IIR Filter applied to Losses in Ferrite Cores

Igor Bertoncello Barboza
GEDRE – Intelligence in Lighting
Federal University of Santa Maria
(UFSM)
Santa Maria, RS, Brazil
igor.b.barboza@gedre.ufsm.br

Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract. (*Abstract*)

Keywords*—*component, formatting, style, styling, insert (key words)

I. Introduction

In 1892, the German engineer Steinmetz defined an equation to estimate the total losses in the main existing cores, through an empiric method [1]. However, nowadays the magnetics are designed at higher switching frequencies and new core materials were developed. Since the cores are made of new materials, excited by distinct waveforms and at higher frequencies, the Steinmetz Equation and your variations do not match the experimental losses found in many power converters anymore. The core total closes – including Hysteresis and Foucault losses – represent the biggest ammount of losses in many power electronics applications, including GaN-based LED drivers [2]. For this reason, the optimized design and losses estimation is essential in order to improve power converters performance. In this context, the University of Princeton, together with the companies Tesla and Google, launched a challenge to obtain the model of core losses. It is known as 2023 Magnet Challenge, which makes avaible a huge ammount of data about the volumetric core losses of many core materials [3].

This work proposes the recognition of an IIR filter function which properly expresses the relation between the volumetric losses P_v (in kW/m^3) and the excitation frequency f (in Hz), for many maximum flux density variations $Bmax$ (in Tesla) and operating temperatures (°C). The selected core material is the ferrite N87, from the company TDK, with the database available in [3].

II. Database Preparation

The data were downloaded from the subpage "MagNet Database", specifing the curves for:

- Triangular excitation, because it happens in mostly inductor currents in power converters;

- At 25°C, since this curve is presented on the datasheet of the N87 ferrite material;

- Frequency range between 50kHz and 447kHz (the widest window available on the database)

- AC flux density range between 10mT and 281mT (the widest window available on the database)

- Duty cycle set to 0.3, because it is the closest value to the Buck converter duty cycle of 0.275 that it is going to be compared soon.

Following the above conditions, the complete data was obtained and it is shown in Fig. 1.

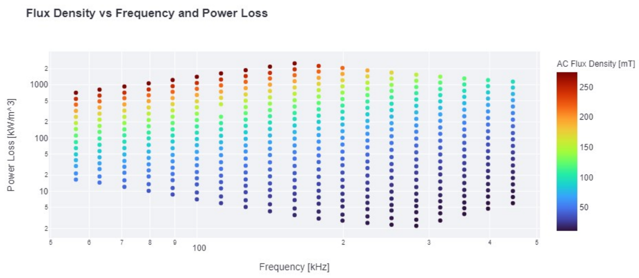


Fig. 1. Official database of the volumetric core losses for the N87 ferrite material, at 25°C, for diferente B_{max} values.

However as it was not feasible to use all the points exposed in Fig. 1, just the points within the curves of 50mT, 100mT, 150mT, 200mT, 250mT and 275mT were used. This is due to the fact that these are the main B_{max} curves found on the datasheets, including the datasheet of the N87 material [4].

After selecting these points, the curves for the abovementioned flux densities were obtained, at 25°C and 50°C, as presented in Fig. 2.

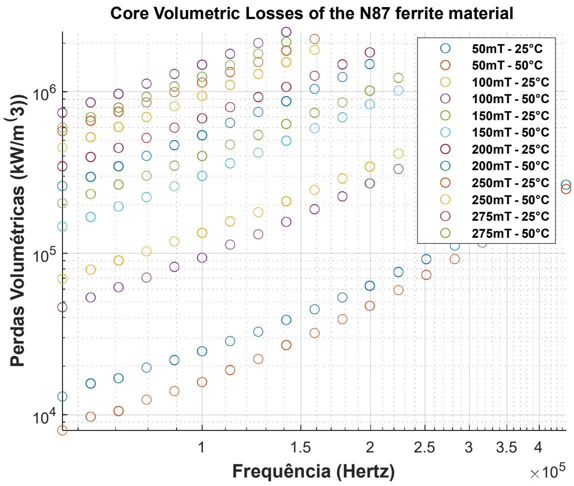


Fig. 2. Volumetric core losses for the N87 ferrite material, at 25°C. Curves of 50mT, 100mT, 150mT, 200mT, 250mT e 275mT.

