PROCESSAMENTO DIGITAL DE SINAIS

PRÁTICA:

**FILTRAGEM DIGITAL & MÉDIA MÓVEL**

**1. Objetivo**

Neste laboratório vamos investigar os sistemas discretos, mais particularmente, vamos concentrar a nossa atenção numa subclasse de sistemas conhecidos como **Sistemas Invariantes de Tempo Linear** (SLIT). Veremos que no caso de um SLIT podemos escrever uma expressão fechada para se calcular a saída do sistema para uma dada entrada. Usando os conceitos de **Resposta ao Impulso** e **Convolução**, vamos introduzir o conceito de filtragem digital.

**2. Introdução**

Um sistema de tempo discreto pode ser modelado matematicamente como uma transformação, uma função ou um operador, que leva uma sequência de entrada a produzir uma sequência de saída, que é

y[*n*] = T{x[*n*]}

No contexto do PDS dizemos que um sistema processa um sinal de entrada e produz um sinal de saídal. Em geral, os sistemas são divididos em duas classes gerais: linear e não-linear. Sistemas Lineares são aqueles que obedecem ao princípio de sobreposição,

L{ a1.x1[*n*] + a2.x2[*n*] } = a1.L{ x1[*n*] } + a2.L{ x2[*n*] }

Sistemas lineares podem ser subdivididos em duas classes: invariante e variante no tempo ou dependente do tempo. Um sistema linear em que um par de entrada-saída é invariante para um deslocamento de tempo *n* é chamado de um Sistema Linear Invariante no Tempo ou SLIT.

SLTIs são muito importantes na prática, porque existe uma teoria de cálculo bem desenvolvido que nos permite analisar, projetar e estudar estes sistemas em grande detalhe. Em particular, podemos caracterizar completamente os SLITs pela sua resposta ao impulso. Se tivermos um SLIT e quisermos prever o que o sistema fará, a única coisa que precisamos fazer é introduzir um impulso e registrar a saída (resposta ao impulso). Uma vez que temos a resposta de impulso do SLIT, podemos utilizar a soma de convolução para encontrar a saída do sistema para uma entrada arbitrária.

2.1 Desenvolvimento da Soma de Convolução

Vamos tentar encontrar uma expressão para a saída de um SLIT para uma entrada arbitrária. Desde que não estejamos fazendo suposições sobre a seqüência de entrada x[*n*], devemos econtrar uma maneira de expressar essa seqüência em termos de algum outro para o qual conhecemos saída do SLIT. Por definicão, a saída de um SLIT L{·} para uma amostra unitária é a resposta ao impulso, denotada h[*n*]. Podemos, portanto, expressar a entrada arbitrária sinal x[*n*] utilizando δ[*n*-k] como nossas funções base. Isto é conveniente porque conhecemos a saída do sistema para um único impulso.

O o desenvolvimento de uma expressão para a saída do sistema (modelo) em função de uma entrada arbitrária é mostrado a seguir:

1. **Objetivo**: queremos encontrar uma expressão para y[*n*] para qualquer sinal de entrada x[*n*]

2. **Dado**: sabemos que a saída de um sistema quando a entrada é um impulso unitário é h[*n*].

y[*n*] = L{ x[*n*] } = L{ δ[*n*] } = h[*n*]

3. Uma vez que sabemos qual é a saída é para um impulso δ[*n*] nós só temos de representar o sinal de entrada x[*n*] em termos de impulsos unitários.



4. Se o sistema é invariante no tempo então sabemos que

L{ δ[n-k] } = h[n-k]

5. Portanto podemos expressar a saída do sistema como



onde usamos o fato do sistema ser linear e invariante ao tempo.

A expressão que foi obtida para a saída do SLIT, y[n], é chamada de soma da convolução linear e é denotada por:

2.2 Exploração Computacional

Vamos mostrar alguns exemplos de como usar o conceito de convolução em um sentido prático. Em particular, usaremos um sistema com resposta ao impulso muito simples e que pode ser usado como um filtro para suavizar sinais ruidosos.

