

Universidade Federal do Espírito Santo (**UFES**)  
Centro Universitário Norte do Espírito Santo (**CEUNES**)  
**Bacharelado em Ciência da Computação**

Trabalho I — Processamento Digital de Sinais  
**Sistema Reverberador**

João Victor do Rozário Recla

Novembro 2022

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Objetivos . . . . .	4
1.2	Sistema Proposto . . . . .	4
<b>2</b>	<b>Sinal de Áudio</b>	<b>4</b>
2.1	Aquisição do Sinal . . . . .	4
2.2	Representações do Sinal . . . . .	5
2.3	Frequência de Amostragem . . . . .	6
2.4	Classificação do Sinal . . . . .	7
<b>3</b>	<b>Sistema Reverberador</b>	<b>8</b>
3.1	Sistema . . . . .	8
3.1.1	1º Método de Schroeder . . . . .	8
3.1.2	2º Método de Schroeder . . . . .	9
3.1.3	Filtro Universal . . . . .	10
3.2	Respostas do Sistema . . . . .	11
3.2.1	Resposta a Função Impulso $\sigma[n]$ : . . . . .	11
3.2.2	Resposta a Função Degrau $u[n]$ : . . . . .	13
3.3	Propriedades do Sistema . . . . .	14
<b>4</b>	<b>Resultados</b>	<b>15</b>
<b>5</b>	<b>Implementação</b>	<b>16</b>
<b>6</b>	<b>Referências</b>	<b>16</b>

## Lista de Figuras

1	Aquisição do sinal e frequência de amostragem do áudio. (Arquivo <i>Main.py</i> )	4
2	Representação do sinal no tempo. . . . .	5
3	Espectro de amplitude do sinal. . . . .	5
4	Espectro de fase do sinal. . . . .	6
5	Informações do sinal de áudio. . . . .	6
6	Filtros All-pass em série. ( <i>1º Método de Schroeder</i> ) . . . . .	8
7	Estrutura do filtro All-pass. . . . .	9
8	Reverberador de Schroeder. ( <i>2º Método de Schroeder</i> ) . . . . .	9
9	Estrutura do filtro de delay. ( <i>Comb-filter IIR</i> ) . . . . .	10
10	Estrutura do filtro universal. . . . .	10
11	Parâmetros de configuração do filtro universal. . . . .	11
12	Resposta ao Impulso no tempo. . . . .	11
13	Espectro de amplitude em resposta ao Impulso. . . . .	12
14	Espectro de fase em resposta ao Impulso. . . . .	12
15	Resposta ao Degrau no tempo. . . . .	13
16	Espectro de amplitude em resposta ao Degrau. . . . .	13
17	Espectro de fase em resposta ao Degrau. . . . .	14
18	Resposta ao sinal no tempo. . . . .	15
19	Espectro de amplitude em resposta ao Sinal. . . . .	15
20	Espectro de fase em resposta ao Sinal. . . . .	16

## 1 Introdução

### 1.1 Objetivos

O objetivo deste trabalho é propor um sistema **LTI**, ou ainda **LIT (Linear e Invariante no Tempo)**, e aplicá-lo no processamento de um sinal de áudio, com canal **mono** e formato **.wav**, usando a linguagem de programação **Python**.

### 1.2 Sistema Proposto

“**Sistema Reverberador**” é o nome do sistema proposto para este trabalho. A sua modelagem se baseia na aplicação do efeito de reverberação [6] em um sinal de áudio. Após o processamento digital de um sinal, a resposta esperada para a saída do sistema é um sinal do próprio áudio condicionado as características do efeito de reverberação.

## 2 Sinal de Áudio

### 2.1 Aquisição do Sinal

Após importar um arquivo de áudio, com canal *mono* e formato *.wav*, para dentro da pasta do projeto, o sinal do áudio foi obtido através do comando **.read()** da biblioteca **soundfile**, que recebe o caminho do arquivo no projeto e retorna o sinal e a frequência de amostragem relacionado ao áudio (Conforme a figura 1).

```
18 import soundfile as Sound
19 #####
20
21
22 #
23 # Obtencao do sinal de audio.
24 #####
25 Sinal, Freq = Sound.read("Audios/Audio_01_82bpm.wav") # Audio 01.
26 # Sinal, Freq = Sound.read("Audios/Audio_02_88bpm.wav") # Audio 02 (Processamento mais lento).
27 # Sinal, Freq = Sound.read("Audios/Audio_03_130bpm.wav") # Audio 03 (Processamento mais lento).
28
```

Figura 1: Aquisição do sinal e frequência de amostragem do áudio.  
(Arquivo *Main.py*)

## 2.2 Representações do Sinal

O sinal do áudio adquirido pode ser representado tanto em função do tempo (Figura 2) como em função da frequência, sendo a frequência apresentada como o espectro de amplitude do sinal (Figura 3) e como o espectro de fase do sinal (Figura 4).