III. IIR Filter Design

The design was done by interactively checking the proper cutoff frequency f_c and order n of the given IIR filter, which is the Butterworth type.

Through this process, it was found out that a 1st order filter with a cutoff frequency equal to 0.99 was the most feasible solution. This was proven truth not just for the maximum magnetic flux variation B_{max} of 50mT and the temperature T of

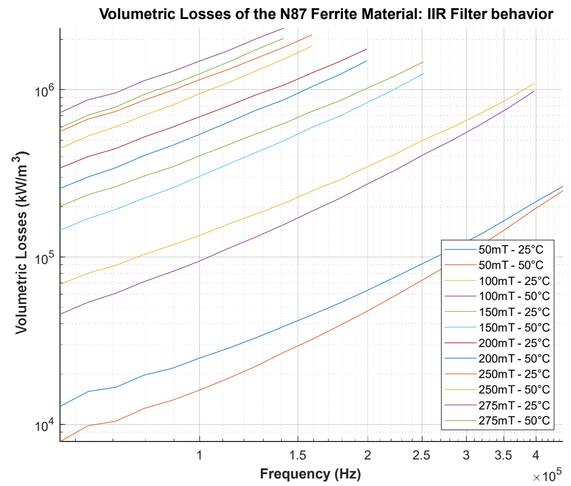
25°C, but also for all the other *Bmax* curves, including 100mT, 150mT, 200mT, 250mT and 275mT, at both 25°C and 50°C.

The equation obtained for the IIR filter *H(s)* is represented by (1). It is an all-pass filter, with one zero and one pole.

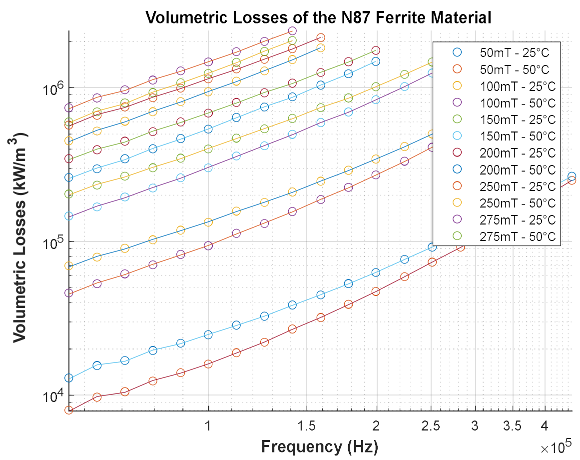
$$H(s) = \frac{0.9845 \cdot s + 0.9845}{s + 0.9691} \tag{1}$$

IV. Application of the IIR Filter Curves

Applying the transfer function of the IIR filter *H(s)* to each curve, presented in Fig. 2, the results of the application of the IIR all-pass filter are observed in Fig. 3(a), being that they can be compared to the original responses in Fig. 3(b).



(a)



(b)

Fig. 3. Results of the application of the IIR all-pass filter on the B_{max} curves. (a) IIR filter in each curve; (b) Comparison of the IIR filter response with the original data of each B_{max}.

V. IIR Filter response to the Impulse Function

The initial goal of this work was to obtain different IIR filters that could reproduce the behavior, each one of them of a distinct B_{max} curve, at 25°C or 50°C operating temperature. So a series of impulses should be the input, the filter the plant itself and the output is the curve which nearly approximates the original behavior, illustrated in Fig. 2.

But, unfortunately, the design of IIR filters of many orders *n* and at different cuttoff frequencies f_c did not result in enough approximations for any of the desired B_{max} curves.

Briefly describing the attempted process, the filter order was varied, without changing the cuttoff frequency f_c, from 2 (a second order filter, with a 40dB/decade attenuation) to 100 (a one-hundrer order filter, with and 1960dB/decade attenuation). After this, keeping a same order *n* of the filter, the cuttoff frequency f_c was varied from 0.1 to 0.99 of the normalized excitation frequency (defined as an input vector of the program code).