2.3 Procedimentos

1. Gerar um sinal senoidal com ruído branco gaussiano aditivo (AWGN).

2. Gerar uma sequência para a resposta ao impulso, tal como: h[*n*] = { u[n] – u[n-8] } / 8.

3. Filtre o sinal de entrada ruidoso x[*n*] com o filtro caracterizado por *h*[*n*] usando a convolução.

4. Plote o resultado e experimente usar diferentes formas para o sinal *h*[*n*].

**3. Exemplo**

Vamos gerar um sinal discreto senoidal com frequência de 10 Hz, amostrado a 300 Hz, corrompido por um AWGN de média zero e variância unitária. Em seguida, vamos gerar uma seqüência resposta ao impulso h[n] = { u[n] – u[n-8] } / 8. Usando a operação da convolução vamos filtrar o sinal de entrada ruidoso *x*[*n*] com o filtro caracterizado por *h*[*n*]. Em seguida vamos plotar os resultados e experimenta diferentes coeficientes e formas para *h*[*n*].

A Figura 1 mostra a saída do sistema apresentado no exemplo anterior. Notar que a utilização deste filtro com uma resposta de impulso muito simples pode ajudar a reduzir o ruído gaussiano. A saída do sistema é menos ruidosa do que a sequência original. Este sistema é chamado de ***filtro de média móvel***, porque a saída é uma média das amostras contidas em uma janela temporal móvel, isto é



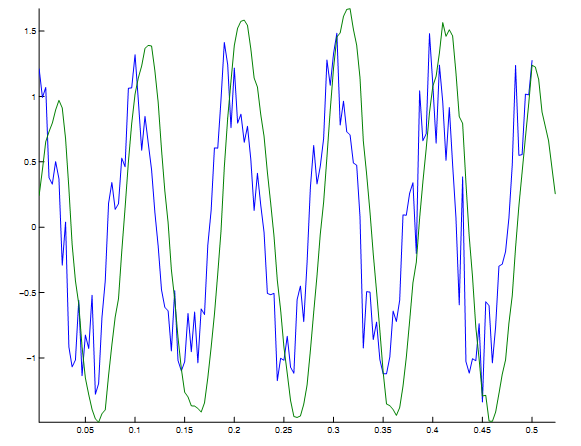
Neste caso todos os coeficientes do filtro são contantes e iguais a 1/8. Observe a relação entre essa expressão e a convolução. Vamos reescrever a saída de modo mais compacto:



Em geral os coeficientes não são iguais entre si, podendo assumir valores distintos a cada instante de tempo *k*:



Esta equação é a soma da convolução de um filtro de **Resposta Impulsiva Finita** (FIR) de ordem 7. A principal questão no projeto de um filtro digital é como definir os coeficientes da resposta ao impulso *h*[*n*], de modo que o ruído existente no sinal de entrada possa ser filtrado (eliminado). Geralmente estamos interessados na filtragem de algumas frequências em particular, e na permanência de outras frequências. Este projeto normalmente é feito no domínio da frequência.



**Figura 1**. Exemplo de um sistema discreto que suaviza o sinal de entrada e melhora a relação sinal-ruído (SNR).   
O sinal azul é uma senoide de 10 Hz, amostrada em 300 Hz e corrompida por um AWGN.   
A saída (sinal verde) é uma versão atrasada e suavizada do sinal de entrada.