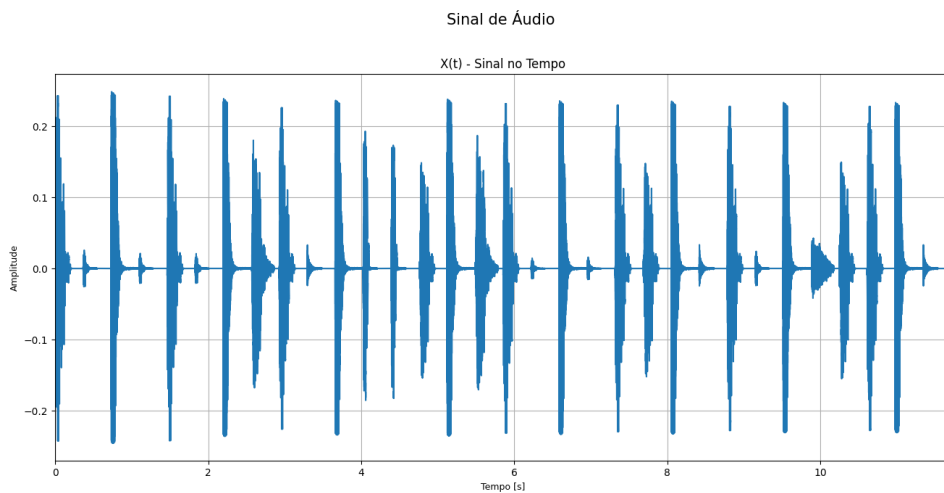


Figura 2: Representação do sinal no tempo.

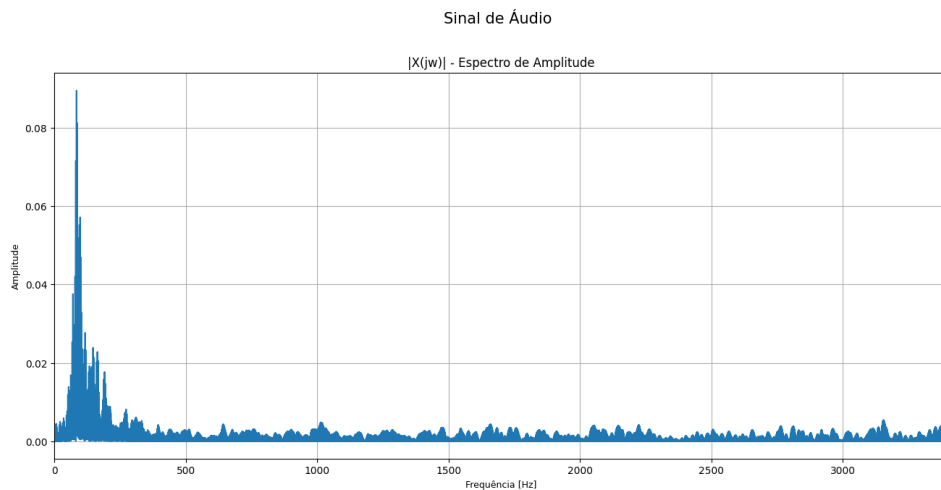


Figura 3: Espectro de amplitude do sinal.