Aiming to illustrate the process described above, Fig. 4 shows the original and obtained curves using this method.

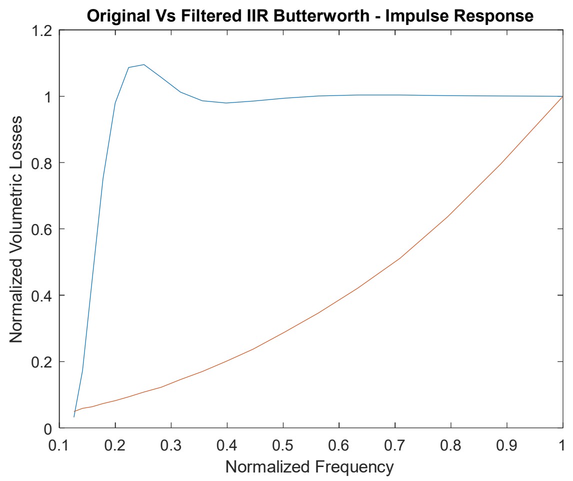


Fig. 4. Original and estimated curves, using the response of the IIR filters to the impulse.

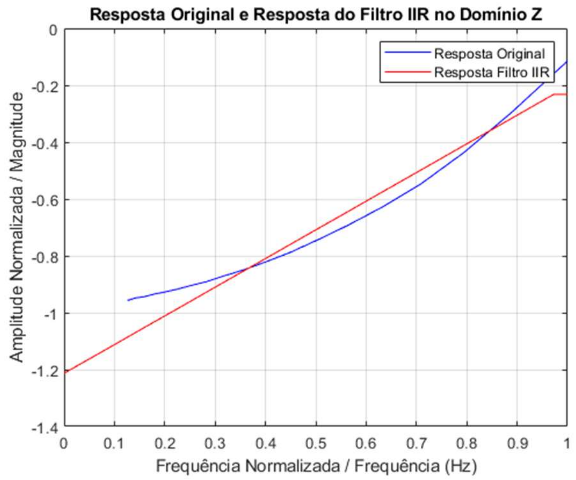
VI. IIR Filter response to the Impulse Function

As a last try, the filter was designed considering the inverse proposal: it is represented as a system efficiency insted of losses, in order to find a transfer function $G_p(z)$, in the *z* domain. A new low-pass IIR filter of the Butterworth type was proposed, with 2nd order and a cutoff frequency equal to 42.3% of the maximum one (normalized). The sampling frequency f_s was meant to be 35. The number of frequency samples was 19 in a first moment, because the vectors for the losse at 50mT and 25°C have this size.

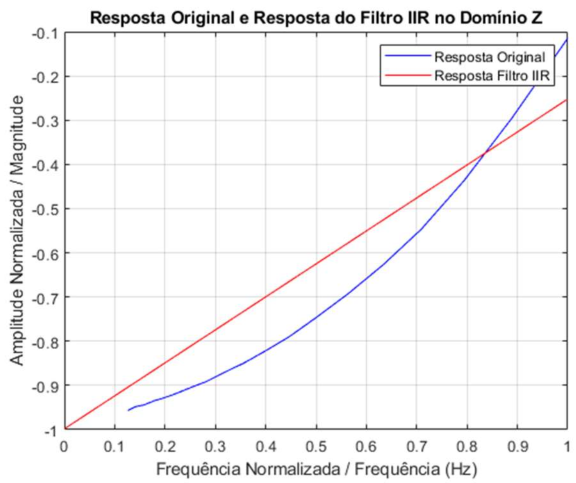
After this conditions were established, two distinct methods were used to convert the IIR filter designed filter from the original domain to a *z* domain filter, hence becoming possible to compare it with the original response of the data.

Two functions were compared to do this conversion to the *z* domain: the bilinear function and the “impinvar” function. The first one is a classical conversion, besides the Tustin function, which was used, but did not work well for this application. The second function applied in this conversion, the “impinvar”, is defined as an “impulse invariance method for analog to digital filter conversion”.

