Experiment 4

Noise cancellation in headphones

Shounak Das, 21D070068
Prajapati Kishan K, 21D070048
Vinay Sutar, 21D070078

November 2023

Contents

1 Aim

At the end of this experiment, we aim to achieve the following objectives:

- To design and implement an analog circuit for noise cancellation in headphones.
- To achieve an attenuation of 20 dB when a noise of 100 Hz frequency is applied.
- To design an analog compensator to stabilize the system, i.e., loop shaping of the loop transfer function.

2 Procedure

Initially, we conducted an analysis of the headphone setup to derive the output transfer characteristics. Subsequently, utilizing this data, we inputted it into MATLAB to generate a second-order approximation for the system. With this information in hand, the next step involved determining an appropriate compensator, denoted as $C(s)$.

Our specific objective was to select a compensator that, at a frequency of 100 Hz, induces a 20 dB attenuation in the output-to-noise transfer function. It's important to note that the context of this analysis is centered around noise cancellation headphones, where the focus lies in optimizing the system's performance in minimizing unwanted noise.

The open-loop system would look like this:

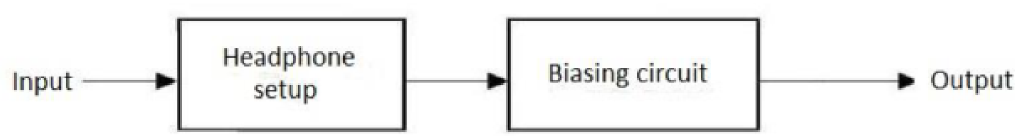

Fig 1: Open loop block diagram

This process underscores the practical application of engineering principles, as we aim to enhance the headphone system's ability to mitigate noise effectively, particularly at the specified frequency of interest. The utilization of MATLAB for system approximation reflects a sophisticated approach to data analysis and system optimization in the realm of experimental noise cancellation.

The closed-loop system would look like this:

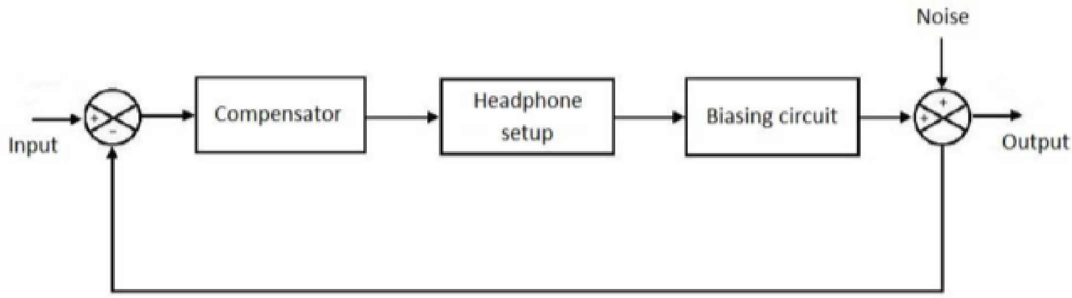

Figure 1: Unity Feedback System for system with Open Loop Transfer function $P(s) = G(s)$ connected to a compensator with open-loop transfer function $C(s)$.

The Biasing circuit is as:

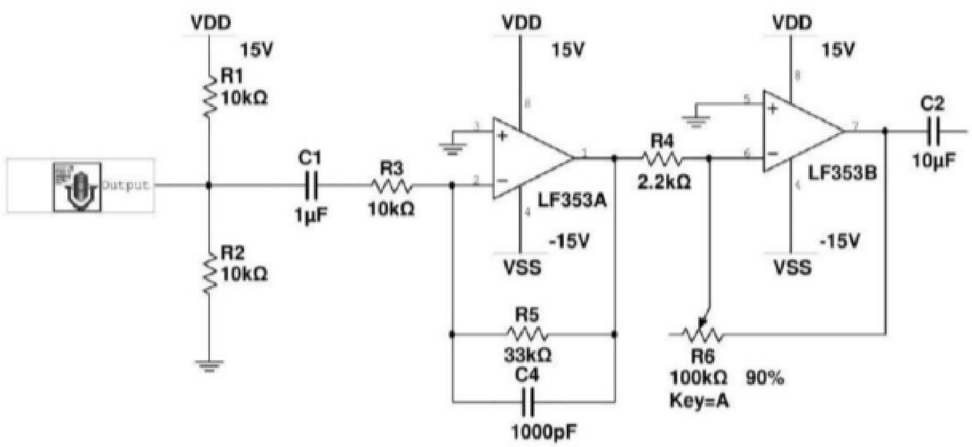

Figure 2: Biasing Circuit

In the given headphone setup, we have meticulously analyzed the relationship between the output $Y(s)$, the input $X(s)$, and the introduced noise $N(s)$, strategically placed at the feedback node. The resulting output transfer functions are defined as follows:

$$Y(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} X(s) + \frac{1}{1 + C(s)G(s)} N(s)$$