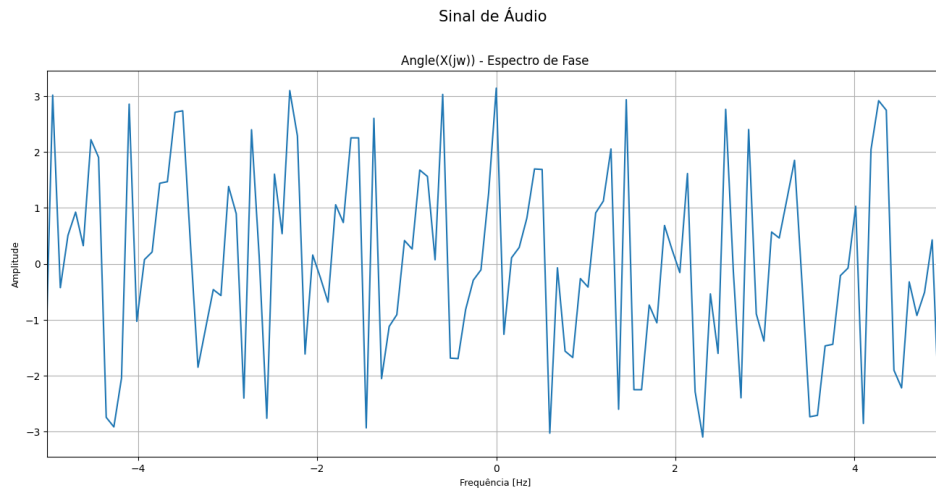


Figura 4: Espectro de fase do sinal.

### 2.3 Frequência de Amostragem

Para representar digitalmente o sinal do áudio no tempo é necessário obter a sua frequência de amostragem, que já é retornado pelo comando `.read()` (Conforme a figura 1). Após obter essa informação podemos usá-la para: (1) calcular a taxa, ou período, de amostragem do sinal (Dado por  $T_s = 1 / \text{Freq}$ ), (2) extrair o tempo de duração do áudio, em segundos, ( $\text{Duração} = \text{Amostras} * T_s$ ) e, assim, (3) poder representar o sinal em tempo discreto.

```
/Trabalho I - Sistema LTI/Main.py"

Canal de áudio:           Mono
Quantidade de amostras:   516293
Frequencia de amostragem: 44100
Taxa de amostragem:       2.2675736961451248e-05
Tempo de áudio (s):       11.71
```

Figura 5: Informações do sinal de áudio.

## 2.4 Classificação do Sinal

Observando as representações do sinal do áudio no tempo (Figura 2) e na frequência (Figuras 3 e 4), podemos analisar o sinal e classificá-lo como sendo:

- **Contínuo** — O Sinal é analógico;
- **Causal** — O sinal é definido a partir do instante de tempo  $t = 0s$ ;
- **Real** — O sinal apresenta valores reais em qualquer instante  $t$ ;
- **Aleatório** — O sinal apresenta valores aleatórios em qualquer instante  $t$ ;
- **Aperiódico** — O sinal não satisfaz a equação  $x(t) = x(t + Ts)$ .

### 3 Sistema Reverberador

A implementação do “**Sistema Reverberador**” é baseada em dois algoritmos apresentados por *Schroeder* [4] em seu artigo: “**Natural Sounding Artificial Reverberation**” [7]. Nesse artigo o objetivo geral descrito pelo autor é a simulação de um efeito artificial de reverberação, por meios digitais, que seja indistinguível do efeito natural de reverberação em um ambiente fechado.

Além da aplicação desses algoritmos, o sistema implementado tem como base outro método que permite a configuração do efeito artificial a partir do ajuste de alguns parâmetros, que podem ser modificados de tal modo que é possível representar outros efeitos apenas com algumas mudanças.

#### 3.1 Sistema

##### 3.1.1 1º Método de Schroeder

O primeiro algoritmo abordado no artigo — e o que será utilizado como base nesse relatório — consiste na passagem de um sinal de áudio por um conjunto de filtros **All-pass** em série (Figuras 6 e 7), sendo de 4-5 a quantidade de filtros necessários para se obter um sinal reverberado com boa qualidade.

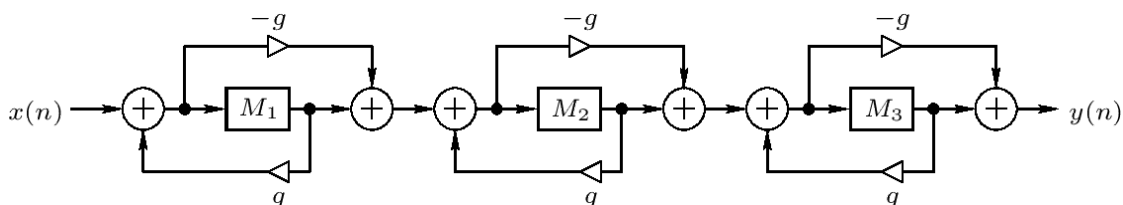


Figura 6: Filtros All-pass em série.  
(1º Método de Schroeder)



