DENEY 4: Matlab'de Temel Haberleşme Sistemleri Uygulamaları

AMAÇ: MATLAB programının temel özelliklerinin öğrenilmesi, analog işaretler ve sistemlerin sayısal bir ortamda benzetiminin yapılması ve incelenmesi.

DENEYİN YAPILIŞI

1. İŞARET ÜRETİMİ

MATLAB'te sayısal ortamda çalışıldığından, herhangi bir sinyalin sürekli zaman gibi görülebilmesi için örnekleme zaman aralığı yeterince küçük seçilmelidir. Örnekleme frekansı ile örnekleme zaman aralığı arasındaki bağıntı $f_s = 1/T_s$'dir.

Örnek 1

İlk olarak 10 Hz'lik sinüsoidal işaretin elde edilmesi incelenecektir.

```
%-------------------------------------------------------------------------
%program deney4_1.m
close all % Ekranda daha önce çizilmiş şekil varsa bu şekilleri kapatır.
clear all  % Daha önceden yapılmış bir işlem varsa hafızayı temizler.
clc % Komut penceresi ekranını temizler.
fm=10;   % İşaretin frekansı 10 Hz
A=100        % Örnekleme frekansı katsayısı
fs=A*fm     %Sinyalin örnekleme frekansı Hz;
ts=1/fs;
n=[0:(1/fs):1];                   % Sinyal 0'dan 1 saniyeye kadar
faz=0;%30
tsy=cos(2*pi*n*fm+faz);  % İşaretimiz
plot(n,tsy, 'k' );           %işaretin zaman izgesinde çizimi
title('Cosinus dalgasi')
xlabel('saniye');
ylabel('genlik');
% axis([xmin xmax ymin ymax])
%-------------------------------------------------------------------------
```

- Çıkışı gözlemleyip çiziniz.
- Örnekleme frekansının değişiminin (A=1, 2, 10 ve 100 değerleri için) etkilerini inceleyiniz.
- "**plot**" yerine "**stem**" komutunu kullanarak farkı gözlemleyiniz.

2. FOURİER DÖNÜŞÜMÜ

Fourier dönüşümü, herhangi bir işaretin frekans domeninde incelenmesini sağlar. MATLAB ortamında oluşturulan tüm işaretler ayrık olduğundan Fourier dönüşüm çifti olarak DFT ve IDFT kullanılır.
`fft(x)` komutu ile x vektörünün DFT'si FFT algoritması kullanılarak hızlı bir şekilde hesaplanır.
`fft(x,N)`komutu ile N noktalı DFT hesaplanır. Eğer x'in uzunluğu N'den küçük ise x'e sıfırlar eklenerek N boyuna getirilir. N değeri, x(n)'in örnek sayısından az olamaz. Eğer N değeri girilmezse N değeri işaretin boyutuna eşit olarak kabul edilir.

Örnek 2

Bu örnekte Fourier dönüşümü kullanılarak gürültülü bir işaret incelenecek ve asıl işaretin frekans bileşenlerini bulunacaktır. Bu amaçla `deney4_2` başlıklı programda, 50 Hz'lik 0.7 genlikli bir sinüs işaret ile 120 Hz'lik birim genlikli sinüs işaret toplanarak, elde edilen işarete sıfır-ortalamalı rasgele gürültü eklenmiştir. Örnekleme frekansı 1000 Hz'dir. Frekans bölgesinde inceleme yapmak için Hızlı Frekans Dönüşümü (FFT) kullanılarak gürültü işaretin ayrık frekans dönüşümü elde edilmiştir.

```
%-----------------------------------------------------------------------------
%program deney4_2.m
close all
clear all
clc
Fs = 1000;                    % Örnekleme frekansı
T = 1/Fs;                     % Sembol periyodu
L = 1000;                     % İşaretin uzunluğu
t = (0:L-1)*T;                % Zaman vektörü
% 50 Hz lik ve 120 Hz lik sinusozoidal işaretin toplamı
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t));     % Gürültülü işaret
plot(Fs*t(1:50),y(1:50))
title('gurultu ile bozulmus isaret')
xlabel('zaman (ms)')
NFFT = 2^nextpow2(L); % FFT uzunluğunun hesaplanışı
Y = fft(y,NFFT)/L; % normalize edilmiş FFT
f = Fs/2*linspace(0,1,NFFT/2);
figure
% Plot tek-yanlı genlik spektrumu
plot(f,2*abs(Y(1:NFFT/2)))
title('Tek-yanli genlik spektrumu')
xlabel('Frekans(Hz)')
ylabel('|Y(f)|')
%-----------------------------------------------------------------------------
```