To achieve a targeted 20 dB attenuation specifically at 100 Hz, we impose the condition:

$$20 \log \left(\frac{1}{1 + C(s)G(s)} \right) = -20 \quad \Rightarrow \quad C(s)G(s) = 9$$

Simultaneously, we prescribe a gain of -0.91 in the closed-loop input-to-output transfer function:

$$20 \log \left(\frac{C(s)G(s)}{1 + C(s)G(s)} \right) = 20 \log(0.9) = -0.91$$

This translates to a requirement for a gain of -0.91 within the closed-loop transfer function. Ensuring system stability is paramount, and to this end, we set criteria for a Gain Margin > 5 and a Phase Margin $> 30°$. Initially, a 10th-order model was employed for approximating the transfer function $G(s$ based on the observed phase response and gain values extracted from the headphone system.

Subsequently, a judicious analysis of the Bode plot of $G(s)$ informed our decision to design a compensator $C(s)$ incorporating both lead and lag components. This nuanced compensator construction is grounded in a comprehensive assessment of the system's observed characteristics. The overarching goal is to not only meet the specified attenuation requirements but also to enhance overall performance and ensure stability through carefully chosen margin values.

3 Result

Frequency response analysis is performed on the headphone setup with sinusoidal waves given as input, from the function generator. Both the input and output are observed in the DSO, and then the magnitude and phase are plotted versus frequency.

The compensator $C(s)$ would be of the form:

$$C(s) = \frac{k(s + a)(s + b)}{(s + c)(s + d)}$$

Upon careful tuning and observing from the Bode plot of the closed-loop transfer function $\frac{C(s)G(s)}{1+C(s)G(s)}$, we obtain:

$$
\begin{aligned}
k &= 0.02 \\
a &= 4000 \\
b &= 4000 \\
c &= 250 \\
d &= 350
\end{aligned}
$$

This gives us the desired gain of approximately 20dB at 100 Hz for the closed-loop input-output transfer function.

