$ parameters on the input and then $|S_{12}|$ and $|S_{22}|$ parameters on the output given the S-params of the device and the source and load impedances. Code will be written that uses scikit-rf to model the transformer and then other libraries used to help optimize 3 lumped parameters, L1, L2, and k, based on this simplified transformer model:

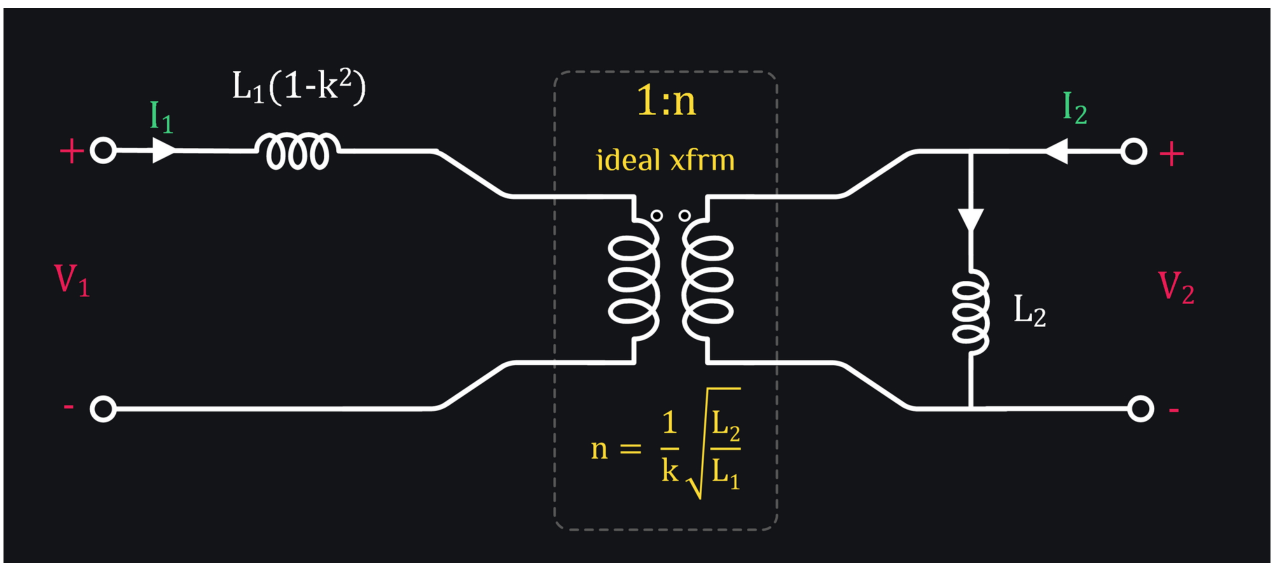

Figure 4: Integrated Transformer Model [5]

This output of the optimizations will then be used to find the ideal geometry using the neural network. This will be done using a basic range search. Each of the parameters for the geometry are in a specific range and so running through those ranges to predict lumped parameters closest to the optimized ones from the algorithm will give the best transformer geometry for the input. This will be done for the output as well and so an optimized transformer geometry will be available for the cascaded LNA. To make sure that the results of the optimizations are feasible, certain test cases will be used in tandem with Cadence's Advanced System Design to see if parameters are optimized and similar to a proper testbench. An example for a source impedance of 50 ohms and a load impedance of 34 - 50j ohms is shown below. Which is derived from the |S11| of a device and then comparing that with the R_{opt} and finding a Z_L:

