

ISTANBUL TECHNICAL UNIVERSITY
COMPUTER AND INFORMATICS FACULTY
COMPUTER ENGINEERING

BLG438E
DIGITAL SIGNAL PROCESSING LABORATORY

DENEY-3

Merve Cansu KÖSEOĞLU
150120704

Mehmet AYSEVİNÇ
150110705

INSTRUCTOR: ASSOC. PROF. Berk ÜSTÜNDAĞ
RES. ASST. Hasan ÜNLÜ

Deneyin Amacı

Deneyde öncelikle iki farklı frekans değerine sahip sinüs sinyallerinin toplanması ve daha sonra bu sinüslerin toplamından oluşan gürültünün (bozulan sinyal) IIR filtre yardımıyla gürültüyü bastırarak normal bir sinüs sinyaline dönüştürülmesi amaçlanmıştır.

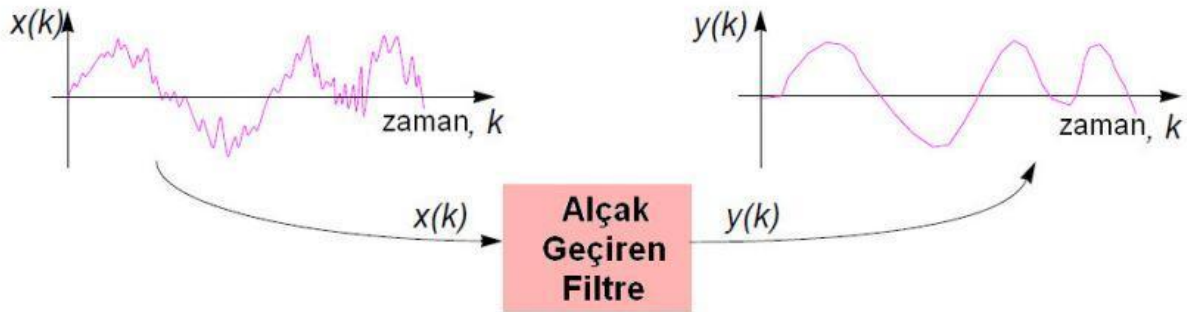
Bu amaçla IIR filtre fark denkleminden yararlanılacaktır. Fark denkleminin katsayıları için ise fdatool yardımıyla oluşturulan (IIR-Elliptic) SOS değişkeni kullanılacaktır.

Deney ile İlgili Ek Bilgi

Digital filtreler genellikle fark denklemlerine göre tanımlanırlar ve dizayn edilirler. Ve davranışlarına göre digital filtreleri 4 gruba ayırabiliriz. Bunlar;

1. Alçak Geçiren
2. Band Geçiren
3. Yüksek Geçiren
4. Çentik Geçiren

Deneyimizde kullandığımız IIR filtre alçak geçiren özellik göstermektedir. Aşağıdaki örneğimizde $x(k)$ sinüs fonksiyonu, 2 sinüs fonksiyonun toplamı sonucu oluşan gürültülü bir fonksiyondur. Bu gürültülü fonksiyon alçak geçiren filtre yardımıyla toplam sonucu oluşan gürültüleri büyük bir ölçüde azalmaktadır. Nitekim bu deneyimizde de oluşan gürültü bu şekilde azaltılmıştır.



Deneyimizde kullandığımız fark denkleminin genel yapısı aşağıda verilmiştir.

$$y[n] = 1/a_0(b_0x[n] + b_1x[n-1] + \dots + b_px[n-p] - a_1x[n-1] - a_2x[n-2] - \dots - a_qx[n-q])$$

Burada p ileri beslemeli filtre derecesi,
 b_i ileri beslemeli filtre katsayılarını,
 q geri beslemeli filtre derecesi,
 a_i geri beslemeli filtre katsayılarını,
 $x[n]$ giriş sinyalini,
 $y[n]$ ise çıkış sinyalini göstermektedir.

Deneyde kullanılan katsayılar aşağıda verilmiştir. Bu katsayıları elde etmek için workspace ekranından `fdatoool` komutu girilip açılan ekranda IIR-Elliptic özelliği seçildikten sonra “Design Filter” seçeneği seçilir. Filtre oluştuktan sonra katsayılar seçeneği seçildiğinde 4 farklı section’a ait pay ve payda katsayıları gelmektedir. Bu katsayılar IIR filtrenin feedforward ve feedback katsayılarıdır. Katsayılar export edilip MATLAB kodu için SOS isimli bir değişken olarak kullanılabilir. Bu katsayılar aşağıda gösterilmektedir. 1-2-3 sütunları pay değerlerini, 4-5-6 ise payda değerlerini temsil etmektedir. Her satır ise 4 farklı section’a ait farklı katsayı değerleridir.