Programa Matlab:

**% Gera sinal senoidal com ruído AWGN**

N = 150; **% Num. de amostras usadas**

n = 1:N; **% índice temporal**

fs = 300; **% frequência de amostragem**

T = 1/fs; **% período de amostragem**

f = 10; **% frequência da senóide (Hz)**

w = 2\*pi\*f; **% frequência angular (rad/s)**

x = sin(w\*n.\*T); **% sinal senoidal sem ruído**

a = 0.25; **% nível de ruído**

ns = a\*randn(1,N); **% ruído gaussiano: md=0 e var=1**

xn = x+ns; **% Sinal ruidoso (ruído aditivo)**

**% Gera a resposta ao impulso do filtro média móvel**

h = 1/8 \* ones(1,8);

% Filtra o sinal ruidoso usando h[n], xf[n] = xn[n]\*h[n]

xf = conv(xn,h); **% saída filtrada**

M = length(xf); **% compr. da convolução: len(xn)+len(h)-1**

plot(n\*T,xn,(1:M)\*T,xf);**%** **sinal ruidoso e sinal filtrado**

grid; axis tight; box off;

Programa Python:

**# -\*- coding: utf-8 -\*-**

**""" Lab.7 - Filtro de Média Móvel**

**Em: 31/03/2014**

**@author: Kaw**

**"""**

from numpy import arange, sin, pi, ones, convolve, reshape

from numpy.random import randn

from matplotlib.pylab import plot, axis, grid

**# Gera sinal senoidal com ruído AWGN**

N = 150 **#** **Num. de amostras usadas**

n = arange(0,N) **# índice temporal**

fs = 300 **# frequência de amostragem**

T = 1./fs **# período de amostragem**

f = 10 **# frequência da senóide (Hz)**

w = 2\*pi\*f **# frequência angular (rad/s)**

x = sin(w\*n\*T) **# sinal senoidal sem ruído**

a = 0.25 **# nível de ruído**

ns= a\*randn(1,N) **# ruído gaussiano: média=0 e variância=1**

xn = reshape(x+ns,len(x)) **# Sinal ruidoso (ruído aditivo)**

**# Gera a resposta ao impulso do filtro média móvel**

h = ones(8)/8 **# filtro de média móvel, ordem 7**

**# Filtra o sinal ruidoso usando h[n], xf[n] = xn[n]\*h[n]**

xf = convolve(xn,h) **# saída filtrada**

M = len(xf) **# compr. da convolução: len(xn)+len(h)-1**

plot(n\*T,xn,arange(0,M)\*T,xf) **# sinal ruidoso e sinal filtrado**

grid('on'); axis('tight')

**5. Questões**

1. Usando o Matlab/Octave/Python:
   1. Crie uma senóide de 30 Hz de 1,0 Vpp, amostrada em 300 Hz e com ruído aditivo branco gaussiano de 10% da amplitude da senóide, de média nula e variância unitária.
   2. Filtre a senóide ruidosa usando um filtro de média móvel com resposta ao impulso da forma *h*[*n*] = { *u*[*n*] – *u*[*n*-M] } / M, onde M é um parâmetro a ser especificado pelo usuário. Trace os resultados para cinco diferentes valores de M. Comente os resultados.
   3. O Matlab/Octave tem funções para implementar filtros digitais a partir de seus coeficientes; uma delas é chamada de [*filter*](http://octave.sourceforge.net/communications/function/filter.html) e a outra de [*filtfilt*](http://octave.sourceforge.net/signal/function/filtfilt.html). Usando o help do Matlab/Octave, repita o procedimento do item anterior usando essas funções. Comente as diferenças. O filtro implementado pela função *filtfilt* é um sistema causal?

x[n] = u[n] 🡺 X(z)=z/(z-1) |z| > 1

Y(z) = X(z).H(z) = z3/[(z-1)(z-1/2)(z-1/4)]

Y(z)/z = c1/(z-1) + c2/(z-1/2) + c3/(z-1/4) c1=8/3, c2=-2, c3=1/3

Y(z) = 8z/2(z-1) – 2z/(z-1/2) + z/3(z-1/4) |z| > 1

y[n] = [8/3 – 2(1/2)n + (1/4)n/3] u[n]

1. Trace a resposta em frequência do filtro de média móvel de ordem *N*, p/ *N* variando de 5 a 50, de 5 em 5. Comente as curvas obtidas.
2. Pesquise sobre um filtro de média denominado PROMEDIAÇÃO, e compare suas características, semelhanças e diferenças em relação ao filtro de média móvel.