

Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications

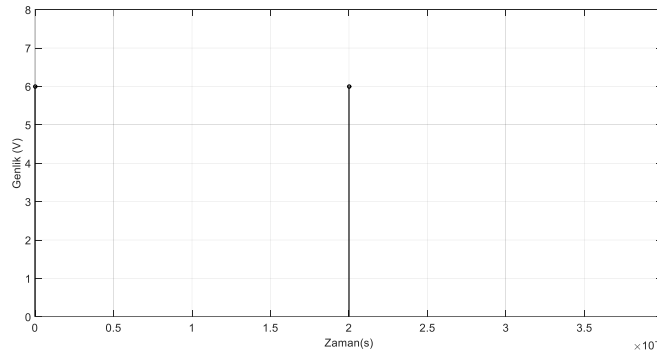
1. Örnekleme

Öncelikle boş bir “m” dosyası oluşturarak aşağıdaki kodları bu boş “m” dosyasının içine yazılacaktır. Periyodik bir sinyal olan $x(t) = A\cos\left(\frac{2\pi}{T_0}t\right) = 6\cos(2000\pi t)$ sinyali incelenmek üzere kullanılacaktır. Dikkat edilecek olursa burada periyot $T_0 = 10^{-3} = 0.001s$ ’dir. Dolayısı ile işaretin frekansı $f = 1000Hz$ ’tir. Ayrıca incelenecek işaret sinüsoidal bir işaret olduğundan periyodunun tersi olan $\left(\frac{1}{T_0}\right)$ tekrarlanma hızı aynı zamanda bu sinyalin içerdiği tek pozitif frekans bileşenidir. Bu özellik sadece sinüsoidal işaretlere özgü bir özellik olup diğer işaretler için söylenemez. Başka bir deyişle sinüsoidal işaret dışındaki diğer işaretler çok sayıda frekans bileşeni içerirler. Bu frekans bileşenleri de aslında farklı frekans, genlik ve fazlardaki sinüsoidal işaretlerdir. Bu durumda bütün işaretlerin farklı frekans, genlik ve fazlardaki sinüsoidal işaretlerin toplam ve farkları biçiminde ifade edilebileceği söylenebilir. İncelenen işaretin $0 \leq t \leq 0.004$ saniye aralığındaki dört periyodu matlab yardımı ile oluşturulmaya çalışılsın. Ancak $0 \leq t \leq 0.004$ aralığında zaman (t) sonsuz tane değer alır. Dolayısı ile bilgisayar belleğinin sonsuz adet değeri saklayabilmesi yada işlemcinin sonsuz adet değerle işlem yapabilmesi mümkün değildir. Bu durumda işaretten belli zaman aralıklarında örnekler alarak oluşturulmak işarete benzeyen bir işaret elde edilebilir. Oluşturulmak istenen işaretin sahip olduğu tek pozitif frekans bileşeni $1000Hz$ göz önüne alınarak örnek alma işlemi yapılsın. En uygun örnek alma hızını (örnekleme frekansı f_s) tespit etmek için denemeler yapılsın.

Program 1.

```
fs=500;%örnekleme frekansı  
T=1/fs;%örnekleme periyodu  
t=0:T:0.004;%0.004 saniyelik zaman dilimi  
x=6*cos(2*pi*1000*t);%oluşturulan sinyal
```

Oluşturulan sinyalin grafiğini $stem(t,x)$ komutu ile çizdiriniz.



Şekil 1. Örnekleme frekansı 500Hz için elde edilmesi gereken işaret

Program 1.’deki $fs=500$ ifadesi $fs=1000$ olarak değiştirilsin ve grafik tekrar çizdirilsin.

