

AI Term Project Progress Report

Eric Wilson
Advanced Radar Research Center
The University of Oklahoma
Norman, Oklahoma, United States
Eric.J.Wilson-1@ou.edu

—, —,
—
—
—
—

Abstract—This class project aims to implement a machine learning based method of designing electrical analog filters using an unbalanced-ladder topology. The AI method developed is compared to standard filter prototypes using G-coefficients and element scaling.

Keywords—AI in Analog Design, Filter Synthesis, G-coefficients, Filter prototypes, Element Scaling

I. PROBLEM STATEMENT

My class presentation covered the recent developments in RF and microwave circuit design methods using AI. The basics of the RF design process were explained, and the potential for limited AI assistance at each step was briefly discussed [1]. Next, the distinction between parameter optimization and topological design was made. [2] was used to demonstrate AI's ability to balance the many parameters required in an RF design whose topology is fixed. Then, the topological AI design algorithms described in [3], [4], [5] were described and methods compared. Particular attention was paid to the wavelet transform used in [3] and how it aids in inverse mapping. Additionally, the two-step learning method and fully connected layer dropout operations in [4] was discussed in depth.

My class project on the other hand involves training my own AI model which takes in circuit performance as an input (i.e. frequency vs. magnitude/phase values) and generates the circuit element values in a ladder topology to reach the desired performance. With a ladder topology circuit design (of high enough order) almost any circuit response can be generated, but choosing each circuit element's type and location in the network is complex. This is a great application of AI because it is easy to calculate the performance of a design, but difficult to determine which values are needed to achieve the circuit characteristics that are desired. The values that are determined by the AI can be compared with standard human design methods such as filter prototypes and scaling using G-coefficients.

II. BACKGROUND

Understanding the standard methods designers use for filter synthesis is critical to evaluating whether using AI for filter design is actually an advantage. The basics of filter synthesis is described below.

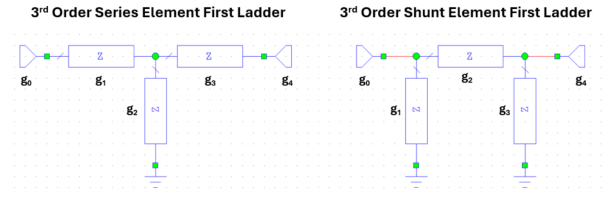


Fig. 1. 3rd order ladder representation: comes in series element first or shunt element first.

In analog filter design the unbalanced-ladder topology for circuit design is often used. Figure 1 shows two versions of a 3rd order ladder circuit.

For standard filter synthesis there are several commonly used designs which dictate where the poles and zeros of the filter lie. Butterworth and Chebyshev filters are common and are synthesized as follows.

— Butterworth Filter Synthesis

$$g_r = 2 \sin \left(\frac{(2r-1)\pi}{2n} \right), \quad r = 1, \dots, n \quad (1)$$

— Chebyshev Filter Synthesis

— Insertion Loss Ripple

$$L_{A,dB} = -10 \cdot \log \left(1 - \left(10^{-\frac{L_{r,dB}}{10}} \right) \right) \quad (2)$$

— Ripple Factor

$$\epsilon = \sqrt{10^{\frac{L_{A,dB}}{10}} - 1} \quad (3)$$

— coefficient for input port is always 1

$$g_0 = 1 \quad (4)$$

— First Circuit Element

$$g_1 = \frac{2}{\eta} \sin \left(\frac{\pi}{2n} \right) \quad (5)$$

$$\eta = \sinh \left[\frac{1}{n} \sinh^{-1} \left(\frac{1}{\epsilon} \right) \right] \quad (6)$$

— Subsequent Elements

$$g_r g_{r+1} = \frac{4 \sin\left(\frac{(2r-1)\pi}{2n}\right) \sin\left(\frac{(2r+1)\pi}{2n}\right)}{\eta^2 + \sin^2\left(\frac{r\pi}{n}\right)}, \quad r = 1, 2, \dots, n-1 \quad (7)$$