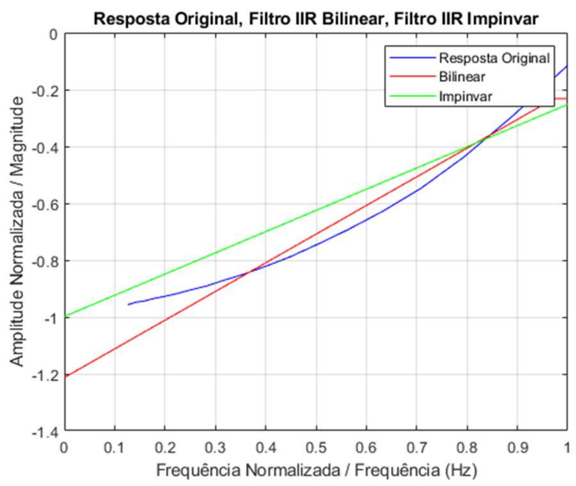
Finally, the Bilinear function presented, under the same IIR filter and determined conditions, better results than the Impinvar function, once the first shown lower erros in comparison to the original data, for the 50mT and 25°C curve of the N87 TDK core material. Fig. 5 shows the results, including the overlapping of original and filtered responses.



(a)



(b)



(c)

Fig. 5. Conversion of a new designed IIR filter from to the *z* domain and comparison with the original response of the N87 TDK core material at 50mT and 25°C. (a) Original curve and Bilinear function; (b) Original curve and Impinvar function; (c) Original curve, Bilinear function and Impinvar function.

Conclusion

This work worked on the development of an IIR filter function which was able to describe the behavior of volumetric core losses according to the excitation frequency in power converters, at different flux densisties and temperatures. This is a lack in the área, that both industry and academia has been given attention recently.

Sadly, FIR and IIR filters usual approaches were used in this work, but without good results, hence it is a valuable knowledge, however with no additions to reach the point.

The “clean” solution was the use of an unique IIR filter transfer function, working as an “all-pass” filter, in order to best represent all the curves. Using this function, it will be possible to apply computacional methods to obtain intermediary curves for other materials, which is the main objective of the proposal.

Finally, the *z* domain (sampled frequency) conversion of the IIR filter allowed for a filtered response closer to the original data, at least of the 50mT 25°C curve, mainly using the Bilinear function.

References

[1] C. P. Steinmetz, “On the Law of Hysteresis,” American Institute of Electrical Engineers, vol. 13. American Institute of Electrical Engineers, Schenectady, Nova York, EUA, pp. 655–657, 1892.

[2] G. K. Grassi, I. B. Barboza, M. A. Dalla Costa, and R. R. Duarte, “Power losses estimation in a GaN-based synchronous Buck LED driver,” 2022, doi: 10.1109/SEPOC54972.2022.9976449.

[3] Princeton, “2023 PELS-Google-Tesla-Princeton MagNet Challenge,” MagNet Challenge, 2023. https://www.princeton.edu/~minjie/magnet.html (accessed Jul. 06, 2023).

[4] TDK, “Ferrites and accessories - SIFERRIT material N87,” no. May. TDK, Tóquio, Japão, pp. 1–11, 2017.