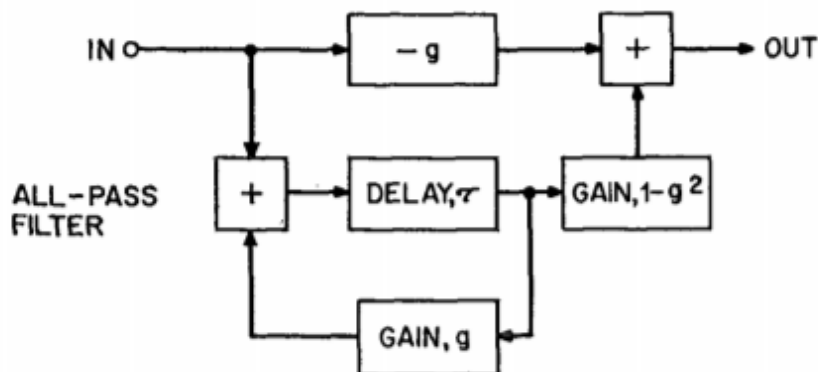


Figura 7: Estrutura do filtro All-pass.

### 3.1.2 2º Método de Schroeder

O segundo algoritmo abordado no artigo é um sistema que consiste na passagem de um sinal de áudio por um conjunto de 4 filtros “**Comb**” em paralelo, seguido da passagem do sinal resultante por um conjunto de 2 filtros **All-pass** em série, obtendo-se assim um sinal reverberado (Figura 8). Para esse método o filtro “**Comb**” utilizado por *Schroeder* [4] é um filtro do tipo **IIR** que aplica um efeito de *delay* no sinal de entrada (Figura 9).

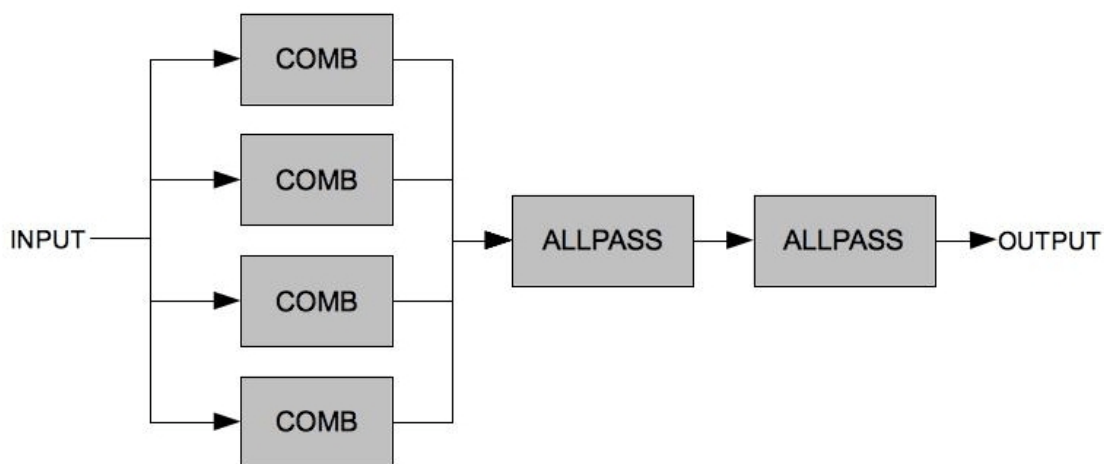


Figura 8: Reverberador de Schroeder.  
 (2º Método de Schroeder)

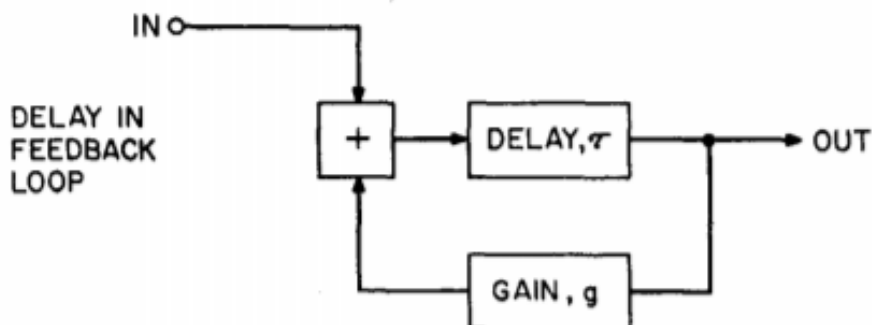


Figura 9: Estrutura do filtro de delay.  
 (Comb-filter IIR)