— Load for Odd Order Filters

$$g_{\text{load}} = 1 \quad (8)$$

— Load for Even Order Filters From:

$$|S_{21}(0)|^2 = \frac{4g_{n+1}}{(g_n + 1)^2} = \frac{1}{1 + \epsilon^2} \quad (9)$$

Solve for g_{n+1} :

- For $S_{11}(0) \geq 0$:

$$g_{n+1} = \frac{(\epsilon + \sqrt{1 + \epsilon^2})^2}{1} \quad (10)$$

- For $S_{11}(0) \leq 0$:

$$g_{n+1} = \frac{1}{(\epsilon + \sqrt{1 + \epsilon^2})^2} \quad (11)$$

III. PROJECT PROGRESS

I wrote a program that calculates the G-coefficients for Butterworth and Chebyshev filters using the recursive equations above. I also wrote the scripts that scale those G-coefficients to get electrical component values for inductors and capacitors. Finally, I wrote a circuit simulator that has been checked against actual circuit simulator to verify accuracy. Figure 2 shows the G-coefficients calculated for Chebyshev filter prototypes of different orders.

I also wrote scripts which simulates the filter circuit to measure its frequency response. The results of my circuit simulator were compared against a commercial microwave circuit simulator (Microwave Office AWR). Figure 3 shows that my simulated filter performance exactly aligns with the commercial solver.

These scripts will be used to generate the training data and the circuit simulator will also be used to evaluate model performance.

IV. PROJECT SCOPE

For this project a ladder topology of 6 circuit elements is chosen over a 100x frequency bandwidth. Each circuit element has the following options: open, short, resistor, capacitor, inductor, series LC, or shunt LC. Many randomly generated circuit designs will be generated and performance calculated over the frequency range. This data will be used to train the model. The performance of the model will be based on how well it can design a circuit to match desired performance.

| N | g_1 | g_2 | g_3 | g_4 | g_5 | g_6 | g_7 | g_8 | g_9 | g_10 | g_11 | g_load shunt first |
|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------------------|
| 1 | 0.2152 | 1.0000 | | | | | | | | | | 0.8067 |
| 2 | 0.6923 | 0.5585 | 1.2396 | | | | | | | | | 0.8067 |
| 3 | 0.8794 | 1.1132 | 0.8794 | 1.0000 | | | | | | | | 0.8067 |
| 4 | 0.9588 | 1.2970 | 1.6078 | 0.7734 | 1.2396 | | | | | | | 0.8067 |
| 5 | 0.9984 | 1.3745 | 1.8283 | 1.3745 | 0.9984 | 1.0000 | | | | | | 0.8067 |
| 6 | 1.0208 | 1.4141 | 1.9183 | 1.5475 | 1.7529 | 0.8235 | 1.2396 | | | | | 0.8067 |
| 7 | 1.0346 | 1.4369 | 1.9637 | 1.6162 | 1.9637 | 1.4369 | 1.0346 | 1.0000 | | | | 0.8067 |
| 8 | 1.0436 | 1.4514 | 1.9899 | 1.6503 | 2.0457 | 1.6503 | 1.7992 | 0.8419 | 1.2396 | | | 0.8067 |
| 9 | 1.0499 | 1.4611 | 2.0065 | 1.6698 | 2.0858 | 1.6698 | 2.0065 | 1.4611 | 1.0499 | 1.0000 | | 0.8067 |
| 10 | 1.0544 | 1.4679 | 2.0177 | 1.6820 | 2.1085 | 1.7009 | 2.0851 | 1.6277 | 1.8197 | 0.8506 | 1.2396 | 0.8067 |