Appendix A. Data prepare to be used

```
clear all
close all
clc

%% Declaração de Dados

%% 50mT

%25°C
freq_50mT_25 =[56240 63190 70850 79500 89160 99900
112140 125890 141250 158730 177930 199170 224180
251230 282480 316450 354610 396820 446420]
Pv_50mT_25 = [12961.9805 15604.4297 16830.1797
19549.1504 21775.6699 24738.2 28571.1602 32667.5195
38661.7695 44974.9414 53250.5703 62939.1797
76507.1797 91968.9062 111679.477 135915.984
169085.391 211298.875 265564];
f_norm_50mT_25 = freq_50mT_25 / max(freq_50mT_25);
Pv_norm_50mT_25 = Pv_50mT_25 / max(Pv_50mT_25);

%50°C
freq_50mT_50 = [56240 63180 70860 79500 89160 99900
112140 125890 141250 158730 177930 199160 224180
251230 282470 316450 354610 396820 446420];
Pv_50mT_50 = [8000.34 9720.24 10546.3701 12386.7695
14019.2197 15946.1 18903.2793 22124.6309 26990.3691
```

```
32040.0098 39063.5195 47316.8711 59145.1211
73419.1328 92091.4922 116392.062 149223.375
193058.875 249983.203];
f_norm_50mT_50 = freq_50mT_50 / max(freq_50mT_50);
Pv_norm_50mT_50 = Pv_50mT_50 / max(Pv_50mT_50);


%% 100mT

%25°C
freq_100mT_25 = [56240 63190 70850 79490 89160 99910
112140 125890 141250 158730 177930 199170 224180
251230 282470 316450 354600 396820];
Pv_100mT_25 = [69580.9531 79203.2 90271.1797
102713.242 118576.008 133625.281 157600.188
179447.469 210024.344 246916.797 290935.188
344088.312 414641.781 500484.5 593202.75 716305.438
876998.312 1090658.25];
f_norm_100mT_25 = freq_100mT_25 /
max(freq_100mT_25);
Pv_norm_100mT_25 = Pv_100mT_25 /
max(Pv_100mT_25);

%50°C
freq_100mT_50 = [56240 63190 70850 79490 89160 99910
112140 125890 141250 158730 177930 199170 224180
251230 282470 316450 354600 396820];
Pv_100mT_50 = [46270.7617 53212.0898 61596.5781
70606.6875 82424.75 93855.6 113079.703 131089.094
156683.016 187450.297 224904.891 271146.625
332914.656 410432.75 497767.5 614463.75 771741.688
981394.312];
f_norm_100mT_50 = freq_100mT_50 /
max(freq_100mT_50);
Pv_norm_100mT_50 = Pv_100mT_50 /
max(Pv_100mT_50);


%% 150mT

%25°C
freq_150mT_25 = [56240 63190 70850 79500 89160 99900
112140 125890 141250 158730 177930 199170 224180
251230];
Pv_150mT_25 = [203877.984 233051.844 266281.406
302003.281 348630.469 400471.875 470179.188 540618.75
632087 742155.812 859830.438 1017005.81 1221364.25
1471763.38];
f_norm_150mT_25 = freq_150mT_25 /
max(freq_150mT_25);
Pv_norm_150mT_25 = Pv_150mT_25 /
max(Pv_150mT_25);

%50°C
freq_150mT_50 = [56240 63180 70860 79500 89160 99900
112140 125890 141250 158730 177930 199160 224180
251230];
Pv_150mT_50 = [146583.547 167996.875 194712.375
222790.984 260048.75 301727.125 360366.781 419640.594
498170.344 594056.438 696136.5 836781 1019147.25
1248092.88];
f_norm_150mT_50 = freq_150mT_50 /
max(freq_150mT_50);
Pv_norm_150mT_50 = Pv_150mT_50 /
max(Pv_150mT_50);


%% 200mT

%25°C
freq_200mT_25 = [56240 63190 70850 79500 89160 99910
112140 125890 141250 158730 177930 199170];
Pv_200mT_25 = [345878.938 395131.344 449674.219
518657.125 599623.375 683833.375 802855.375
927695.062 1071741.62 1257193.88 1478312.5
1753108.5];
f_norm_200mT_25 = freq_200mT_25 /
max(freq_200mT_25);
Pv_norm_200mT_25 = Pv_200mT_25 /
max(Pv_200mT_25);

%50°C
freq_200mT_50 = [56240 63180 70860 79500 89160 99900
112140 125890 141250 158730 177930 199100];
Pv_200mT_50 = [261250.516 297294.688 345533.406
400924.375 467704.438 538197.312 641623.875
749640.562 873957.688 1041204.5 1235279.62
1485049.62];
f_norm_200mT_50 = freq_200mT_50 /
max(freq_200mT_50);
Pv_norm_200mT_50 = Pv_200mT_50 /
max(Pv_200mT_50);


%% 250mT

%25°C
freq_250mT_25 = [56240 63180 70850 79500 89170 99900
112140 125890 141250 158730];
Pv_250mT_25 = [572013.25 661204.375 751797.062
861220.312 995656.562 1142673 1325138 1531711.5
1798721.62 2123832.25];
f_norm_250mT_25 = freq_250mT_25 /
max(freq_250mT_25);
Pv_norm_250mT_25 = Pv_250mT_25 /
max(Pv_250mT_25);

%50°C
freq_250mT_50 = [56240 63180 70860 79500 89160 99900
112140 125890 141250 158730];
Pv_250mT_50 = [452297.062 523987.594 606320
698803.688 814303.375 941706.188 1103591.62
1288716.88 1526658.88 1821607.62];
f_norm_250mT_50 = freq_250mT_50 /
max(freq_250mT_50);
Pv_norm_250mT_50 = Pv_250mT_50 /
max(Pv_250mT_50);


%% 275mT

%25°C
```

```
freq_275mT_25 = [56240 63180 70850 79500 89160 99910
