

Universidade Federal do Rio Grande do Norte

Departamento de Engenharia de Computação e Automação

DCA3602 - Processamento Digital de Sinais

Aluno: Lucas Tomaz de Moura

10 de janeiro de 2025

Implementação de algoritmo de FFT: Overlap-save ou Overlap-add

Simulação 1:

Inicialmente, implemente o algoritmo de convolução escolhido (overlap-save ou overlap-add) e valide a implementação com sinais aleatórios para a entrada $x[n]$ e para o filtro FIR $h[n]$. Utilize como comprimento do sinal de entrada (em amostras) um valor não múltiplo de L , em que $L = N - M + 1$, sendo N o número de pontos da FFT e M o número de amostras da resposta ao impulso $h[n]$. Escolha ainda um valor para M em torno de 150 amostras. Utilize um valor de comprimento para o sinal de entrada aproximadamente igual a $15,3 L$.

- O algoritmo implementado no passo anterior deve ser validado comparando o resultado de saída do processamento com o resultado de uma convolução no tempo entre o sinal de entrada $x[n]$ e o filtro $h[n]$ escolhidos.
- Os resultados obtidos com a validação do Passo 1 também devem ser incluídos no relatório do projeto.

Simulação 2:

Após a validação do método de convolução implementado anteriormente, o utilize em substituição à função filter do MATLAB (ou conv, ou similar em outra linguagem de programação) que tenha sido utilizada no projeto anterior de filtragem do sinal de voz da UII.

- Utilize dessa vez um sinal de voz com duração mínima de 10 segundos. Novamente, grave uma amostra de seu próprio sinal de voz.
- Utilize o mesmo filtro FIR projetado anteriormente, no projeto de filtragem de voz desenvolvido na UII.

Implementação dos parâmetros:

Todo o projeto da implementação usando FFT foi desenvolvido no software octave. Abaixo está o modelo de pseudocódigo descrevendo a lógica de execução do código que representa a simulação 1. Na figura 1, está elencado a saída dos sinais e uma comparação entre os dois métodos de implementação.

Resultados

A partir dos resultados obtidos, foi possível observar que o erro entre a implementação utilizando a Transformada Rápida de Fourier (FFT) com o método de overlap-save e a convolução direta no domínio do tempo ainda está alto. A diferença máxima entre os sinais gerados por ambos os métodos sugere que há discrepâncias significativas na forma como o sinal filtrado foi processado. Levando isso em conta, os resultados de implementação neste primeiro momento não são satisfatórios, cabendo uma melhor análise detalhada de cada elemento do programa e suas saídas.

Algoritmo 1: Filtro de Ruído com Overlap-Save

```
1:  $N \leftarrow 1024$                                 ▷ Número de pontos para a FFT
2:  $M \leftarrow 150$                                 ▷ Tamanho do filtro FIR
3:  $L \leftarrow N - M + 1$                             ▷ Tamanho do bloco útil
4:  $x\_length \leftarrow \text{arredondar}(15.3 \times L)$ 
5:  $x \leftarrow \text{gerarSinalAleatorio}(1, x\_length)$     ▷ Sinal de entrada com comprimento correto
6: Se  $\text{mod}(\text{length}(x), L) = 0$  Então
7:    $x \leftarrow \text{concatenar}(x, \text{gerarSinalAleatorio}(1, 1))$     ▷ Adicionar uma amostra extra
8: end Se
9:  $h \leftarrow \text{gerarSinalAleatorio}(1, M)$             ▷ Filtro FIR aleatório
10:  $x \leftarrow \text{concatenar}(\text{zeros}(1, M - 1), x)$     ▷ Preenchimento inicial com M-1 zeros
11:  $\text{num\_blocks} \leftarrow \text{teto}((\text{length}(x) - (M - 1))/L)$     ▷ Número de blocos
12:  $y \leftarrow \text{zeros}(1, \text{length}(x) + M - 1)$     ▷ Inicialização do vetor de saída
13:  $H \leftarrow \text{fft}(h, N)$                             ▷ FFT do filtro
14: Para  $i \leftarrow 1$  to  $\text{num\_blocks}$  Faça
15:    $\text{start\_idx} \leftarrow (i - 1) \times L + 1$         ▷ Índice de início do bloco
16:    $\text{end\_idx} \leftarrow \text{mínimo}(\text{start\_idx} + N - 1, \text{length}(x))$     ▷ Índice de fim do bloco
17:    $x\_block \leftarrow x[\text{start\_idx} : \text{end\_idx}]$         ▷ Extrair bloco
18:    $x\_block \leftarrow \text{concatenar}(x\_block, \text{zeros}(1, N - \text{length}(x\_block)))$     ▷ Zero-padding
19:    $X\_block \leftarrow \text{fft}(x\_block, N)$                 ▷ FFT do bloco
20:    $Y\_block \leftarrow X\_block \times H$                 ▷ Convolução no domínio da frequência
21:    $y\_block \leftarrow \text{ifft}(Y\_block, N)$             ▷ IFFT para saída no domínio do tempo
22:    $\text{useful\_part} \leftarrow y\_block(M : M + L - 1)$     ▷ Parte útil do bloco
23:    $\text{start\_y} \leftarrow (i - 1) \times L + 1$         ▷ Índice de início no vetor de saída
24:    $\text{end\_y} \leftarrow \text{start\_y} + \text{length}(\text{useful\_part}) - 1$     ▷ Índice de fim no vetor de saída
25:    $y[\text{start\_y} : \text{end\_y}] \leftarrow \text{useful\_part}$     ▷ Armazenar no vetor de saída
26: end Para
27:  $y \leftarrow y[M : \text{end}]$                             ▷ Remover o preenchimento inicial
28:  $y\_direct \leftarrow \text{conv}(x(M : \text{end}), h)$         ▷ Convolução direta no tempo
29:  $y\_adjusted \leftarrow y[1 : \text{length}(y\_direct)]$     ▷ Ajuste de comprimento para comparação
30:  $\text{error} \leftarrow \text{norm}(y\_adjusted - y\_direct)$         ▷ Erro entre os métodos
31: Exibir("Erro entre a implementação de FFT e a convolução direta: " +  $\text{error}$ )
32:  $\text{max\_diff} \leftarrow \text{máximo}(\text{abs}(y\_adjusted - y\_direct))$     ▷ Diferença máxima
33: Exibir("Diferença máxima: " +  $\text{max\_diff}$ )
34: CriarFigura()
35: Subplot(2, 1, 1)
36: Plot( $y\_direct$ )
37: Título("Resultado da convolução direta")
38: Subplot(2, 1, 2)
39: Plot( $y\_adjusted$ )
40: Título("Resultado da convolução usando overlap-save")
```