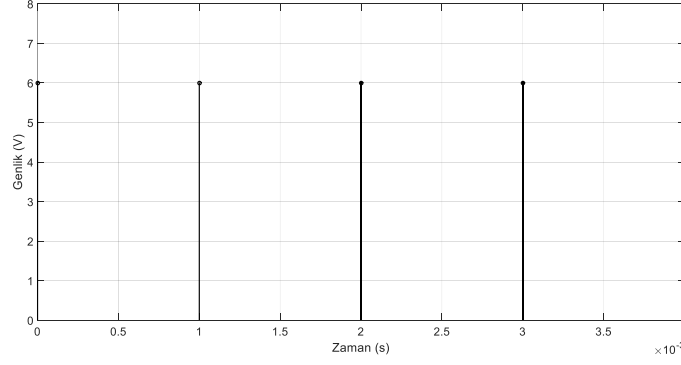
Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

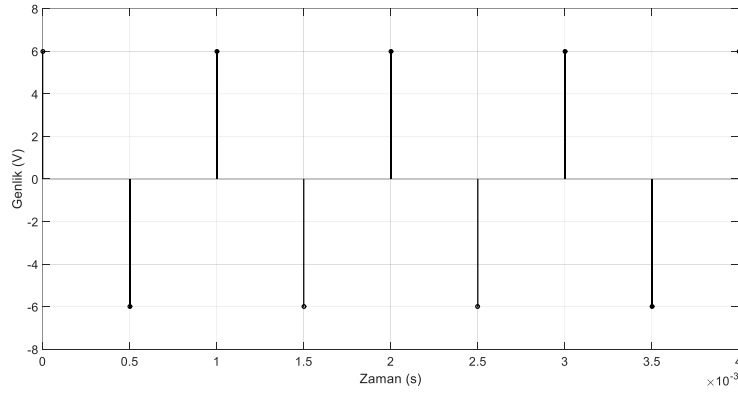
İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications



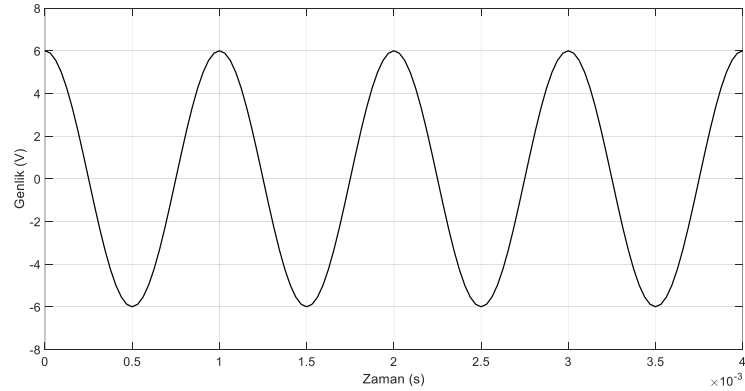
Şekil 2. Örnekleme frekansı 1000Hz için elde edilmesi gereken işaret

Program 1.'deki $f_s=500$ ifadesi $f_s=2000$ olarak değiştirilsin ve grafik tekrar çizdirilsin.



Şekil 3. Örnekleme frekansı 2000Hz için elde edilmesi gereken işaret

Oysa esas elde edilmesi gereken işaretin grafiği aşağıdaki gibidir.



Şekil 4. $6\cos(2000\pi t)$ işaretinin grafiği

Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications

Daha önce elde edilen işaretlerden Şekil 3.'te verilene dikkat edilecek olursa +6V ve -6V noktaları birleştirilirse elde edilmesi gereken işarete benzeyen bir işaret elde edilebilir. Gerçekte de bir işaret örneklenirken o işaretin içerdiği en yüksek pozitif frekansın en az iki katı bir örnekleme frekansı ile örneklenmelidir buna Nyquist örnekleme kriteri denilir.