### 3.1.3 Filtro Universal

Tomando como base o artigo “**Implementação de efeito de áudio: atraso e reverb-  
 ração.**” [5] e o filtro universal (Figura 10), construído a partir da junção de um filtro **IIR**  
 com um filtro **FIR**, é possível reproduzir tanto os métodos de Schroeder, quanto criar ou-  
 tros efeitos apenas configurando alguns parâmetros, conforme a figura 11, e combinando  
 os filtros com a equação (1).

$$y[n] = \mathbf{BL} * x[n] + \mathbf{FF} * x[n - M] + \mathbf{BL} * \mathbf{FB} * y[n - M] \quad (1)$$

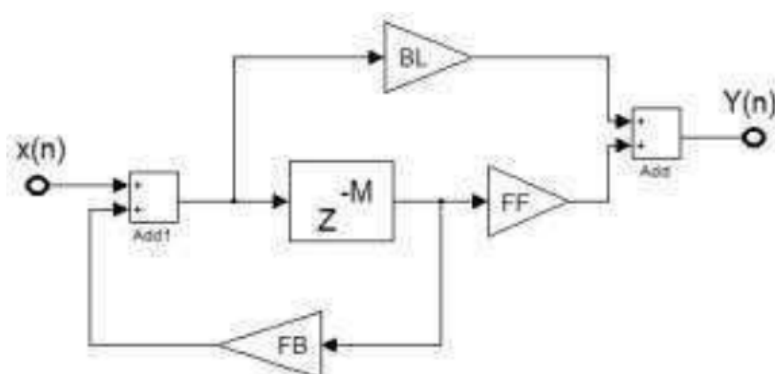


Figura 10: Estrutura do filtro universal.

	<b>BL</b>	<b>FB</b>	<b>FF</b>
<b>Filtro FIR</b>	1	0	Gd
<b>Filtro IIR</b>	1	Gd	0
<b>Filtro passa-tudo</b>	Gd	-Gd	1
<b>Delay puro</b>	0	0	1

Figura 11: Parâmetros de configuração do filtro universal.

## 3.2 Respostas do Sistema

Para o primeiro método de Schroeder (Figura 6), utilizado para montar esse relatório, obtemos as seguintes respostas do sistema às funções “Impulso” e “Degrau”:

### 3.2.1 Resposta a Função Impulso $\sigma[n]$ :

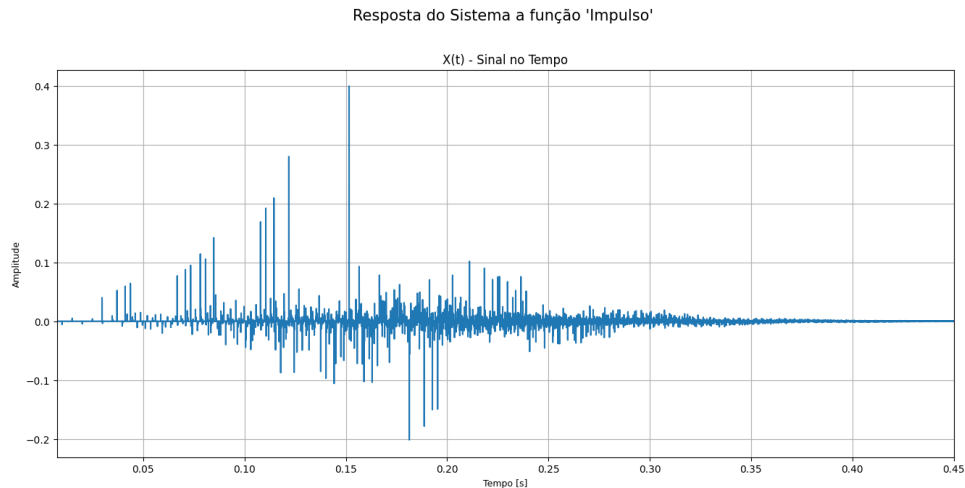


Figura 12: Resposta ao Impulso no tempo.