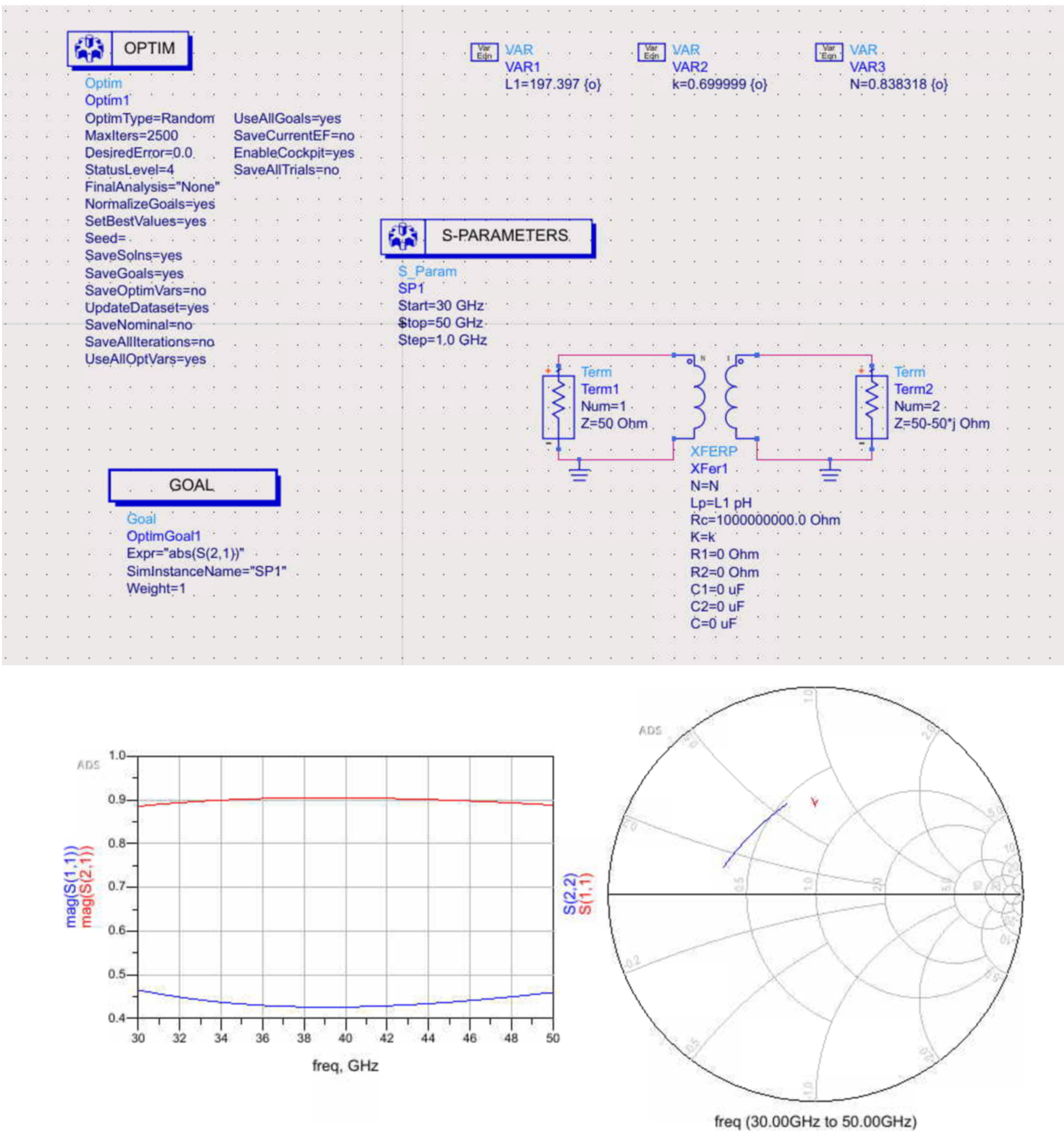

Figure 5: ADS Example

4. Results

The first step of the methodology was successful. Upon re-training a model for the stacked transformer, these results were shown.