- Çıkış grafiklerini çiziniz.
- İşaretin uzunluğunun (L) arttırılmasının etkilerini gözlemleyiniz.(Ör. L=10000)

Örnek 3

Bu örnekte FFT uzunluğu N değerinin değişimi incelenecektir. `deney4_3` programında bir periyodunda 10 örnek olan ve toplamda 30 örnekten oluşan bir kosinüs işareti oluşturulmuştur. N için 3 farklı değer tanımlanıp, her bir N değeri için x(n) işaretinin dönüşümünü hesaplanmıştır. N-noktalı FFT için frekans ekseni 0'dan N-1'e kadardır. Normalize ederek 0'dan 1-(1/N)'e kadar olması sağlanır. Sonuç eğrileri "plot" komutu ile elde edilir.

```
%-----------------------------------------------------------------------------
%program deney4_3.m
n = [0:29];
x = cos(2*pi*n/10);
N1 = 15;
N2 = 30;
N3 = 64;
```

```
N4 = 256;
X1 = fft(x,N1);
X2 = fft(x,N2);
X3 = fft(x,N3);
X4 = fft(x,N4);
F1 = [0 : N1 - 1]/N1;
F2 = [0 : N2 - 1]/N2;
F3 = [0 : N3 - 1]/N3;
F4 = [0 : N4 - 1]/N4;
subplot(4,1,1)
plot(F1,abs(X1),'-x'),title('N = 15'),axis([0 1 0 20])
subplot(4,1,2)
plot(F2,abs(X2),'-x'),title('N = 30'),axis([0 1 0 20])
subplot(4,1,3)
plot(F3,abs(X3),'-x'),title('N = 64'),axis([0 1 0 20])
subplot(4,1,4)
plot(F4,abs(X4),'-x'),title('N = 256'),axis([0 1 0 20])
Inv_X1 = ifft(X1,N1);
Inv_X2 = ifft(X2,N2);
Inv_X3 = ifft(X3,N3);
Inv_X4 = ifft(X4,N4);
figure,
subplot(4,1,1)
plot(Inv_X1,'-x'),title('N = 15')
subplot(4,1,2)
plot(Inv_X2,'-x'),title('N = 30')
subplot(4,1,3)
plot(Inv_X3,'-x'),title('N = 64')
subplot(4,1,4)
plot(Inv_X3,'-x'),title('N = 256')
%-----------------------------------------------------------------------
```

- Elde edilen şekilleri inceleyiniz ve çiziniz.
- FFT uzunluğunun (N) işaretin uzunluğundan küçük, eşit ya da büyük olması durumlarını yorumlayınız
- N değeri işaretin uzunluğundan küçük olduğunda FFT-IFFT sonucunda işaretin elde edilemediğini gösteriniz.