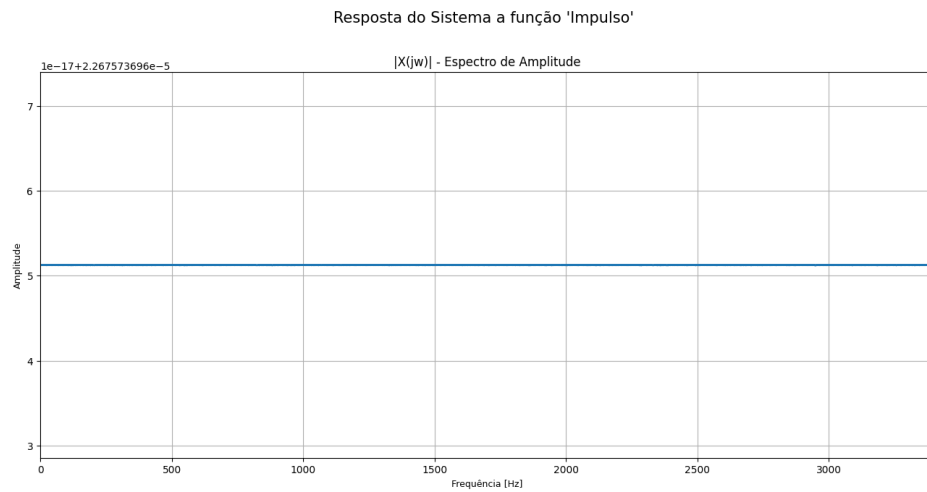


Figura 13: Espectro de amplitude em resposta ao Impulso.

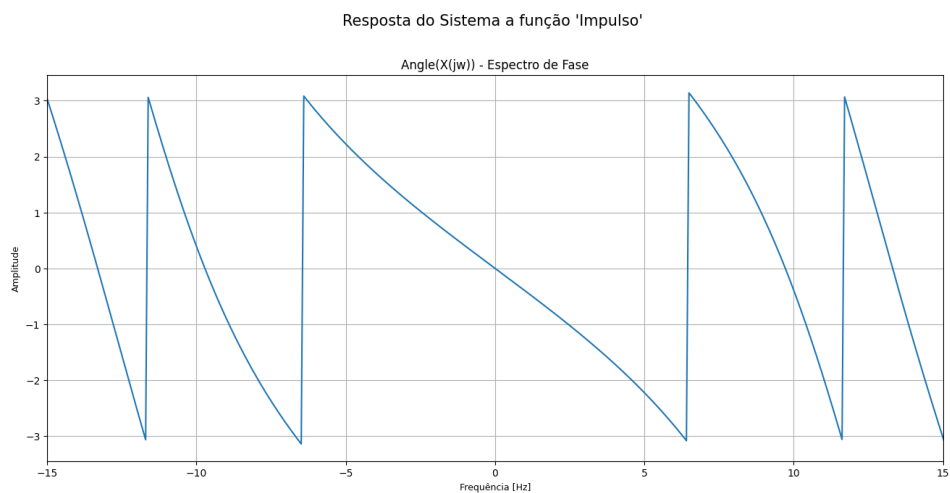


Figura 14: Espectro de fase em resposta ao Impulso.

### 3.2.2 Resposta a Função Degrau $u[n]$ :



Figura 15: Resposta ao Degrau no tempo.

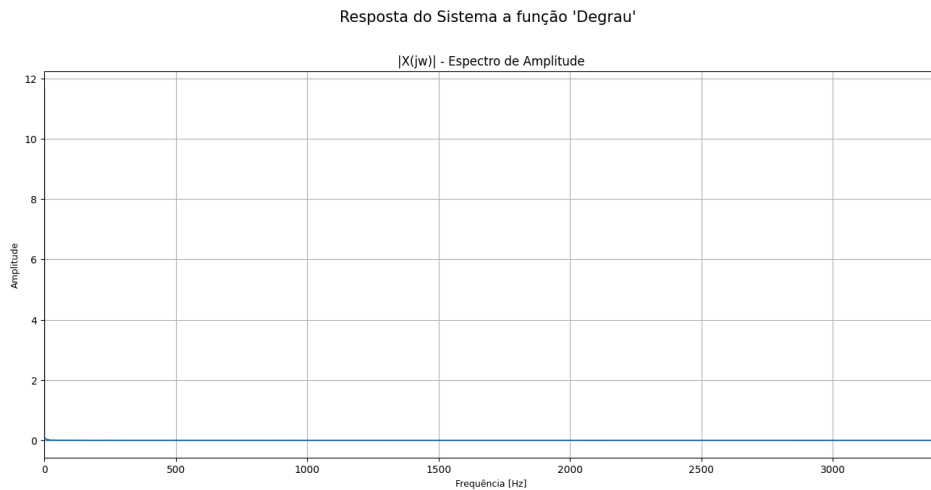


Figura 16: Espectro de amplitude em resposta ao Degrau.