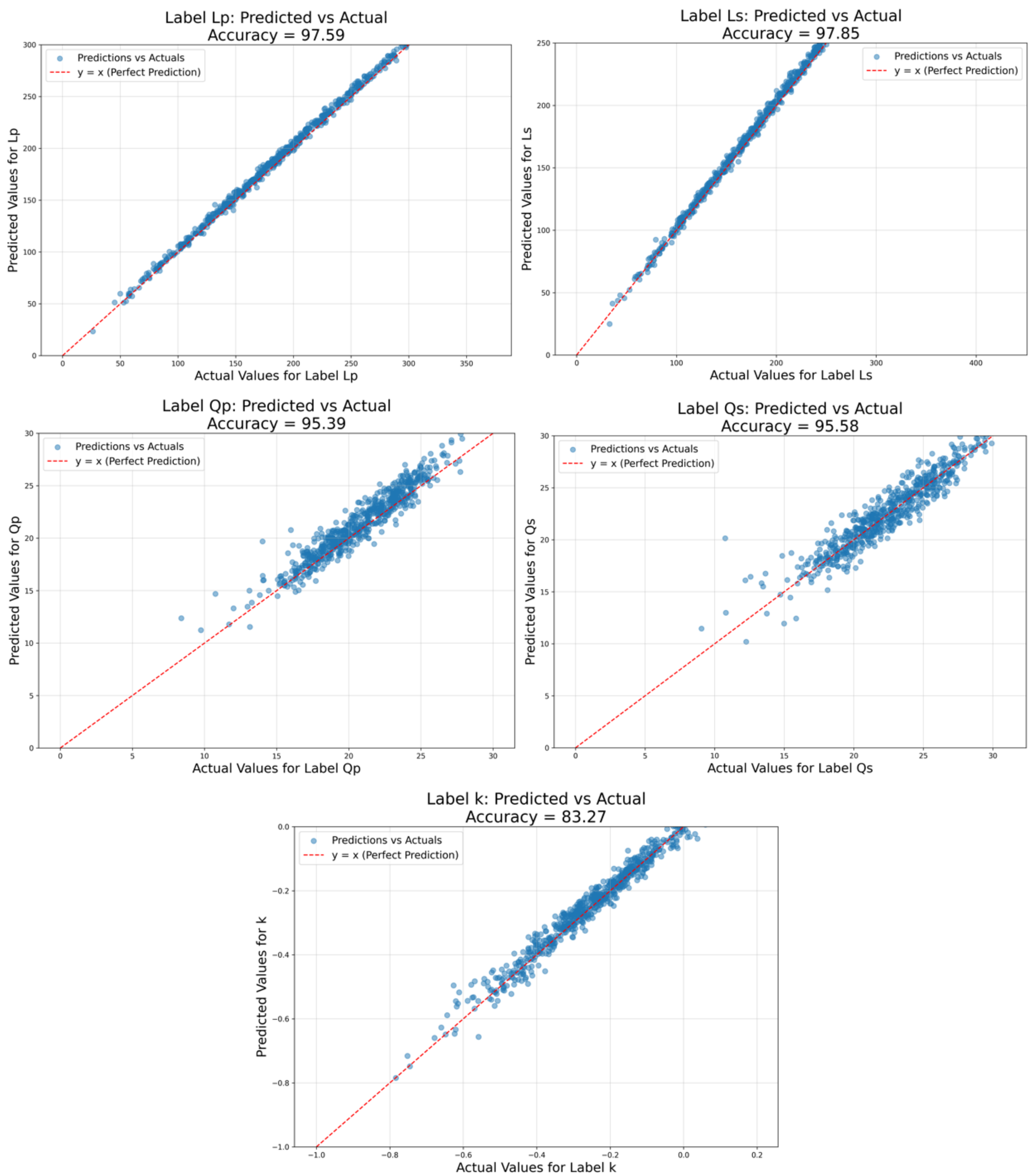

Figure 6: New Stacked Transformer Results

The results below are from the results in the previous work where this model was trained. In the new case, L_p and L_s accuracy are slightly lower, but still high. Q_p and Q_s are extremely similar, but in the new results, K is higher, which is important as that was a point of weakness in the earlier work.