Örnek 4

Bir önceki örnekte x(n) işaretinin uzunluğu 3 periyot ile sınırlanmıştı. Bu bölümde büyük bir N değeri seçilecek ve temel periyodun tekrarlama sayısının değişimi incelecektir.

```
%-----------------------------------------------------------------------
%program deney4_4.m
n = [0:29];
x1 = cos(2*pi*n/10); % 3 periyot
x2 = [x1 x1]; % 6 periyot
x3 = [x1 x1 x1]; % 9 periyot
N = 2048;
X1 = abs(fft(x1,N));
X2 = abs(fft(x2,N));
X3 = abs(fft(x3,N));
F = [0:N-1]/N;
subplot(3,1,1)
plot(F,X1),title('3 periyot'),axis([0 1 0 50])
subplot(3,1,2)
```

```
plot(F,X2),title('6 periyot'),axis([0 1 0 50])
subplot(3,1,3)
plot(F,X3),title('9 periyot'),axis([0 1 0 50])
%-------------------------------------------------------------------------
```

- Programdan elde edilen grafikleri çiziniz.
- İşaretin zaman bölgesindeki süresinin artmasının frekans bölgesinde oluşturduğu etkilerini yorumlayınız.

3. KONVOLÜSYON

Ayrık zamanlı, doğrusal zamanla değişmeyen sistemlerde (LTI), sistemin $x(n)$ girişine verdiği cevabı bulmak için, $x(n)$ dizisi ve sistemin birim dürtü cevabı $h(n)$ dizisinin konvolüsyonu hesaplanır.

$$y(n)=x(n)*h(n) \tag{1}$$

$$y(n)=\sum_{n=-\infty}^{\infty} x(k)h(n-k) \tag{2}$$

Matlab'de bu işlem **conv** komutu ile gerçekleştirilir.

Örnek 5

Ayrık zamanlı

$$x_1(n_1)=1-1.3e^{\left(\frac{n_1}{5}\right)} \qquad -2 \geq n_1 \geq 1$$

$$x_2(n_2)=e^{(-0.7n_2)} \qquad 0 \geq n_2 \geq 4$$

işaretleri için

- x_1 ve x_2 işaretlerini çiziniz.
- "conv" komutu ile dizilerinin konvolüsyonunu hesaplayınız ve yeni bir şekilde çizdiriniz.

4. SÜZGEÇ YAPILARI

Sayısal bir sistemin transfer fonksiyonu H(z) olarak gösterilmektedir. Bu transfer fonksiyonu sistemin bir giriş işaretine olan etkisini ve aynı zamanda sistemin süzgeç etkisini tanımlar. Süzgeçler verdikleri frekans cevabına göre

Alçak geçiren (Lowpass, LP)
Yüksek geçiren (High pass, HP)
Bant geçiren (Bandpass, BP)
Bant durduran 'geçirmeyen' (Band reject, BR)
Tüm geçiren (All pass, AP)

şeklinde ayrılmaktadır.

Transfer fonksiyonu H(z)'nin genel biçimi

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+...+b_nz^{-n}}{a_0+a_1z^{-1}+a_2z^{-2}+...+a_nz^{-n}} \tag{3}$$

olarak verilmekte ve *n*.derece sayısal süzgeç yapısını belirtmektedir. Sayısal süzgeçler, standart fark denklemleri ile de tanımlanır. Genel olarak fark denklemi;

$$y(n)=\sum_{k=N_1}^{N_2} b_k x(n-k)-\sum_{k=1}^{N_3} a_k y(n-k) \tag{4}$$

biçimindedir. (4) eşitliğinde N_1=0 değeri için, a_0=1 olacak şekilde transfer fonksiyonu elde edilir. Örnek olarak;

$$H_1(z)=\frac{0.2066+0.4131z^{-1}+0.2066z^{-2}}{1-0.3695z^{-1}+0.1958z^{-2}}$$

transfer fonksiyonuna karşılık gelen fark denklemi

$$y(n)=0.2066x(n)+0.4131x(n-1)+0.2066x(n-2)+0.3695y(n-1)-0.1958y(n-2)$$

biçimindedir. Transfer fonksiyonu verilen bir sistemi incelemek için, transfer fonksiyonun genlik ve faz eğrilerinin çizilmesi gerekir.