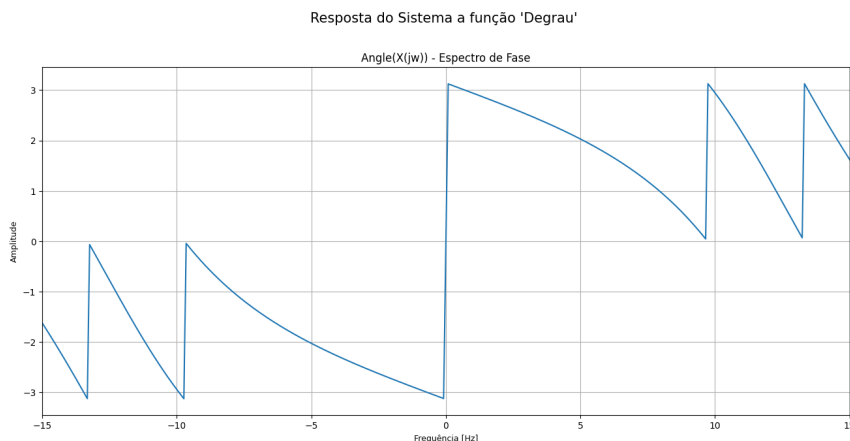


Figura 17: Espectro de fase em resposta ao Degrau.

### 3.3 Propriedades do Sistema

O “**Sistema Reverberador**”, com base no 1º Método de Schroeder (Figura 6) e na configuração do filtro universal pela equação (1), pode ser classificado conforme as seguintes propriedades:

- **Com memória:**

O sistema depende de valores do sinal de entrada no *passado*, ou seja, de pontos deslocados no tempo com base em um delay.

- **Estável:**

A saída do sistema não diverge, enquanto o sinal de entrada não for divergente.

- **Causal:**

A saída do sistema depende somente de valores do sinal de entrada no *presente* ou no *passado*.

- **Linear:**

O sistema satisfaz os princípios da superposição de *aditividade* e *mudança de escala*.

- **Invariante:**

A saída do sistema apresenta o mesmo deslocamento no tempo aplicado no sinal de entrada.

## 4 Resultados

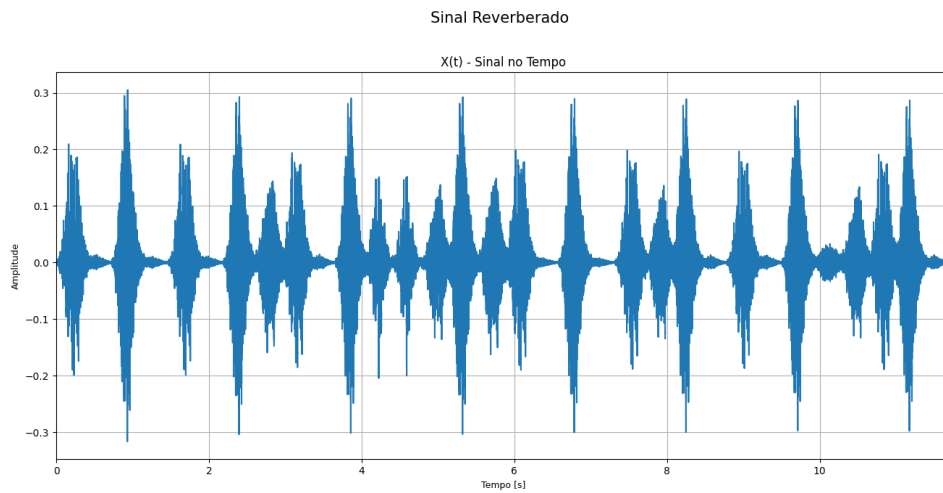


Figura 18: Resposta ao sinal no tempo.