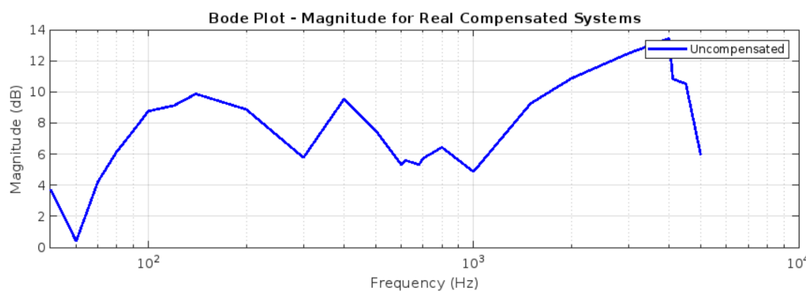

Figure 3: Magnitude Response for origial system

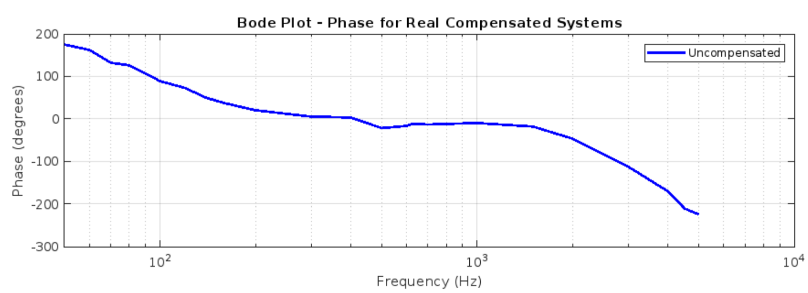

Figure 4: Phase Response for the Observed Data

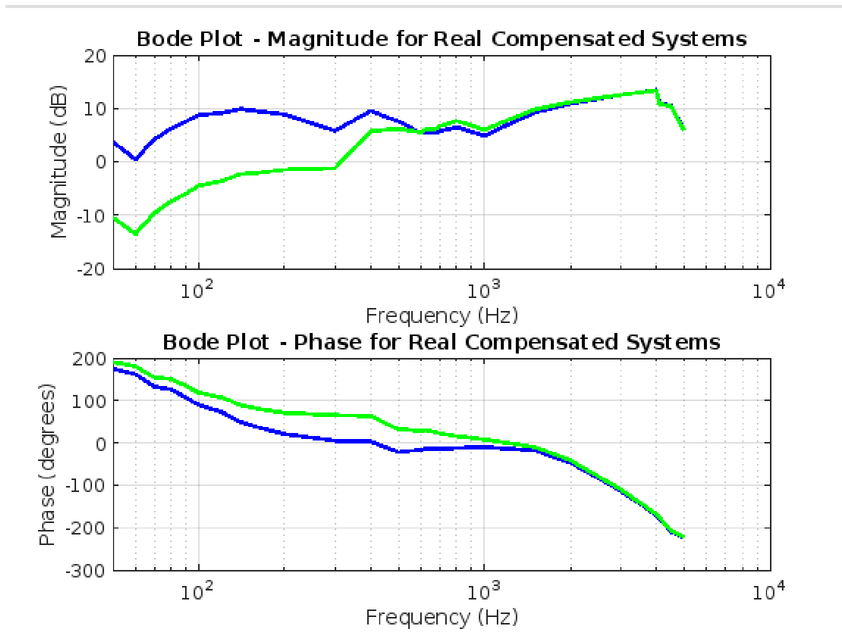

Figure 5: Comparison between Original and compensated bode plots (Blue - Original, Green - Compensated (MATLAB))

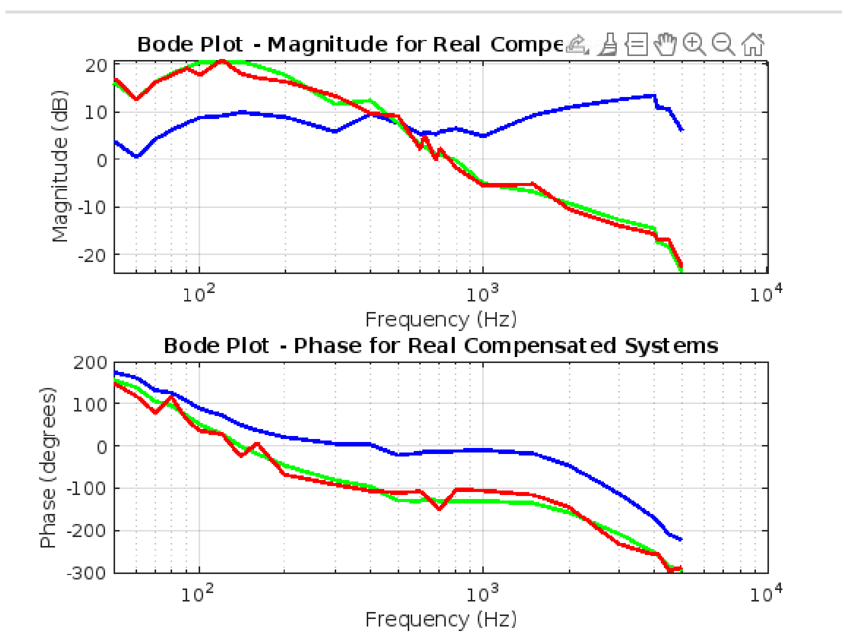

Figure 6: Comparison between simulated compensator and real compensator bode plots (Blue - Original, Green - Compensated (MATLAB), Red - Compensated (Real))