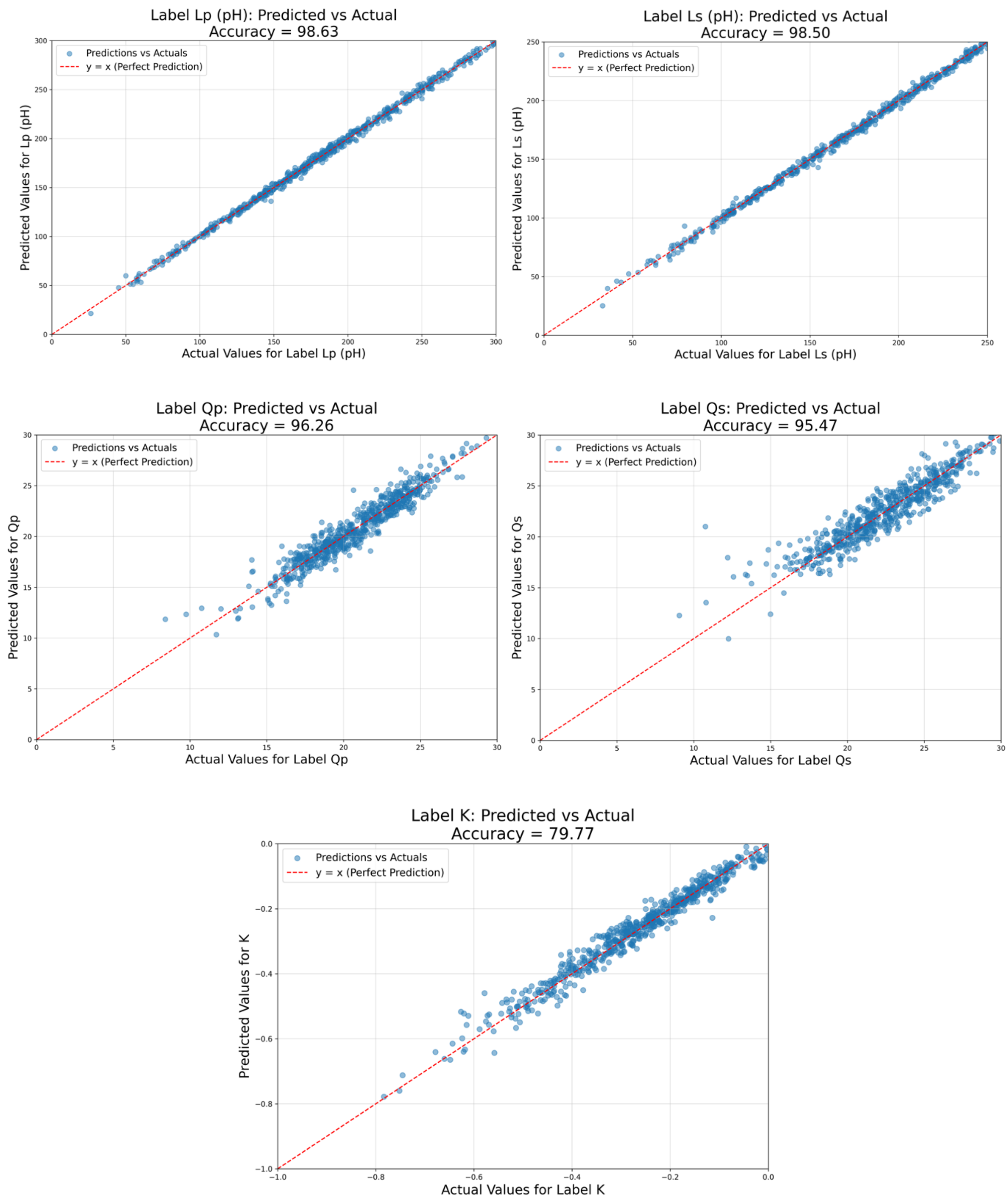

Figure 7: Stacked Transformer Results

In regard to the next part of the project, a model of the transformer was built using scikit-rf, which used a 2-port network to model the transformer based on the model in Figure 4. This was for an input model where input impedance was 50 ohms and load impedance was 50 – 50j ohms. The model predicted these values, as well as these:

L1 = 150.92 pH, L2 = 149.47 pH, k = 0.6900

40.0 GHz: |S11| = 0.5372, |S21| = 0.8435

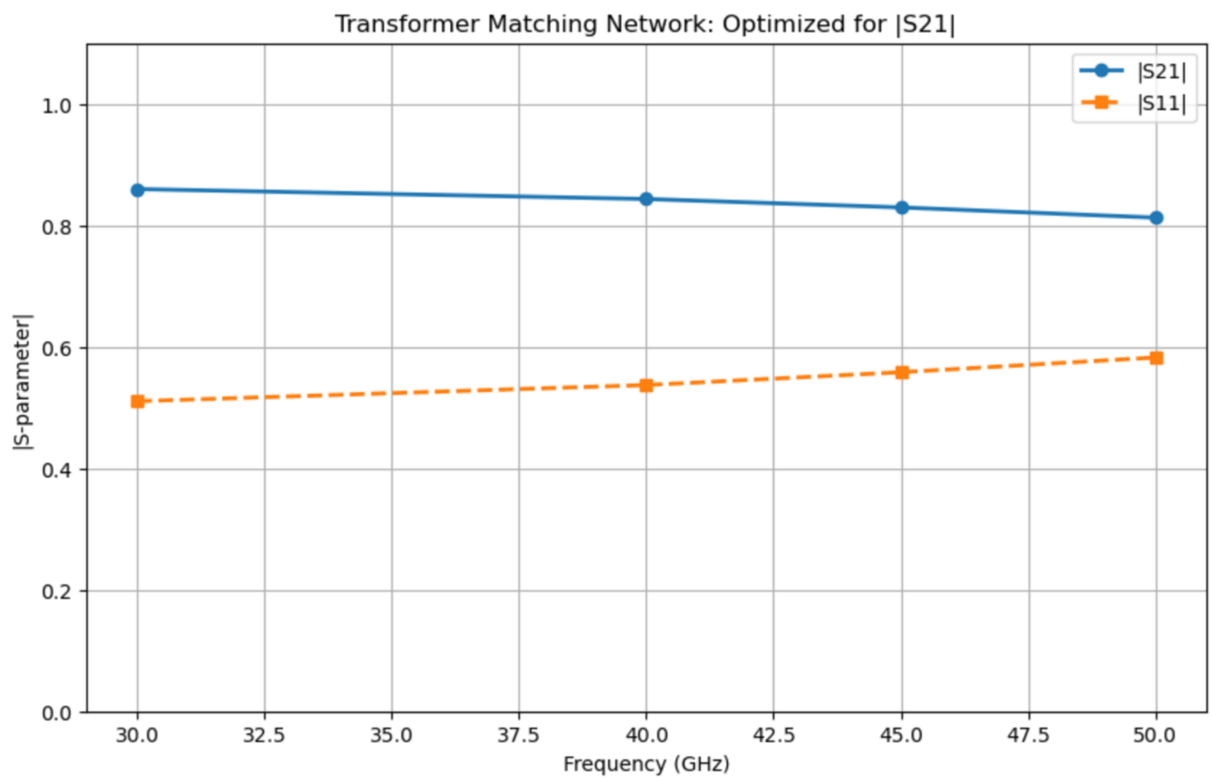

Figure 8: S-params

The values found by simulating a similar situation on EMX ADS gives similar values for S11 and S21 predicted by the model.

A range search was done with the optimal parameters found above using the neural network trained for the stacked transformer. Using a simple range method gave these values:

```
Target (Lp, Ls, k): 1.5092e-10 1.4947e-10 0.69
Predicted (Lp, Ls, k): 1.4205513000488281e-10 2.35255615234375e-10 -0.22062889099121094
Optimized geometry (48-D): [10.08147715 17.50328377  0.         10.29418934  0.         25.11060438
 10.6302384  25.70764881 25.70764881  8.75629777 31.07389452  0.
  8.75629777 31.07389452  0.          9.53930274 29.91280458 29.91280458
 10.36362766  0.         53.10243478 11.29153757 10.02502509 28.30064298
  0.          9.71241152  0.         23.01170581  9.31706133 47.15494144
 47.15494144  9.43474816 23.46240823  0.          9.43474816 23.46240823
  0.          9.63659962 62.90376809 62.90376809  9.54263988  0.
 23.68251956  9.15676125 62.39020107 19.15032149 16.55242657 26.74821641]
```

Figure 9: Range Search Results

Another form of a range search was used, which optimized by searching for good k values first which gave these results:

```
=== SMART LOCAL RESULT ===
Target     (Lp, Ls, k): 1.5092e-10 1.4947e-10 0.69
Predicted  (Lp, Ls, k): 1.3986441040039062e-10 2.626966857910156e-10 0.03283231019973755
Q values   (Qp, Qs):    23.07272 26.395958
Geometry vector (48D): [ 12.48963165  24.92331455   0.          11.96867304   0.
  60.5889736   11.78071308   9.82829533   9.82829533  11.91444312
  31.50402645   0.          11.91444312  31.50402645   0.
  12.46028876  18.69450394  18.69450394  12.44918647   0.
 118.17257559  14.72704561  12.17806693  20.32179774   0.
  12.5706963    0.          52.3838716   12.95376579  33.12847144
  33.12847144  11.95517324  38.65695658   0.          11.95517324
  38.65695658   0.          11.80498532  80.01043048  80.01043048
  12.41037267   0.          25.8593712   14.85554608 102.60683103
  17.46165727  15.86604313  24.05727141]
```

Figure 10: Optimized Range Search Results

These results seem quite off, but the reasoning will be described in the next section.

