Análise na frequência

Osmar Tormena Junior, Prof. Me.*

1 Introdução

A análise do espectro de sinais digitais pode ser realizada através da DFT, comumente implementada através da FFT. Sistemas FIR podem ser implementados no domínio espectral, trazendo grandes vantagens computacionais. Sistemas IIR são passíveis de implementação temporal, apenas, mas seus resultados podem ser analisados no domínio da frequência.

Nesta prática, os discentes deverão utilizar algum sinal de seu interesse, como áudio ou biomédicos, como objeto de estudo das análises em frequência.

2 Filtros FIR

Filtros FIR são únicos em suas aplicações, pois se sua resposta impulsiva $h(n)$ possuir simetria, sua resposta em frequência possuirá fase linear, o que é extremamente desejável.

Filtros FIR costumam ter ordem elevada, superando em muito o custo computacional de um filtro IIR, para os mesmos parâmetros. Parte deste problema pode ser mitigado por sua implementação através do algoritmo da *convolução rápida*.

Filtros FIR de fase linear, por janelamento, podem ser implementados através da função `fir1`. Um projeto *equiripple*, otimizado, através do algoritmo de Parks-McClellan, pode ser projetado através das funções `firpmord` e `firpm`.

Exercício 1: Obtenha um filtro FIR de fase linear, e aplique-o, através da função `filter` em seu sinal de interesse. Verifique se o filtro atende às suas especificações.

2.1 Convolução rápida

A soma de convolução, para um sistema FIR causal de ordem M, é dada por

$$y(n) = \sum_{m=0}^{M} h(m)x(n-m),$$

considerando que o sinal $x(n)$ tem comprimento N, o sinal $y(n)$ terá comprimento $M + N - 1$.

Como comumente $N > M$, é possível completar $h(n)$ com zeros, de forma que ambos $x(n)$ e $h(n)$ agora tenham comprimento N, sendo possível escrever a convolução circular

$$\tilde{y}(n) = \sum_{m=0}^{N-1} h(m)x((n-m))_N,$$

onde $\tilde{y}(n)$ é uma sequência com N amostras, claramente diferente de $y(n)$.

Porém, completando $x(n)$ e $y(n)$ com zeros, de forma que agora ambas têm comprimento $L = M + N - 1$, podemos escrever

$$y'(n) = \sum_{m=0}^{L-1} h(m)x((n-m))_L,$$

onde é possível verificar que $y(n) = y'(n)$. Infelizmente, este caminho incorre em aumentar o tamanho da DFT, completando as sequências com muitos zeros, o que causa perda da eficiência computacional.

Um bom compromisso é considerar $L = \text{máx}(M, N)$. Como $N > M$, na maioria dos

*tormena@utfpr.edu.br

casos $y'(n)$ será igual à $y(n)$, exceto as $M-1$ primeiras amostras, que sofrerão com *aliasing* temporal.

Segmentando um sinal $x(n)$ em blocos, de comprimento N (sendo N uma potência inteira de 2) , sobrepostos em suas $M-1$ primeiras amostras, obtendo $X(k)$ pela N-FFT, aplicando

$$Y'(k) = H(k)X(k),$$

e finalmente aplicando a N-IFFT, obtendo $y'(n)$, descartando as primeiras $M-1$ amostras, pode-se concatenar esses blocos, obtendo $y(n)$. Este algoritmo é chamado de *overlap and save.*

Exercício 2: Implemente o filtro obtido no exercício anterior através da convolução rápida.

3 Filtros IIR

Filtros IIR nunca possuem fase linear, porém diferentes filtros causam distorções em diferentes níveis, o que pode ter diferentes níveis de tolerância, em diferentes aplicações. Abandonando a causalidade (i.e. processamento em tempo real), é possível implementar filtros IIR de fase zero, contornando o problema da distorção.

Para um dado conjunto de parâmetros, filtros IIR costumam atender com menor custo computacional (i.e. menor ordem N). Porém fatores como instabilidades numéricas da forma direta podem diminuir esta vantagem, forçando o uso de soluções cascata ou no espaço de estados, comumente mais custosas.

É comum projetar filtros IIR como discretizações das aproximações clássicas de Butterworth, Chebyshev[1] e Cauer[2].

As funções `buttord`, `cheb1ord`, `cheb2ord` e `ellipord` são úteis para obter as especificações de um filtro, a partir dos parâmetros da resposta em frequência. As funções `butter`, `cheby1`, `cheby2` e `ellip` podem ser usadas no projeto dos filtros.

Exercício 3: Implemente a filtragem, nas aproximações de Butterworth, Chebyshev e Cauer, sobre o sinal, analisando o custo computacional e a distorção introduzida pelo filtro.

3.1 Filtros digitais de fase zero

O pós-processamento, com um filtro IIR (ou FIR), pode ser feito filtrando o sinal de entrada progressivamente ($x(n)$) e regressivamente ($x(N-1-n)$), combinando as saídas para produzir uma fase zero (sem atraso e sem distorção). A função `filtfilt` pode ser usada para implementar este tipo de filtragem.

Um dos efeitos deste tipo de filtragem é que o filtro é, essencialmente, aplicado duas vezes ao sinal, sendo possível então diminuir à metade a ordem projetada.

Exercício 4: Implemente a filtragem de fase zero, e compare com quaisquer distorções observadas no exercício anterior.

[1] Tipos I e II (ou invertido).
[2] Mais conhecido como filtro elíptico.