112140 125890 141250];
Pv_275mT_25 = [741135.812 859186.25 970938.688
1123011.12 1290328.75 1472468.25 1718660.12 2003653
2347360.25];
f_norm_275mT_25 = freq_275mT_25 /
max(freq_275mT_25);
Pv_norm_275mT_25 = Pv_275mT_25 /
max(Pv_275mT_25);

%50°C
freq_275mT_50 = [56240 63180 70860 79500 89160 99900
112140 125890 141250];
Pv_275mT_50 = [600210.875 697181.875 798465.375
931559.25 1078281.38 1239184.12 1463338.88 1722458.25
2033524.88];
f_norm_275mT_50 = freq_275mT_50 /
max(freq_275mT_50);
Pv_norm_275mT_50 = Pv_275mT_50 /
max(Pv_275mT_50);


%% Plotagem das Curvas de Diferentes Induções
Magnéticas (Bmax), em 25°C, para Diversas Frequências de
Excitação do Núcleo

% 50mT
%scatter(freq1,Pv_50mT); %plot(Freq,Pv_50mT);
%hold on;

% Formatação do Gráfico com todas as Bmax
  % Adicionar rótulos aos eixos x e y
  xlabel('Frequency (Hz)');
  ylabel('Volumetric Losses (kW/m^3)');

  % Adicionar título ao gráfico
  title('N87 Ferrite Material Volumetric Core Losses');

  % Legendas
  legend('50mT');

%% Projeto Filtro IIR 50mT - 50°C

ordem_50mT = 1; % Escolha a ordem do filtro (ajuste
conforme necessário)
fc_50mT = 0.99; % Escolha a frequência de corte do filtro
(ajuste conforme necessário)
[b, a] = butter(ordem_50mT, fc_50mT, 'low');

% Resposta do filtro IIR projetado
freq_filt_50mT = linspace(0, 1, 1000); % Vetor de
frequência para a resposta do filtro
H = freqz(b, a, freq_filt_50mT, fc_50mT); % Calcula a
resposta em frequência do filtro
figure;
%plot(freq_filt_50mT,H)

% Aplicação do filtro IIR às Funções Originais, de 50mT a
275mT, em 25°C e
% 50°C
Pv_filtered_50mT_25 = filter(b, a, Pv_50mT_25);
Pv_filtered_50mT_50 = filter(b, a, Pv_50mT_50);

Pv_filtered_100mT_25 = filter(b, a, Pv_100mT_25);
Pv_filtered_100mT_50 = filter(b, a, Pv_100mT_50);

Pv_filtered_150mT_25 = filter(b, a, Pv_150mT_25);
Pv_filtered_150mT_50 = filter(b, a, Pv_150mT_50);

Pv_filtered_200mT_25 = filter(b, a, Pv_200mT_25);
Pv_filtered_200mT_50 = filter(b, a, Pv_200mT_50);

Pv_filtered_250mT_25 = filter(b, a, Pv_250mT_25);
Pv_filtered_250mT_50 = filter(b, a, Pv_250mT_50);

Pv_filtered_275mT_25 = filter(b, a, Pv_275mT_25);
Pv_filtered_275mT_50 = filter(b, a, Pv_275mT_50);

%% Plot da resposta original e filtrada
figure;

%% Respostas Originais
scatter(freq_50mT_25,Pv_50mT_25);
hold on;

scatter(freq_50mT_50,Pv_50mT_50);
hold on;

scatter(freq_100mT_25,Pv_100mT_25);
hold on;

scatter(freq_100mT_50,Pv_100mT_50);
hold on;

scatter(freq_150mT_25,Pv_150mT_25);
hold on;

scatter(freq_150mT_50,Pv_150mT_50);
hold on;

scatter(freq_200mT_25,Pv_200mT_25);
hold on;

scatter(freq_200mT_50,Pv_200mT_50);
hold on;

scatter(freq_250mT_25,Pv_250mT_25);
hold on;

scatter(freq_250mT_50,Pv_250mT_50);
hold on;

scatter(freq_275mT_25,Pv_275mT_25);
hold on;

scatter(freq_275mT_50,Pv_275mT_50);
hold on;


%% Respostas Filtradas
plot(freq_50mT_25,Pv_filtered_50mT_25);
hold on;

plot(freq_50mT_50,Pv_filtered_50mT_50);
```

```
hold on;

plot(freq_100mT_25,Pv_filtered_100mT_25);
hold on;

plot(freq_100mT_50,Pv_filtered_100mT_50);
hold on;

plot(freq_150mT_25,Pv_filtered_150mT_25);
hold on;

plot(freq_150mT_50,Pv_filtered_150mT_50);
hold on;

plot(freq_200mT_25,Pv_filtered_200mT_25);
hold on;

plot(freq_200mT_50,Pv_filtered_200mT_50);
hold on;

plot(freq_250mT_25,Pv_filtered_250mT_25);
hold on;

plot(freq_250mT_50,Pv_filtered_250mT_50);
hold on;

plot(freq_275mT_25,Pv_filtered_275mT_25);
hold on;

plot(freq_275mT_50,Pv_filtered_275mT_50);
hold on;

title('Volumetric Losses of the N87 Ferrite Material');
xlabel('Frequency (Hz)');
ylabel('Volumetric Losses (kW/m^3)');

legend('50mT - 25°C','50mT - 50°C','100mT - 25°C','100mT - 50°C','150mT - 25°C','150mT - 50°C','200mT - 25°C','200mT - 50°C','250mT - 25°C','250mT - 50°C','275mT - 25°C','275mT - 50°C');

%100mT - Original 50°C','100mT - Filtrada 50°C','150mT - Original 25°C','150mT - Filtrada 25°C','150mT - Original 50°C','150mT - Filtrada 50°C','200mT - Original 25°C','200mT - Filtrada 25°C','200mT - Original 50°C','200mT - Filtrada 50°C','250mT - Original 25°C','250mT - Filtrada 25°C','250mT - Original 50°C','250mT - Filtrada 50°C','275mT - Original 25°C','275mT - Filtrada 25°C','275mT - Original 50°C','275mT - Filtrada 50°C'

grid on;
```

