EA614 ANÁLISE DE SINAIS

TESTE COMPUTACIONAL 4

Entrega: 29/11/2017

Verifica-se que a Série de Fourier de Tempo Discreto (SFTD) transforma sinais discretos no tempo em espectros também periódicos e discretos na frequência, como exemplificado na Figura 1.

Se possuímos um sinal de tempo discreto não-periódico $x[n]$, podemos aproximar o cálculo do seu espectro através do uso da SFTD. Para isto, é necessário replicar periodicamente o sinal $x[n]$, obtendo $x_P[n]$. Como o cálculo da SFTD ocorre apenas em um período do sinal $x_P[n]$, define-se a Transformada Discreta de Fourier (TDF) de $x[n]$ como a SFTD de $x_P[n]$ no intervalo $0 \leq n < N$.

Programas como o Python e o MATLAB utilizam um algoritmo rápido para o cálculo da TDF, a *Fast Fourier Transform* (FFT). No Python, o espectro de um sinal é obtido com a função `numpy.fft.fft()` e a transformada inversa é calculada com a função `numpy.fft.ifft()`.

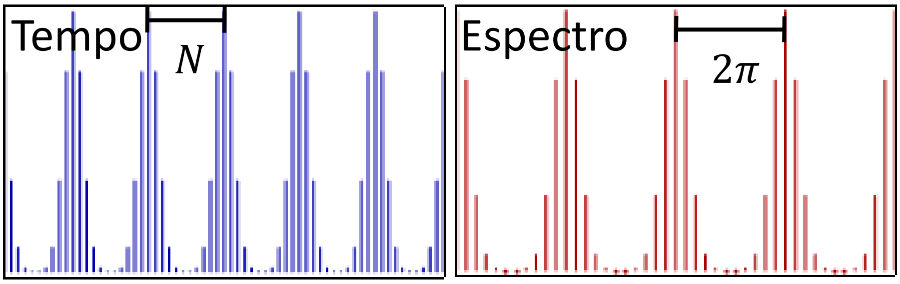

Figura 1: Esboço de um sinal discreto e periódico no tempo (esquerda) e seu espectro, que também é discreto e periódico (direita).

Além disto, sabemos que a convolução linear é uma operação que relaciona dois sinais $a[n]$ e $b[n]$ e resulta em um terceiro sinal $c[n] = a[n] * b[n]$ mais comprido que os dois sinais convoluídos. Se, no entanto, $a[n]$ e $b[n]$ são sinais periódicos, então sua convolução também será periódica com período igual ao mínimo múltiplo comum do período dos dois sinais convoluídos.

Agora, lembremos, como discutido acima, que a TDF tem como pressuposto implícito que a representação discreta de um sinal corresponde, na realidade, à descrição de um sinal periódico $x_P[n]$, cujo cada período é igual ao sinal limitado $x[n]$ sendo transformado. Sendo assim, calcular a convolução de $a[n]$ e $b[n]$ através da propriedade da multiplicação do espectro obtido pela TDF assume a convolução de sequências periódicas e resulta, portanto, também em uma sequência periódica. Logo, a convolução obtida desta maneira é dita cíclica ou circular, e representada por $c[n] = a[n] \circledast b[n]$.

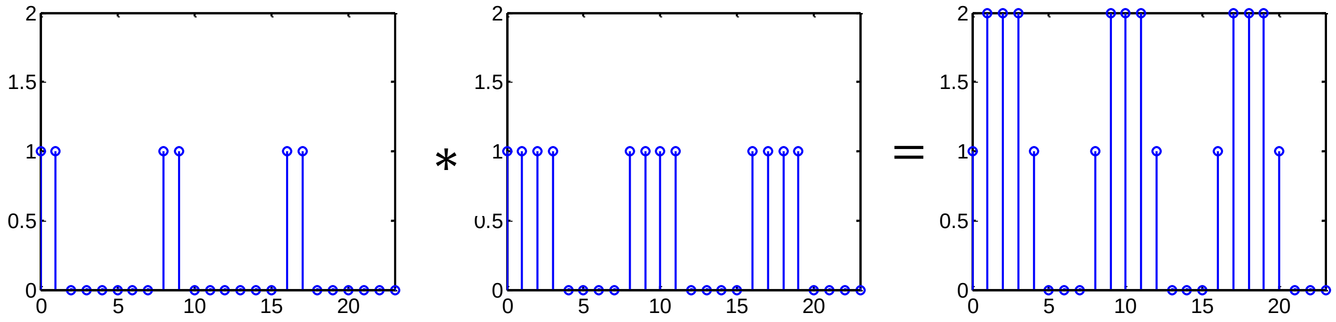

Figura 2: Exemplo de convolução circular. Assume-se que os dois sinais sendo convoluídos são periódicos e verifica-se que o sinal resultante também é periódico.

Exercício 3: Convolução

a) O Python possui a função `scipy.signal.sawtooth()` que gera uma onda dente-de-serra. Amostre o período fundamental da onda $x(t) = \text{sawtooth}(2\pi t)$ com 12 amostras e exiba o decurso temporal de $x[n]$. Usando a função `matplotlib.pyplot.stem()`.

b) Assumindo a mesma taxa de amostragem de a), obtenha 8 amostras do sinal $h(t) = \delta(t - 0{,}25)$, exibindo o decurso temporal de $h[n]$. Comente o que aconteceria se $h(t) = \delta(t - 0{,}3)$.

c) Calcule a convolução linear $y[n] = x[n] * h[n]$, usando para isto a função `y=numpy.convolve(x,h)`. Exiba o decurso temporal de $y[n]$, obtendo para isto os índices adequados. Explique, analiticamente, qual o comprimento do sinal resultante.

d) Exiba os gráficos dos espectros $X[k]$ e $H[k]$ obtidos com a função `numpy.fft.fft`.

e) Sabemos que a convolução dos sinais no domínio do tempo é equivalente à multiplicação do seus espectros. Calcule o espectro $Y[k]$ usando multiplicação elemento-a-elemento `Y=X*H`. Qual o resultado obtido? Porquê?

f) Obtenha agora 12 amostras do sinal $h(t) = \delta(t - 0{,}25)$, calcule $H[k]$ e calcule novamente o espectro $Y[k]$ usando a multiplicação elemento-a-elemento “`*`”.

g) Obtenha o sinal $y'[n]$ no tempo usando para isto a transformada inversa de Fourier, dada pela função `y=numpy.fft.ifft(Y)`. Compare este resultado com $y[n]$. Comente.

h) Concatene os sinais $x[n]$ e $h[n]$ com zeros de forma que estas duas sequências tenham o mesmo comprimento que o resultado da convolução linear obtido em c). Agora calcule $y''[n] = x[n] \circledast h[n]$, através da multiplicação elemento-a-elemento no espectro. Compare $y''[n]$ com $y'[n]$ e $y[n]$.

i) Comente qual a condição necessária para que a convolução linear e a convolução circular proporcionem resultados idênticos.