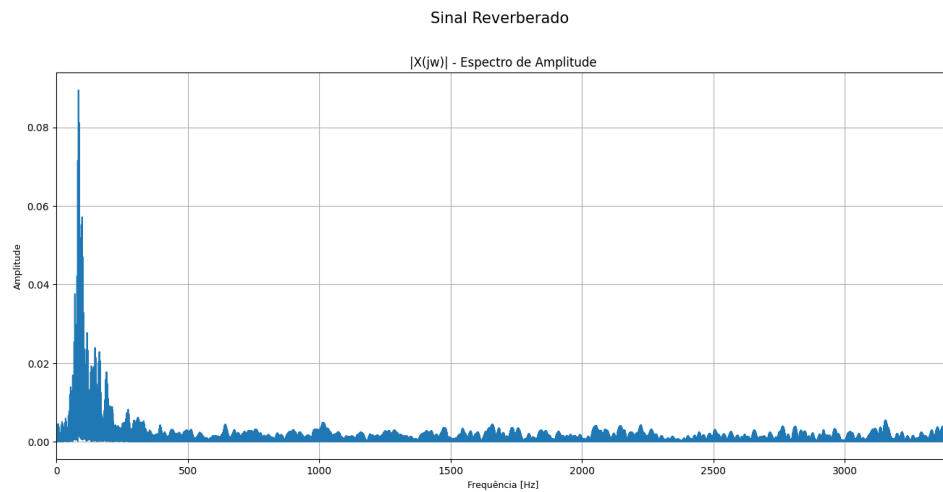


Figura 19: Espectro de amplitude em resposta ao Sinal.

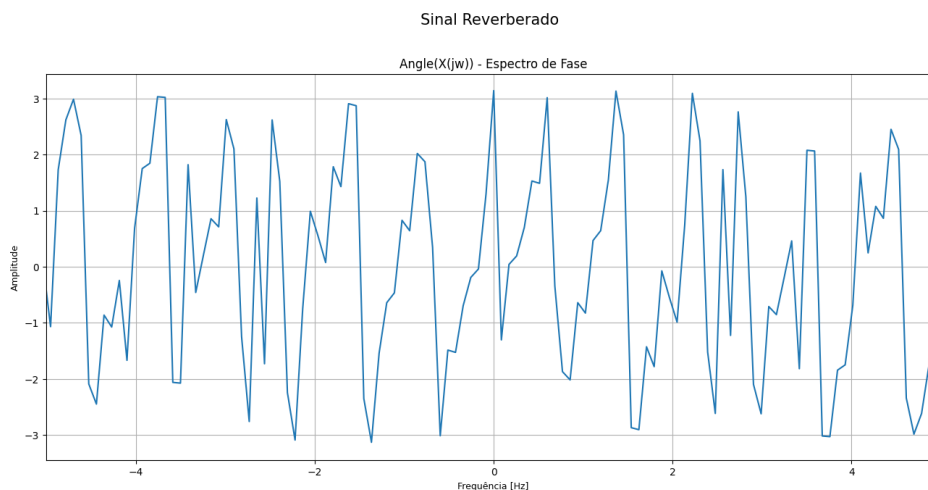


Figura 20: Espectro de fase em resposta ao Sinal.

## 5 Implementação

Os arquivos de áudio, códigos e gráficos que compõem o relatório e a implementação do sistema podem ser encontrados neste endereço eletrônico: [GitHub-Implementação](#)

## 6 Referências

- [1] Sean Costello. *Schroeder Reverbs: The forgotten algorithm*. Abr. de 2022. URL: <https://valhalladsp.com/2009/05/30/schroeder-reverbs-the-forgotten-algorithm/>.
- [2] Rishikesh Daoo. *Coding a basic reverb algorithm – Part 2: An introduction to audio programming*. Mar. de 2019. URL: <https://medium.com/the-seekers-project/coding-a-basic-reverb-algorithm-part-2-an-introduction-to-audio-programming-4db79dd4e325>.
- [3] Irina Dornean, Marina Topa e Botond Sandor Kirei. «Digital Implementation of Artificial Reverberation Algorithms». Em: *Acta Technica Napocensis* 49.4 (2008).
- [4] Manfred R. Schroeder. Jul. de 2021. URL: [https://en.wikipedia.org/wiki/Manfred\\_R.\\_Schroeder](https://en.wikipedia.org/wiki/Manfred_R._Schroeder).
- [5] Pedro Merencio Primo PASSOS et al. «Implementação de efeito de áudio: atraso e reverberação.» Em: (2015).
- [6] *Reverberação*. Nov. de 2022. URL: <https://pt.wikipedia.org/wiki/Reverbera%C3%A7%C3%A3o>.
- [7] Manfred R. Schroeder. «Natural Sounding Artificial Reverberation». Em: *Journal of the Audio Engineering Society* 10.3 (jul. de 1962), pp. 219–223. URL: <http://www.aes.org/e-lib/browse.cfm?elib=849>.