

Capítulo 6: Implementação de Filtros Discretos no Tempo

1. Introdução aos Filtros Digitais

Os **filtros digitais** são fundamentais no processamento de sinais, permitindo a modificação de espectros de frequência para diversas aplicações, como remoção de ruídos, compressão de dados e melhoria da qualidade de áudio e imagem.

Os filtros podem ser classificados em:  **Filtros FIR (Finite Impulse Response)** – Resposta finita ao impulso.  **Filtros IIR (Infinite Impulse Response)** – Resposta infinita ao impulso.

A implementação de filtros digitais envolve **equações de diferenças**, que descrevem a relação entre entrada e saída do sistema.

2. Estruturas de Implementação

A implementação de filtros digitais pode ser feita por diferentes estruturas matemáticas e computacionais. As principais abordagens incluem:

2.1. Forma Direta

A **forma direta** é a implementação mais intuitiva, baseada na equação de diferenças do filtro. Existem duas versões:

- **Forma Direta I:** Utiliza coeficientes do numerador e denominador diretamente.
- **Forma Direta II:** Mais eficiente em termos de memória, pois reduz o número de operações.

2.2. Forma Cascata

Na **forma cascata**, o filtro é decomposto em seções menores de segunda ordem chamadas **biquads**. Essa abordagem melhora a estabilidade numérica e reduz erros de arredondamento.

2.3. Forma Paralela

A **forma paralela** expande a função de transferência do filtro em somas de componentes menores, facilitando a implementação em hardware e software.

3. Implementação Computacional

A implementação de filtros digitais pode ser feita em **Python** e **MATLAB**. Aqui está um exemplo de um filtro FIR usando **SciPy** em Python:

```
python
```

```
import numpy as np
import scipy.signal as signal
import matplotlib.pyplot as plt





# Definição dos coeficientes do filtro FIR
num_taps = 51
cutoff_freq = 0.3 # Frequência de corte normalizada
fir_coeffs = signal.firwin(num_taps, cutoff_freq)

# Resposta em frequência do filtro
w, h = signal.freqz(fir_coeffs)

plt.plot(w / np.pi, 20 * np.log10(abs(h)))
plt.title("Resposta em Frequência do Filtro FIR")
plt.xlabel("Frequência Normalizada")
plt.ylabel("Magnitude (dB)")
plt.grid()
plt.show()
```

Esse código cria um **filtro FIR** e exibe sua resposta em frequência.

4. Aplicações dos Filtros Digitais

Os filtros digitais são amplamente utilizados em:  **Processamento de áudio** – Equalização e remoção de ruídos.  **Processamento de imagens** – Suavização e detecção de bordas.  **Comunicações digitais** – Modulação e demodulação de sinais.  **Sistemas biomédicos** – Filtragem de sinais ECG e EEG.