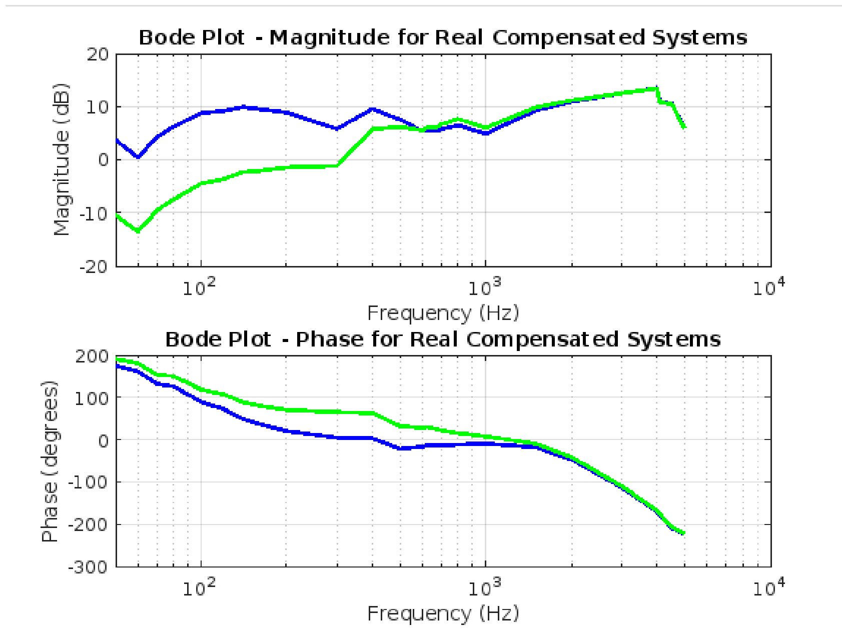

Figure 7: Comparison between closed loop simulated compensator and real compensator bode plots (Blue - Original, Green - Compensated (MATLAB), Red - Compensated (Real))

4 Calculations

- The circuit for 2nd Order compensator given below :-

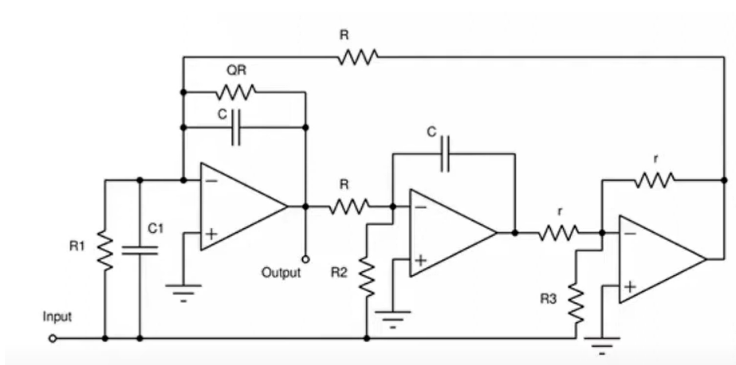

Figure 8: Circuit Diagram

- The Transfer Function of the Compensator is given below :-

$$\frac{V_o(s)}{V_i(s)} = \frac{(C1/C)^2 s^2 + \frac{1}{C}\left(\frac{1}{R1} - \frac{r}{R \cdot R3}\right) s + \frac{1}{C^2 \cdot R \cdot R2}}{s^2 + \frac{1}{Q \cdot C \cdot R} s + \frac{1}{R^2 \cdot C^2}}$$

- The Phase and Magnitude Bode-Plot of compensator is shown below :-

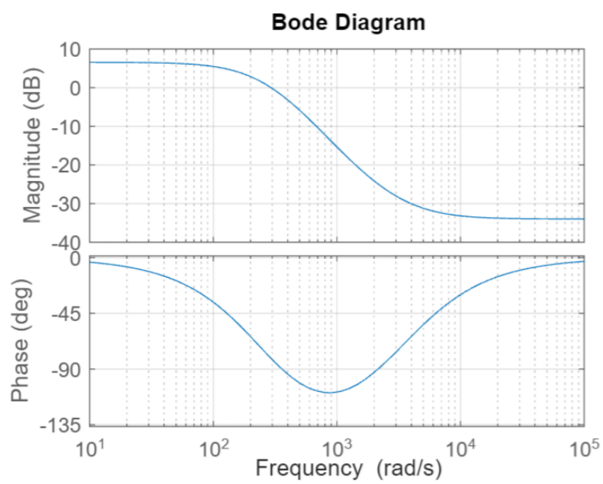

Figure 9: Phase and Magnitude Response of Compensator

- Comparing the Second order compensator transefer function to our transfer function obtained from Bode-Plots , Given in the Results Section we obtained :-

  $R \cdot C = 3.65 \times 10^{-3}$, $Q = 0.497$, $\frac{C1}{C} = 0.1414$, $R2 \cdot C = 8.55 \times 10^{-4}$, $\frac{1}{R1 \cdot C} = 160 + \frac{r}{R3} \times 273.861$

- The Guessed Value used for circuit making are as follows :-

$$\begin{aligned} C &= 220\,\text{Kpf} \\ R &= 16.5\,\text{Kohm} \\ Q \cdot R &= 8.2\,\text{Kohm} \\ C1 &= 31.1\,\text{nf} \\ R2 &= 3.88\,\text{Kohm} \\ R1 &= 10\,\text{Kohm} \\ r &= R3 \quad \text{(can take any value)} \end{aligned}$$