SOS <4x6 double>						
	1	2	3	4	5	6
1	1	1.7434	1	1	-1.3740	0.5431
2	1	0.6902	1	1	-1.0375	0.7062
3	1	0.0852	1	1	-0.7400	0.8610
4	1	-0.1325	1	1	-0.6083	0.9605

Bu katsayıları yukarıda tanımladığımız fark denkleminde a_i ve b_i olarak gösterilen dizileri yerine kullandık. Örnek olarak gösterirsek, denklemde verilen a_0 değişkeni için yapılan tanımlama $SOS(a,4)$ şeklindedir.

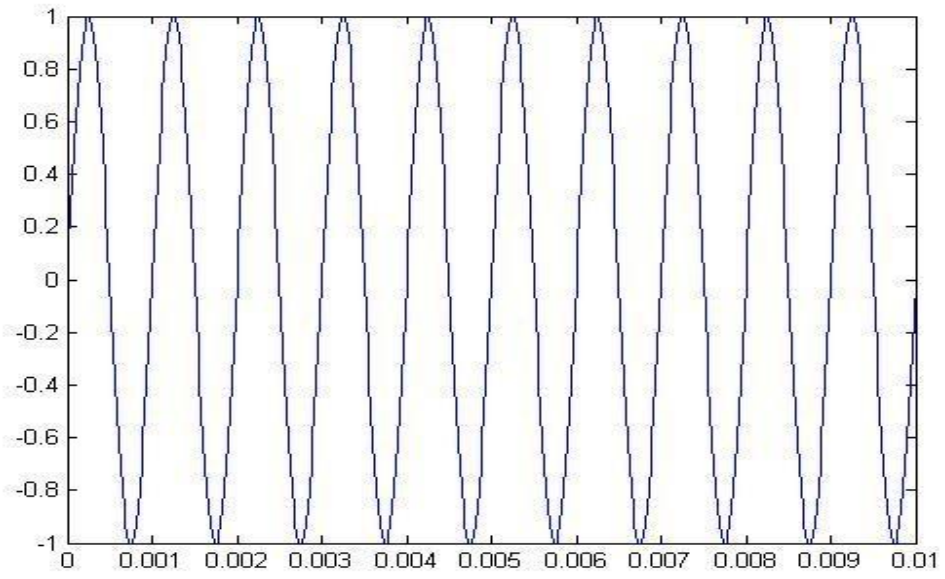
Son olarak toplam için çizilecek fonksiyon, SOS dizisinden gelen scale faktörleri ile çarpılarak grafiği çizilmiştir.

Filtreleme işleminden sonra gürültü büyük ölçüde azaltılmış ancak %100 performans gösterilememiştir. Aşağıda deney ile ilgili ekran alıntıları bulunmaktadır.

Öncelikle farklı frekans değerlerine sahip sinüs grafikleri gösterilmiştir. İlk resimdeki 1kHz frekans değerine sahipken, 2. sinüs grafiği 10 kHz dir. Farklı değerlerde almamızın sebebi toplandığında oluşacak gürültü miktarını arttırmaktır.

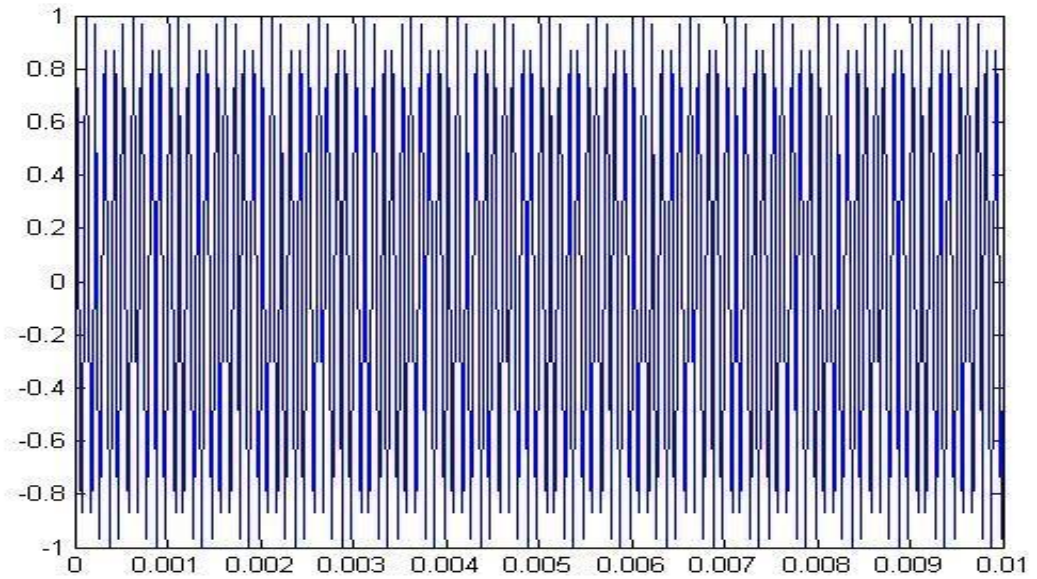
Sinus 1 fonksiyonu:

$f = 1\text{kHz}$

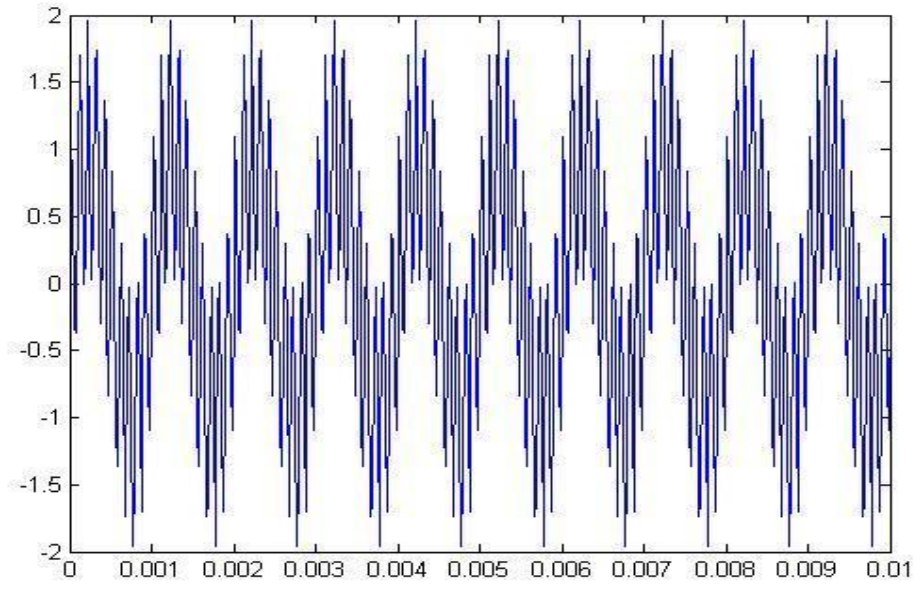


Sinus 2 fonksiyonu:

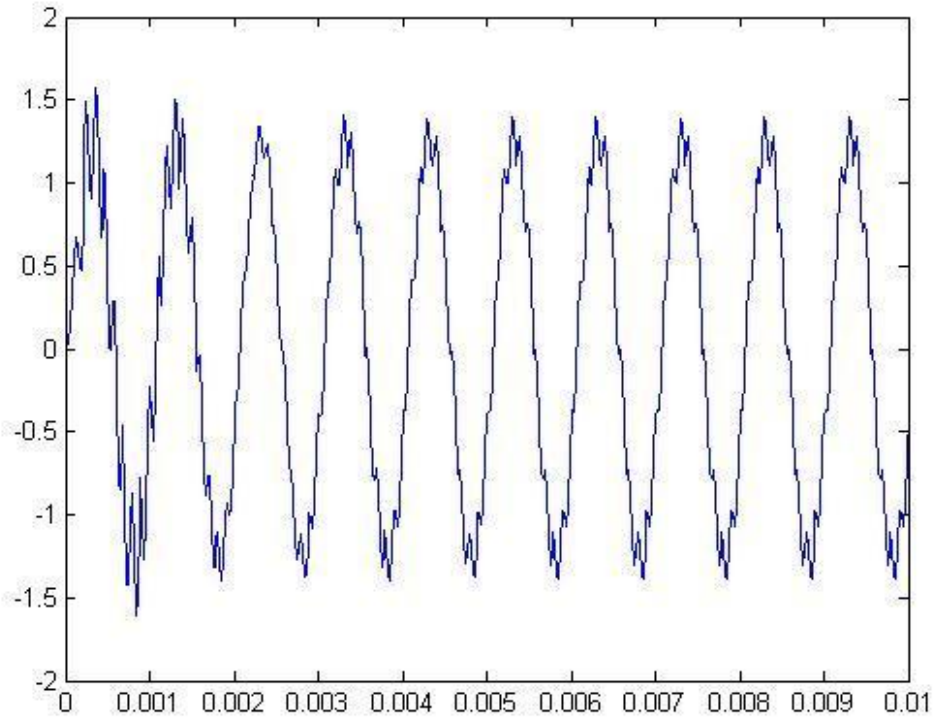
$f = 10\text{kHz}$



İki sinüs grafiğinin toplamı aşağıdaki gibidir.



Filtreden geçirdikten sonra gürültü oldukça azalmıştır, filtre alçak geçiren filtre olduğundan sonuç sinyali 1 kHz lik düşük frekanslı orijinal sinyale benzemiştir.



Deney başarısını görme amaçlı düşük frekanslı sinüs grafiği ve filtreleme sonucu oluşan sinyal bir arada gösterilmiştir. Mavi olan orijinal sinyalken, yeşil olan sonuç sinyalidir. Birebir benzerlik olmasa da gürültü önemli ölçüde azaltılmıştır.