Matlab'deki **freqz** komutu ile H(z) değerleri elde edilir.

```
[H,W] = FREQZ(B,A,N)

    H(z)=B(z)/A(z) biçimindeki transfer fonksiyonunun değerleri elde
    edilir. Burada B, B(z)'nin katsayı vektörü, A ise A(z)'nin katsayı
    vektörüdür. N ise H(z)'nin hesaplandığı nokta sayısını belirtir.
```

Fark denklemi, giriş işaretinden çıkışın elde edilme adımlarını gösterir. Örnek olarak

$$y(n)=0.04x(n-1)+0.17x(n-2)+0.25x(n-3)+0.17x(n-4)+0.04x(n-5)$$

$$y(n)=0.42x(n)-0.42x(n-2)+0.44y(n-1)-0.16y(n-2)$$

$$y(n)=0.33x(n+1)+0.33x(n)+0.33x(n-1)$$

eşitliklerini inceleyelim. Bu üç fark denklemi farklı karakteristiklere sahip filtreleri temsil etmektedir.

İlk filtrenin çıkışı sadece giriş işaretinin geçmiş değerlerine bağlıdır. Örneğin $y(10)$'u hesaplamak için $x(9)$, $x(8)$, $x(7)$, $x(6)$ ve $x(5)$ gereklidir. Bu tür bir süzgeç *Sonlu Dürtü Yanıtlı* (FIR, Finite Impulse Response) süzgeç olarak adlandırılır. FIR süzgeçlerin, transfer fonksiyonlarının (H(z)'nin) paydası 1'dir.

İkinci süzgeç ise giriş işaretinin geçmiş değerlerinin yanı sıra çıkış işaretinin de geçmiş değerlerini kullanarak yeni çıkış işaretini elde eder. Bu tip süzgeçler ise *Sonsuz Dürtü Yanıtlı* (IIR, Infinite Impulse Response) süzgeç olarak adlandırılırlar. IIR süzgeçlerin transfer fonksiyonun paydası sabit bir değer değildir.

Üçüncü süzgeç de FIR yapıdadır, çünkü çıkış değeri sadece giriş işaretine bağlıdır. Fakat burada giriş işaretinin gelecek değerleri de gereklidir. Bu durum giriş işaretinin gerçek zamanlı üretildiği durumlarda sorun olabilmektedir.

MATLAB'de herhangi bir giriş işaretine sayısal süzgeç uygulamanın en kolay yolu **filter** fonksiyonunu kullanmaktır.

```
filter(B,A,x)

      H(z)=B(z)/A(z) biçimindeki sayısal filtreyi giriş işareti x'e
      uygular. B, B(z)'nin katsayı vektörü, A ise A(z)'nin katsayı
      vektörüdür.
```

IIR Süzgeç Tasarımı

MATLAB'de sayısal süzgeç tasarımında kullanılan fonksiyonlar analog süzgeç yapılarına dayanmaktadır. Bunlar, Butterworth, Chebyshev Type 1, Chebyshev Type 2 ve Elliptic süzgeçlerdir.