2. Nicemleme (Kuantalama)

Herhangi bir işareti sayısal olarak (1 ve 0'lar şeklinde) ifade edebilmek birçok avantaj sağlar. Bunlardan birkaçını belirtmek gerekirse; sayısal bilgiyi belleğe kaydetmek daha kolaydır flash disk, bellek kartı yada harddisk gibi hacimce daha küçük depolama birimlerinde sayısal bilgi depolanabilir. Ayrıca bir sinyale karşılık gelen 1 ve 0'lar çeşitli algoritmalar ile azaltılabilir örneğin ard arda gelen 10 tane 0 yerine 10-0 kullanılabilir bu sayede de sıkıştırma denilen işlem gerçekleştirilmiş olur ve depolama için gerekli olan bellek miktarı azalır. Bir diğer avantaj ise sayısal verinin hatalara karşı kodlanabilmesidir. Örneğin belli uzunluktaki (örneğin 800bit) bir sayısal verinin sayısal karşılığının bilinen bir sayıya bölümünden kalan da bu 800bit ile birlikte iletilirse, alıcıda 800bitten oluşan veri kısmı aynı bilinen sayıya bölünebilir kalan bulunabilir ve vericinin ilettiği kalan ile karşılaştırılıp hata olup olmadığı belirlenebilir. Hata olduğu tespit edildiğinde vericinin bunu tekrar göndermesi istenebilir yada kullanılan hata kodlama algoritmasına göre bu hata düzeltilebilir. Sayısal verinin bir diğer avantajı ise kendisine kolayca gerekli fazladan sayısal bilgi eklenebilmesidir. Örneğin iletilmek istenen 800bitlik veriye hata kodlamanın gerektirdiği ekstra bitler eklenebildiği gibi, hedef adresi, gönderen adresini belirten ek bitlerde konulabilir. Bu sayede aynı kablo yada ortam üzerinden video, resim, ses ve metin iletilebildiği gibi ortamın çoklu kullanıcıya paylaştırılması kolaylaşmış olur.

Örnekleme kısmında belirtildiği gibi bir sinyali bilgisayarların işleyebileceği sayısal veriye dönüştürmek için ondan zamanın belli aralıklarında örnekler almak gerekir. Ancak alınan bu örnek değerlerinin hangi genliğe sahip olacağını önceden bilebilmek mümkün değildir. Yani örnek değerlerinin Şekil (3)'teki gibi +6 yada -6 gibi iki farklı değerden birini alması gerçek uygulamalarda mümkün değildir. Oysa örnek değerlerinin örneğin 3bit ile kodlanıp sayısala dönüştürülebilmesi için $2^3 = 8$ farklı değerden birini alması gerekir. Bu sebeple örnek değerleri daha önceden belirlenen değerlerden birine yuvarlanır bu işleme nicemleme (kuantalama) denilir. Şekil 5.'te 3bitlik bipolar bir kuantalayıcının giriş çıkış karakteristiği gösterilmektedir [xxxx]. Şekil 5.'te yatay eksen girişi düşey eksen ise çıkışı göstermektedir. Dikkat edilecek olursa çıkış değerleri $0, \Delta, 2\Delta, 3\Delta, -\Delta, -2\Delta, -3\Delta, -4\Delta$ değerlerinden birine eşit olmak zorundadır. Her bir çıkış değerinin ise ikili düzende 3bitlik bir sayı karşılığı vardır. Şekil 5.'e göre giriş olan örnekleme sonucundaki örnek değeri $\left(-3\frac{\Delta}{2}, -\frac{\Delta}{2}\right]$ arasında değer alıyorsa $-\Delta$ değerine kuantalanacaktır. Yani $\left(-3\frac{\Delta}{2}, -\frac{\Delta}{2}\right]$ arasındaki sonsuz tane farklı değer hepsi $-\Delta$ değerine kuantalanacaktır. Yapılacak olan hatanın uç değerleri $-\frac{3\Delta}{2}$ değeri $-\Delta$ 'ya kuantalandığında $e_q = -\Delta - \left(-\frac{3\Delta}{2}\right) = \frac{\Delta}{2}$ ve $-\frac{\Delta}{2}$ değeri $-\Delta$ 'ya kuantalandığında $e_q = -\Delta - \left(-\frac{\Delta}{2}\right) = -\frac{\Delta}{2}$ olacaktır. Bu durum Şekil 5.'ten görülebilmektedir.