5 Code

Code below gives the bode plot of the compensated open loop system:

```
freq_uncomp = [50, 60, 70, 80, 90, 100, 120, 140, 160, 200, 300, 400, 500,
600, 620, 680, 700, 800, 1000, 1500, 2000, 3000, 4000, 4100, 4500, 5000];
Vin_uncomp = [104, 260, 236, 252, 236, 248, 252, 244, 248, 252, 280, 260,
244, 256, 248, 256, 240, 248, 256, 248, 252, 252, 264, 292, 256, 212];
Vout_uncomp = [160, 272, 384, 512, 560, 680, 720, 760, 740, 700, 544, 780,
580, 472, 472, 472, 464, 520, 448, 720, 880, 1060, 1240, 1016, 860, 420];
phase_uncomp = [175, 162, 132, 126, 107, 88.7, 72.6, 49.4, 36.9, 20.1,
5.1, 3, -22, -16.8, -12.5, -13, -13.6, -12.1, -10, -18.3, -46.8, -113,
-171, -179, -210, -224];

% Real compensated system data
freq_real_comp = [50, 60, 70, 80, 90, 100, 120, 140, 160, 200, 300, 400,
500, 600, 620, 680, 700, 800, 1000, 1500, 2000, 3000, 4000, 4100, 4500,
5000];
Vin_real_comp = [0.56, 0.56, 0.56, 0.6, 0.62, 0.544, 0.54, 0.544, 0.544,
0.544, 0.544, 0.552, 0.552];
Vout_real_comp = [1.68, 2.04, 2.04, 1.84, 1.72, 0.28, 0.3, 0.28, 0.33,
0.376, 0.4, 0.232, 0.104];
phase_real_comp =[148.40, 118.41, 78.84, 115.84, 62.84, 36.83, 27.70,
-24.75, 6.04, -68.77, -91.89, -106.63, -110.73, -107.97, -115.07, -142.49,
-152.32, -104.16, -105.92, -116.65, -145.18, -232.86, -258.25, -254.51,
-296.30, -288.39];

% Calculate magnitude for the three systems
magnitude_uncomp = Vout_uncomp ./ Vin_uncomp;

magnitude_real_comp = [17.00, 12.54, 16.14, 17.69, 19.09, 17.68, 20.77, 17.97, 17.
magnitude_dB_uncomp = 20 * log10(magnitude_uncomp);
magnitude_dB_real_comp = (magnitude_real_comp);

% Simulation
s = zpk('s');
sys = 0.02*(s+4000)*(s+4000)/((s+250)*(s+300));

[mag_tf, phase_tf, wout] = bode(sys, freq_uncomp);
magnitude_dB_simulated=20*log10( magnitude_uncomp .* transpose(squeeze(mag_tf)) );
phase_simulated = phase_uncomp + transpose(squeeze(phase_tf));

% Create Bode plots for magnitude and phase for the three systems
figure;
```

```
% Magnitude plots for all three systems
subplot(2, 1, 1);
semilogx(freq_uncomp, magnitude_dB_uncomp, 'b', 'LineWidth', 2);
hold on;
semilogx(freq_uncomp, magnitude_dB_simulated, 'g', 'LineWidth', 2);
semilogx(freq_real_comp, magnitude_dB_real_comp, 'r', 'LineWidth', 2);
grid on;
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title('Bode Plot - Magnitude for Real Compensated Systems');
%legend('Uncompensated', 'Simulated Compensated', 'Real Compensated');
hold off;

% Phase plots for all three systems
subplot(2, 1, 2);
semilogx(freq_uncomp, phase_uncomp, 'b', 'LineWidth', 2);
hold on;
semilogx(freq_uncomp, phase_simulated, 'g', 'LineWidth', 2);
semilogx(freq_real_comp, phase_real_comp, 'r', 'LineWidth', 2);

%--------------------------------------
grid on;
xlabel('Frequency (Hz)');
ylabel('Phase (degrees)');
title('Bode Plot - Phase for Real Compensated Systems');
%legend('Uncompensated', 'Simulated Compensated', 'Real Compensated');
```