Appendix B. Conversion to *Z* Domain Using the Bilinear Function

```
clear all
close all
clc

% Vetores de frequência e resposta original
freq_50mT_25 = [56240 63190 70850 79500 89160 99900 112140 125890 141250 158730 177930 199170 224180 251230 282480 316450 354610 396820 446420];
Pv_50mT_25 = [12961.9805 15604.4297 16830.1797 19549.1504 21775.6699 24738.2 28571.1602 32667.5195 38661.7695 44974.9414 53250.5703 62939.1797 76507.1797 91968.9062 111679.477 135915.984 169085.391 211298.875 265564];

% Normalização dos vetores
Pv_50mT_25_norm = Pv_50mT_25 / max(Pv_50mT_25);
freq_50mT_25_norm = freq_50mT_25 / max(freq_50mT_25);

A = 300e3; % Potência inicial maior que a maior potência do vetor "Pv_50mT_25"
Gp = (A-Pv_50mT_25)/A; % Função resultante da conversão

% Projetar filtro IIR passa-baixas usando a função butter
ordem = 2; % Ordem do filtro
fc = 0.423; % Frequência de corte normalizada

[b, a] = butter(ordem, fc, 'low');

% Converter filtro para o domínio Z usando a função bilinear
fs = 35; % Frequência de amostragem
[bz, az] = bilinear(b, a, fs);

% Frequência de amostragem para o filtro no domínio Z
freq_samples = 19; % Número de amostras de frequência
w = linspace(0, pi, freq_samples);
w_hz = w / pi * fs / 2;

% Calcular resposta em frequência do filtro no domínio Z
[H, ~] = freqz(bz, az, w);

% Plotar resposta original e resposta do filtro no domínio Z
figure;

% Subplot com a resposta original
subplot(2, 1, 1);
plot(freq_50mT_25_norm, -Gp, 'b');
xlabel('Frequência Normalizada');
ylabel('Amplitude Normalizada');
title('Resposta Original');

% Subplot com a resposta do filtro no domínio Z
subplot(2, 1, 2);
```

```
plot(w_hz, -abs(H), 'r');
xlabel('Frequência (Hz)');
ylabel('Magnitude');
title('Resposta do Filtro IIR no Domínio
Z');
xlim([0, 1]);

% Plot das duas respostas sobrepostas em
outra figura
figure;
plot(freq_50mT_25_norm, -Gp, 'b');
hold on;
plot(w_hz, -abs(H), 'r');
xlabel('Frequência Normalizada /
Frequência (Hz)');
ylabel('Amplitude Normalizada /
Magnitude');
title('Resposta Original e Resposta do
Filtro IIR no Domínio Z');
legend('Resposta Original', 'Resposta
Filtro IIR');
xlim([0, 1]);

grid on;
```