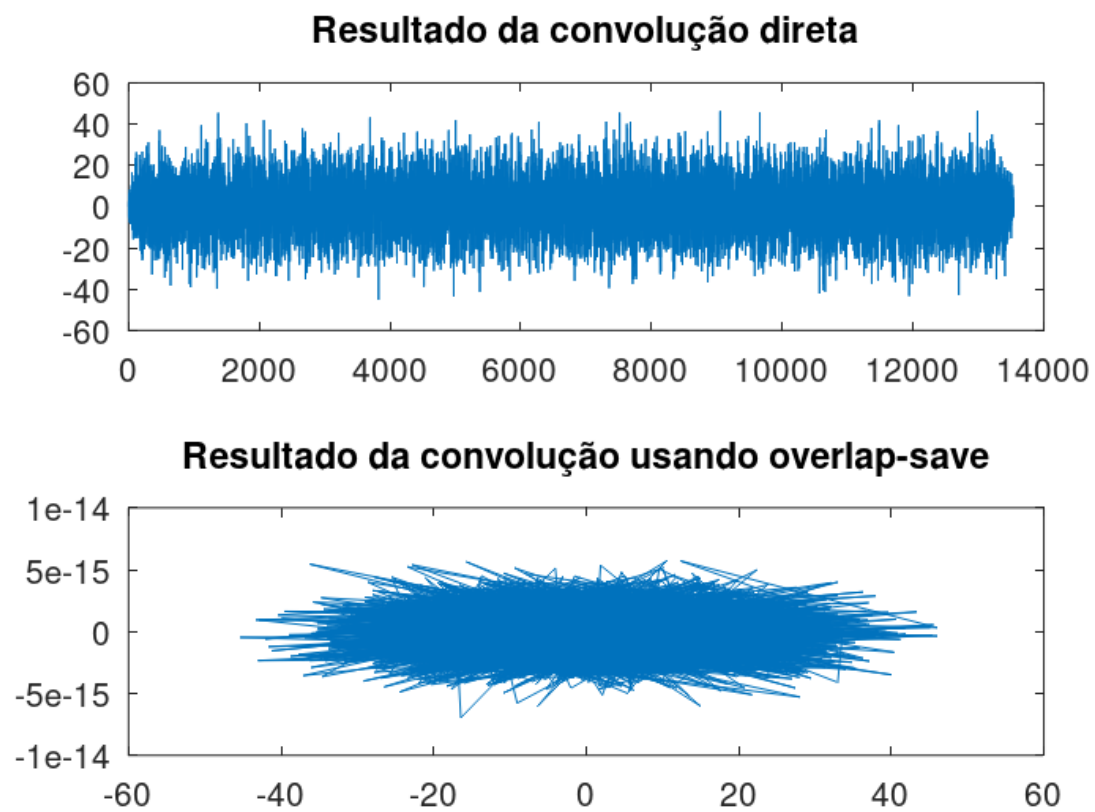


Figura 1: . Representação gráfica dos resultados da aplicação da convolução usando fft e overlap-save, respectivamente.