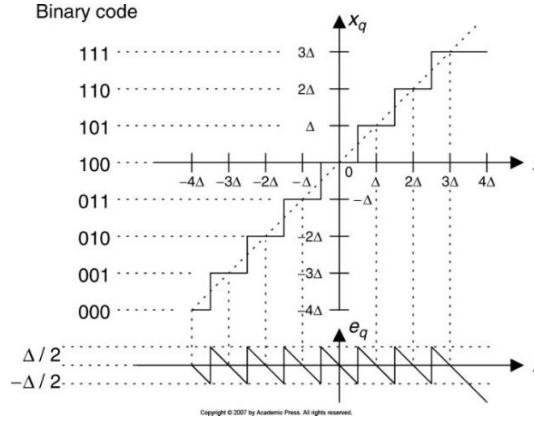
Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications



Şekil 5. Bipolar kuantalayıcı karakteristiği

Aşağıdaki program ile bir işaret önce örneklenmekte ve sonra kuantalanmaktadır. Örneklenecek olan işaret $x(t) = 6\cos(2000\pi t)$ olsun. Program 1'deki örnekleme frekansı kısmını $f_s=32000$ olarak değiştirilsin. Kuantalama yaparken giriş sinyalinin $-6.5V$ ile $+6.5V$ arasında değerler aldığı varsayalım. Kuantalayıcının çıkışının 2 bit ile kodlandığı varsayılmaktadır.

Program 2

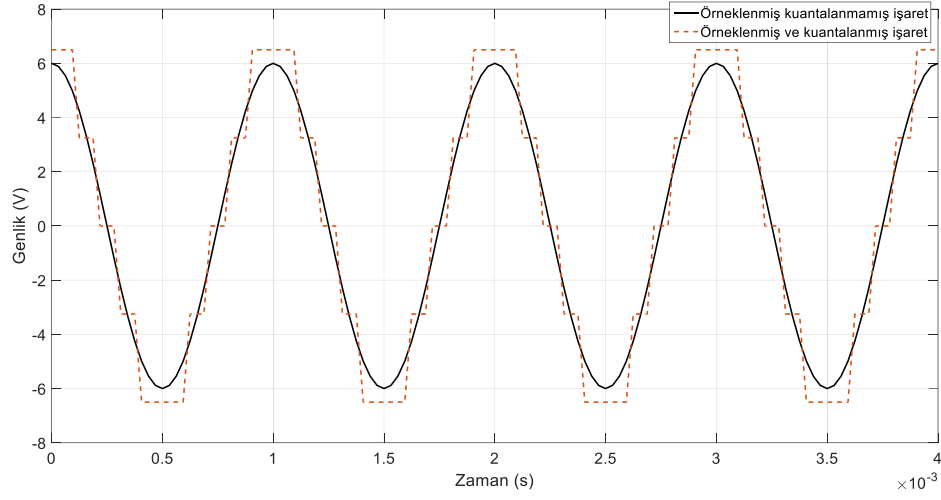
```
bitsay=2; %kuantalayıcı çıkışının kod uzunluğu
xmin=-6.5; %kuantalayıcı girişinin minimum değeri
xmax=6.5; %kuantalayıcı girişinin maximum değeri
L=2^bitsay; %seviye sayısı
delta=(xmax-xmin)/L; %adım büyüklüğü
boy=length(x); %giriş sinyalinin örnek sayısı
for k=1:boy
    I=round((x(k)-xmin)/delta); %alınan değer minimum değere yuvarlatılmış olarak kaç adım
    %uzaklıkta olduğu
    if (I==boy)
        I=I-1; %+6.5V'u aşmamak için
    end
    if (I<0)
        I=0; %-6.5V'u aşmamak için
    end
    xq(k)=xmin+I*delta; %kuantalanmış değer
end;
```