Butterworth süzgeç için kullanılan komut **butter**'dır. **buttord** komutu ile istenilen zayıflamaya sahip filtre derecesi belirlenmektedir.

```
[B,A] = BUTTER(N,Wn)

      N. dereceden alçak geçiren süzgeç tasarlar ve N+1 uzunluğunda B (pay)
      ve A(payda) süzgeç katsayılarını verir. Kesim frekansı Wn , 0.0 < Wn
      < 1.0 arasındadır. Burada 1.0 örnekleme hızının yarısını
      göstermektedir. Eğer Wn iki bileşenden oluşuyorsa Wn = [W1 W2] , 2N
      dereceli geçiş bandı W1 < W < W2 şeklinde olan süzgeç elde edilir.
      Ayrıca üst geçiren süzgeç [B,A] = BUTTER(N,Wn,'high') ile band
      durduran süzgeç ise [B,A] = BUTTER(N,Wn,'stop') ile tasarlanabilir.

[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs)

      Geçiş bandında en fazla Rp (dB) kadar ve durdurma bandında en az Rs
      (dB) değeri kadar kayıp veren en düşük dereceli sayısal Butterworth
      süzgeçin derecesini verir. Wp ve Ws geçiş ve durdurma bandının 0 ile
      1 arasında normalize edilmiş köşe frekanslarını göstermektedir.
      Fonksiyonun çıkışı olan Wn ise istenen özellikte süzgeç için gerekli
      olan frekansı vermektedir. Wp ve Ws için örnek değerler
      Altgeçiren: Wp=0.1 Ws=0.2
      Üstgeçiren: Wp=0.2 Ws=0.1
      Bandgeçiren: Wp=[0.2 0.7], Ws=[0.1,0.8]
      Band Durduran: Wp=[0.1 0.8], Ws=[0.2 0.7];
```

Benzer şekilde diğer süzgeç yapıları için kullanılan komutların listesi ;

```
      cheb1ord    Birinci çeşit Chebyshev süzgeç derecesi bulma
      cheby1      Chebyshev 1. çeşit sayısal ve analog süzgeç tasarımı
      cheb2ord    İkinci çeşit Chebyshev süzgeç derecesi bulma
      cheby2      Chebyshev 2. çeşit sayısal ve analog süzgeç tasarımı
      ellipord    Elliptic süzgeç derecesi bulma
      ellip       Elliptic veya Cauer sayısal ve analog süzgeç tasarımı
```

Örnek 6

Bu bölümde Butterworth Süzgeç tasarımı programı verilmektedir. Örnekte verilen alt geçiren süzgeç tasarımıdır ancak %'li kısımlar kaldırılarak diğer tasarımlarında nasıl yapılabileceği görülebilir.

```
%-----------------------------------------------------------------------------
%program deney4_6.m
clear all;close all;clc
close all
wg=[0.25];   % Alt geçiren
wd=[0.5];
% wg=[0.2] % Üst geçiren
% wd=[0.1]
%wg=[0.25 0.5];  % Band geçiren
%wd=[0.1 0.7];
%wg=[0.25 0.5];  % Band Durduran
%wd=[0.1 0.7];
gddb=1;
sddb=40;
[N,Wn]=buttord(wg,wd,gddb,sddb);
[B,A] = BUTTER(N,Wn); % LPF
fs=2000;
[H,W] = FREQZ(B,A,fs/2+1);

Hg=20*log10(abs(H));
plot(W/pi,Hg),title('Filtrenin genlik
yaniti'),ylabel('Genlik'),xlabel('Frekans')

fm1=100;               % İşaretin frekansı
fm2=1000;
ts=1/fs;
n=[0:(1/fs):1];                  % Sinyal 0'dan 1 saniyeye kadar
x1=cos(2*pi*fm1.*n);  % İşaretimiz
x2=cos(2*pi*fm2.*n);  % İşaretimiz

s=x1+x2;

% figure(1),plot(x1),title('fm=100'),axis([0 200 -1 1])
% figure(2),plot(x2),title('fm=1000'),axis([0 200 -1 1])

H=filter(B,A,s);

figure
subplot(411),plot(x1),axis([0 200 -2 2]),title('x1')
subplot(412),plot(x2),axis([0 200 -2 2]),title('x2')
subplot(413),plot(s),axis([0 200 -2 2]),title('x1 ve x2 nin toplami')
subplot(414),plot(H),axis([0 200 -2 2]),title('Filtre cikisi')
%-----------------------------------------------------------------------------
```

- Programı inceleyip, çalıştırınız.
- Alçak geçiren filtre için çıkış işaretini elde edip çiziniz.
- Programda gerekli değişiklikleri yaparak yüksek geçiren filtre için çıkışları çiziniz.