APPENDIX C. CONVERSION TO *Z* DOMAIN USING THE IMPINVAR FUNCTION

```
clear all
close all
clc

% Vetores de frequência e resposta
original
freq_50mT_25 = [56240 63190 70850 79500
89160 99900 112140 125890 141250 158730
177930 199170 224180 251230 282480
316450 354610 396820 446420];
Pv_50mT_25 = [12961.9805 15604.4297
16830.1797 19549.1504 21775.6699 24738.2
28571.1602 32667.5195 38661.7695
44974.9414 53250.5703 62939.1797
76507.1797 91968.9062 111679.477
135915.984 169085.391 211298.875
265564];

% Normalização dos vetores
Pv_50mT_25_norm = Pv_50mT_25 /
max(Pv_50mT_25);
freq_50mT_25_norm = freq_50mT_25 /
max(freq_50mT_25);

A = 300e3; % Potência inicial maior que
a maior potência do vetor "Pv_50mT_25"
Gp = (A-Pv_50mT_25)/A; % Função
resultante da conversão

% Projetar filtro IIR passa-baixas
usando a função butter
ordem = 2; % Ordem do filtro
fc = 0.423; % Frequência de corte
normalizada

[b, a] = butter(ordem, fc, 'low');

% Converter filtro para o domínio Z
usando a função impinvar
fs = 45; % Frequência de amostragem
[bz, az] = impinvar(b, a, fs);

% Frequência de amostragem para o filtro
no domínio Z
freq_samples = 19; % Número de amostras
de frequência
w = linspace(0, pi, freq_samples);
w_hz = w / pi * fs / 2;

% Calcular resposta em frequência do
filtro no domínio Z
[H, ~] = freqz(bz, az, w);

% Plotar resposta original e resposta do
filtro no domínio Z
figure;

% Subplot com a resposta original
subplot(2, 1, 1);
plot(freq_50mT_25_norm, -Gp, 'b');
xlabel('Frequência Normalizada');
ylabel('Amplitude Normalizada');
title('Resposta Original');

% Subplot com a resposta do filtro no
domínio Z
subplot(2, 1, 2);
plot(w_hz, -abs(H), 'r');
xlabel('Frequência (Hz)');
ylabel('Magnitude');
title('Resposta do Filtro IIR no Domínio
Z');
xlim([0, 1]);

% Plot das duas respostas sobrepostas em
outra figura
figure;
plot(freq_50mT_25_norm, -Gp, 'b');
hold on;
plot(w_hz, -abs(H), 'r');
xlabel('Frequência Normalizada /
Frequência (Hz)');
ylabel('Amplitude Normalizada /
Magnitude');
title('Resposta Original e Resposta do
Filtro IIR no Domínio Z');
legend('Resposta Original', 'Resposta
Filtro IIR');
xlim([0, 1]);
grid on;
```