Kuantalanmış sinyal ve kuantalanmamış sinyalin grafiği `plot(t,x,t,xq);` komutu ile üst üste çizdirelebilir. Elde edilmesi gereken sonuç aşağıdaki gibidir. Ancak burada çizimde `stem` yerine `plot` komutu kullanıldığından örneklerin arasındaki boşluklar matlab tarafından doldurulmuştur. Aslında gerçekte durum bu şekildedir. Yani örneklenmiş, kuantalanmış bir işaret ikili düzene dönüştürüldükten sonra tekrar sinyal formuna dönüştürülürken (sayısalan analoga dönüşüm) matlabın yaptığı gibi aradaki boşluklar bir alçak geçiren süzgeç yardımı ile doldurulur.

Sayısal İşaret İşleme Dersi Laboratuvarı

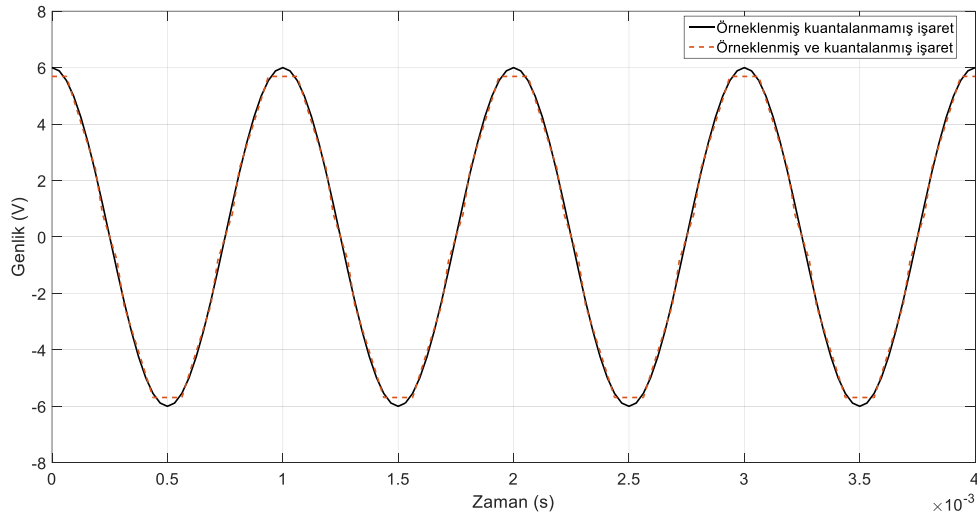
Ad Soyad:
Numara:
İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications



Şekil 6. $6\cos(2000\pi t)$ işaretinin örnekleme ve 4 seviyeye kuantalanmış biçimleri

Kuantalama seviye sayısı 4'ten 16'ya Program 2'de *bitsay=4;* olarak değiştirilerek çıkarıldığında, elde edilmesi gereken grafik aşağıdaki gibi olacaktır.



Şekil 7. $6\cos(2000\pi t)$ işaretinin örnekleme ve 16 seviyeye kuantalanmış biçimleri

Görüldüğü gibi kuantalama seviyesi arttıkça kuantalama hatası azalmaktadır. Kuantalama hatası aşağıdaki gibi elde edilebilir.

Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

İmza:

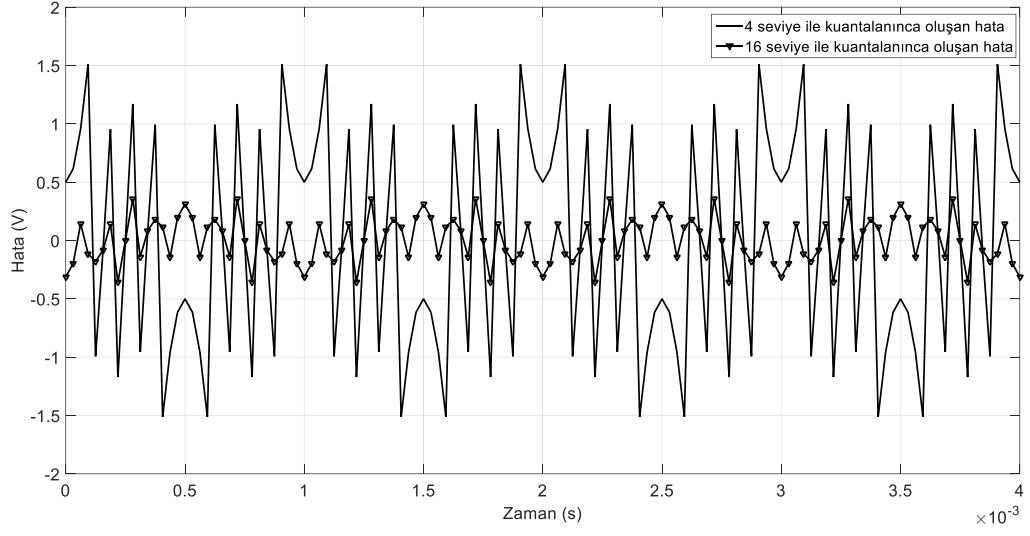
Li Tan Digital Signal Processing: Fundamentals and Applications

Program 3

$e=xq-x$; %anlık kuantalama hatası

$\text{plot}(t,e)$; %anlık kuantalama hatasının grafiği

$e_{ort}=\text{mean}(\text{abs}(e))$; %ortalama kuantalama hatası



Şekil 8. 4 Seviye ve 16 seviye kuantalayıcı çıkışında oluşan anlık kuantalama hataları

Sayısal İşaret İşleme Dersi Laboratuvarı

Ad Soyad:

Numara:

İmza:

Li Tan Digital Signal Processing: Fundamentals and Applications

3. Değerlendirme

Bu kısımda `load 'train.mat'` komutu ile örneklenmiş bir ses ele alınsın. Matlab komut satırına `load 'train.mat'` yazıldıktan sonra matlab workspace'te `y` isimli dizi ve `Fs` isimli değer oluşacaktır. Bunlardan `y` örneklenmiş sesi içeren diziyi `Fs` ise örnekleme frekansını içerir. Program 2'de aşağıdaki değişiklikleri yapınız.

`fs=Fs;%örnekleme frekansı`

`T=1/fs;%örnekleme periyodu`

`boy=length(y);% örnek sayısı`

`t=[0:1:boy-1]*T;%sinyalin zaman dilimi`

`x=y/max(abs(y)); %sinyal mutlak değerinin maksimumu ile normalize ediliyor`

Programınızı çalıştırınız.

`subplot(3,1,1);` yazdıktan sonra `plot(t,x);` ile kuantalanmamış sinyali çizdiriniz.

Sinyaliniz kaç sn uzunluğunda? _____

Sinyali Program 2'de aşağıdaki değişiklikleri yapıp kuantalayınız.

`bitsay=2; %kuantalayıcı çıkışının kod uzunluğu`

`xmin=-1; %kuantalayıcı girişinin minimum değeri`

`xmax=1;% kuantalayıcı girişinin maximum değeri`

`subplot(3,1,2);` yazdıktan sonra yukarıdakine benzer biçimde kuantalanmış sinyalin grafiğini çizdiriniz.

`subplot(3,1,3);` yazdıktan sonra kuantalama hatasını Program 3'teki gibi elde edip çizdiriniz.

Kuantalanan sinyali dinleyiniz.

`sound(xq,Fs);`

Şimdide kuantalanmamış sinyali dinleyiniz.

`sound(x,Fs);`

Sinyalin maksimum ve minimum değerlerini (`xmax`, `xmin`) -1.5V ile +1.5V olarak değiştiriniz.

Kuantalama hatası nasıl değişti? _____

Kuantalama hatası neden değişti? _____

Şimdide kod uzunluğunu 2 bitten 8 bite çıkarınız ve elde ettiğiniz sonuçları yorumlayınız.