| N | g_1 | g_2 | g_3 | g_4 | g_5 | g_6 | g_7 | g_8 | g_9 | g_10 | g_11 | g_load shunt first |
|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------------------|
| 1 | 0.6667 | 1.0000 | | | | | | | | | | 0.5195 |
| 2 | 1.3601 | 0.7066 | 1.9250 | | | | | | | | | 0.5195 |
| 3 | 1.5529 | 1.1058 | 1.5529 | 1.0000 | | | | | | | | 0.5195 |
| 4 | 1.6269 | 1.2051 | 2.3198 | 0.8452 | 1.9250 | | | | | | | 0.5195 |
| 5 | 1.6625 | 1.2436 | 2.4956 | 1.2436 | 1.6625 | 1.0000 | | | | | | 0.5195 |
| 6 | 1.6821 | 1.2626 | 2.5617 | 1.3308 | 2.4304 | 0.8738 | 1.9250 | | | | | 0.5195 |
| 7 | 1.6940 | 1.2733 | 2.5940 | 1.3627 | 2.5940 | 1.2733 | 1.6940 | 1.0000 | | | | 0.5195 |
| 8 | 1.7019 | 1.2801 | 2.6123 | 1.3781 | 2.6527 | 1.3571 | 2.4641 | 0.8841 | 1.9250 | | | 0.5195 |
| 9 | 1.7072 | 1.2846 | 2.6238 | 1.3867 | 2.6805 | 1.3867 | 2.6238 | 1.2846 | 1.7072 | 1.0000 | | 0.5195 |
| 10 | 1.7111 | 1.2877 | 2.6315 | 1.3921 | 2.6960 | 1.4006 | 2.6797 | 1.3671 | 2.4789 | 0.8889 | 1.9250 | 0.5195 |

Fig. 2. Computed Chebyshev G-coefficients. These values have been verified against other sources.

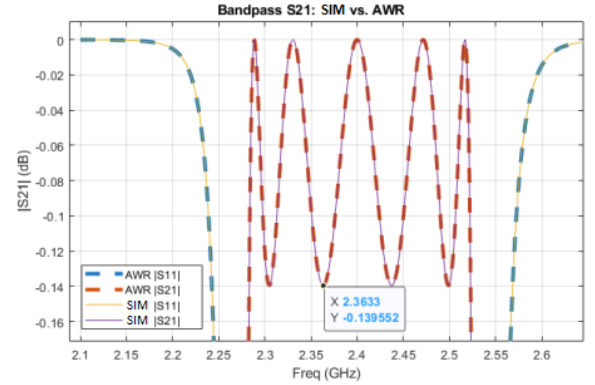


Fig. 3. S21 of a bandpass filter simulated with AWR and my own script. Notice the very zoomed in vertical axis to accurately measure the filter passband ripple. My circuit simulator lines up exactly with the commercial solver.

V. CONCLUSION

A program has been written that can calculate the G-coefficients for Butterworth and Chebyshev filters. More scripts have been written that can scale those G-coefficients to get electrical component values for inductors and capacitors. Finally, I wrote a circuit simulator that has been checked against actual circuit simulator to verify accuracy. These scripts will be used to generate the training data and the circuit simulator will also be used to evaluate model performance.

REFERENCES

- [1] P.-Y. Lee and T.-C. Chen, "Ai-driven innovations in ic designs: From planning to implementation," in *2024 International VLSI Symposium on Technology, Systems and Applications (VLSI TSA)*, April 2024, pp. 1–2.
- [2] L. Xue, H. Fan, Y. Ding, and B. Liu, "A design methodology of mmic power amplifiers using ai-driven design techniques," in *2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, July 2023, pp. 1–4.

- [3] K. Xu *et al.*, “Flexible inverse design of microwave filter customized on demand with wavelet transform deep learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2024.
- [4] E. A. Karahan, J. Zhou, Z. Liu, Z. Shao, S. Fisher, and K. Sengupta, “Deep learning enabled design of rf/mmwave ic and antennas,” in *2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 2024, pp. 769–772.
- [5] H. Taşkıran, E. Sağlıcan, and E. Afacan, “Ann-based analog/rf ic synthesis featuring reinforcement learning-based fine-tuning,” in *2024 20th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, July 2024, pp. 1–4.