5. Analysis

As said in the section above, the accuracy of the neural network has gotten slightly better, with the accuracy of the k value increasing from 79% to 83%. This may be for a variety of factors: the activation method, or the focus on k value as more important. This is important because in the earlier paper, the k value was a point of concern because it had a low prediction accuracy. While 83% is an improvement, further improvements still need to be made so that the accuracy of k is comparable to the accuracies of the other parameters. More work still has to be done so that the bridged transformer can give reliable results that can be used to train the neural network, so it has good accuracy (according to data collected from the last project). This is likely from the incorrect algorithm to generate the transformer structures and needs to be improved.

Next, while the results from the transformer model seem to match up with the simulator values, the actual |S21| and |S11| values are not good. A |S21| of 0.84 means that 84% of the power is transmitted from the first to the second port. A value of above 0.9 is considered a good gain, and so this value can be improved. A |S11| value of 0.5372 is extremely large and means that ~53% of the power is reflected back to port 1, which is extremely high. Usually, values under 0.1 are considered good, and so this value can be considerably improved. The fact that these values are un-optimal suggests that there is likely a problem in the construction of the transformer model, or an optimization function that does not focus on the maximization of the |S21| and |S11|. In further experiments, the model can be improved upon to get results that are better.

According to Figure 9, the mismatch between the input target parameters of the range search and the output parameters is quite large. This may be for a variety of factors. For one, this work tries to use 3 out of the 5 output parameters of the neural network to map to the other 48 input parameters. This creates many to one ambiguity, where multiple input combinations could yield similar outputs. Moreover, the neural network is trained in the forward direction (geometry $\rightarrow$ outputs), and so a range search to find the inverse may cause amplification of any errors in the learned representations. One more issue is that the search may also push parameters into regions of the input space that are poorly represented, where the network is bad at guessing. All of these factors tend to lower the accuracy of the output. One reason for the negative k values found may be that the input data had some negative k values in poorly represented areas, causing the network to predict these values.

To combat this, an optimized range search was also tried. This started at better k values so that the k value that the search found would not be negative. The result of this test is shown in Figure 10. While the k value does become a positive value, the values that it found for L_1 and L_2 are much different to the target values and so this search does not seem to perform any better. If given more time, one improvement could be to train a reverse model, where the geometries are predicted from the lumped parameter outputs. Moreover, the other two parameters could have been constrained as well such that the model can find a more closely related match.

6. Discussion

The challenge in designing LNAs is that getting good input matching while also maximizing gain is difficult. With further refinement, the work done in this paper can be applied to real world designs, significantly decreasing the time taken to design these circuits.

While the accuracy can be strongly improved, this project shows promising results of a much more streamlined tool of RF electromagnetic structure synthesizing that could likely make the majority of all RF design significantly more efficient. Output transformer optimization also needs to be done to finish the optimization of the cascaded LNA, but that is similar to the input transformer, just with the optimizations flipped. Further work will be in making sure that these models are trained to a higher accuracy and then creating an end-to-end algorithm that gives the designer a geometry for an input and output transformer based solely on device parameters.

The first part of the project was successful, with the k value prediction accuracy increasing, but the second part of the project was not so successful. Significantly more work needs to be put into the python algorithms such that the transformer is appropriately modeled and can give reliable results no matter the use case.

7. References

[1] E. A. Karahan, Z. Liu, and K. Sengupta, “Deep-Learning-Based Inverse-Designed Millimeter-Wave Passives and Power Amplifiers,” *IEEE J. Solid-State Circuits*, vol. 58, no. 11, pp. 3074–3088, Nov. 2023, doi: 10.1109/JSSC.2023.3276315.

[2] Z. Liu, T. Sharma, C. R. Chappidi, S. Venkatesh, Y. Yu, and K. Sengupta, “A 42–62 GHz Transformer-Based Broadband mm-Wave InP PA With Second-Harmonic Waveform Engineering and Enhanced Linearity,” *IEEE Trans. Microw. Theory Tech.*, vol. 69, no. 1, pp. 756–773, Jan. 2021, doi: 10.1109/TMTT.2020.3037092.

[3] B. Razavi, *RF microelectronics*, 2. edition. in The Prentice Hall communications engineering and emerging technologies series. Upper Saddle River, NJ Munich: Prentice Hall, 2012.

[4] T. Yao *et al.*, “Algorithmic Design of CMOS LNAs and PAs for 60-GHz Radio,” *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1044–1057, May 2007, doi: 10.1109/JSSC.2007.894325.

[5] RFInsights, “Integrated Transformer Models.” Jan. 01, 2023. [Online]. Available: https://www.rfinsights.com/insights/design/concepts/rfic-transformer-model/