The code below gives the bode plots for the compensated closed loop system:

```
freq_uncomp = [50, 60, 70, 80, 90, 100, 120, 140, 160, 200, 300, 400, 500,
600, 620, 680, 700, 800, 1000, 1500, 2000, 3000, 4000, 4100, 4500, 5000];
Vin_uncomp = [104, 260, 236, 252, 236, 248, 252, 244, 248, 252, 280, 260,
244, 256, 248, 256, 240, 248, 256, 248, 252, 252, 264, 292, 256, 212];
Vout_uncomp = [160, 272, 384, 512, 560, 680, 720, 760, 740, 700, 544, 780,
580, 472, 472, 472, 464, 520, 448, 720, 880, 1060, 1240, 1016, 860, 420];
phase_uncomp = [175, 162, 132, 126, 107, 88.7, 72.6, 49.4, 36.9, 20.1,
5.1, 3, -22, -16.8, -12.5, -13, -13.6, -12.1, -10, -18.3, -46.8, -113,
-171, -179, -210, -224];
```

```
% Calculate magnitude for the three systems
magnitude_uncomp = Vout_uncomp ./ Vin_uncomp;
%magnitude_real_comp = Vout_real_comp ./ Vin_real_comp;

% Convert magnitude to dB
magnitude_dB_uncomp = 20 * log10(magnitude_uncomp);
%magnitude_dB_real_comp = 20 * log10(magnitude_real_comp);

% Simulation
s = zpk('s');

sys = 1 /( 1 + 0.02*(s+4000)*(s+4000)/((s+250)*(s+300)));

[mag_tf, phase_tf, wout] = bode(sys, freq_uncomp);
magnitude_dB_simulated=20*log10( magnitude_uncomp .* transpose(squeeze(mag_tf)) );
phase_simulated = phase_uncomp + transpose(squeeze(phase_tf));

% Magnitude plots for all three systems
subplot(2, 1, 1);
semilogx(freq_uncomp, magnitude_dB_uncomp, 'b', 'LineWidth', 2);
hold on;
semilogx(freq_uncomp, magnitude_dB_simulated, 'g', 'LineWidth', 2);
%semilogx(freq_real_comp, magnitude_dB_real_comp, 'r', 'LineWidth', 2);
grid on;
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title('Bode Plot - Magnitude for Real Compensated Systems');
%legend('Uncompensated', 'Simulated Compensated')%, 'Real Compensated');
hold off;

% Phase plots for all three systems
subplot(2, 1, 2);
semilogx(freq_uncomp, phase_uncomp, 'b', 'LineWidth', 2);
hold on;
semilogx(freq_uncomp, phase_simulated, 'g', 'LineWidth', 2);
%semilogx(freq_real_comp, phase_real_comp, 'r', 'LineWidth', 2);
grid on;
xlabel('Frequency (Hz)');
ylabel('Phase (degrees)');
title('Bode Plot - Phase for Real Compensated Systems');
```

```
%legend('Uncompensated', 'Simulated Compensated')%, 'Real Compensated');
```

6 Problems Faced and Their Solution

Over-Reliance on Simulation Results

Our journey was not without its challenges. The allure of simulated perfection, provided by the SISO tool, posed a potential pitfall. We recognized the need to temper our reliance on simulation results and embark on real-world validations. The transition from the virtual realm to the physical headphone system demanded careful scrutiny and adjustment. This phase of the experiment emphasized the importance of bridging the gap between simulation expectations and the unpredictable intricacies of real-world performance.

Sensitivity to System Changes

A second challenge emerged in the form of the compensator's sensitivity to changes within the headphone system. The initial design, while effective under certain conditions, faced the risk of faltering in the face of unexpected variations. Addressing this sensitivity necessitated a reevaluation of our compensator design strategy. The quest for robustness and adaptability became paramount, leading us to explore compensator configurations capable of dynamically adjusting to evolving system parameters.

Choosing Compensator

We initially used a single lag compensator. Using the SISO tool on MATLAB, we tried obtaining a compensator that met the required specifications. We then realized the need to use 2 compensators, a lag and a lead compensator, which upon tuning with the SISO tool on MATLAB gave us the desired compensator.

In the pursuit of enhanced noise cancellation, these challenges served as catalysts for innovation. Our experiment not only pushed the boundaries of compensator design but also underscored the importance of a holistic and adaptive approach in crafting solutions for complex engineering challenges.

Addressing these potential issues involves a holistic approach to compensator design, encompassing both simulation-based tuning and real-world validation, along with considerations for robustness and adaptability to ensure the designed compensator performs reliably in diverse operating conditions.