



KARADENİZ TEKNİK ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

RAPOR

FIR Bant Geçiren Filtre

Ad-Soyad	<u>Uğurcan YILMAZ</u>
Numara	383242
Ders Sorumlusu	<u>Doç.Dr. Sedat GÖRMÜŞ</u>

FIR Bant Geçiren Filtre Oluşturma İşlemi

FIR filtreler, giriş sinyaline sonlu bir zaman dilimi boyunca tepki veren filtrelerdir.

FIR filtreler, dijital sinyallerin işlenmesinde yaygın olarak kullanılan bir filtreleme yöntemidir. Hamming penceresi, FIR filtre tasarımında kullanılan bir pencere fonksiyonudur. Bu yöntem, yüksek bant geçirgenliği ve düşük durdurma bant geçişine sahip FIR filtrelerin tasarımında özellikle kullanışlıdır

```
void FIR_bantgeciren_filtre_tasarla(uint32_t alt_frekans, uint32_t ust_frekans,
                                     uint32_t ornekleme_orani, uint32_t katsayilarin_sayisi,
                                     double *filtre_katsayilari)
{
    double Fn = ornekleme_orani / 2.0;
    double Wn1 = (double)alt_frekans / Fn;
    double Wn2 = (double)ust_frekans / Fn;
    double M = (katsayilarin_sayisi - 1) / 2.0;
    for (uint32_t i = 0; i < katsayilarin_sayisi; i++)
    {
        if (i == (uint32_t)M)
        {
            filtre_katsayilari[i] = (Wn2 - Wn1) / PI;
        } else
        {
            filtre_katsayilari[i] = (sin(Wn2 * PI * (i - M)) - sin(Wn1 * PI * (i - M))) / (PI * (i - M));
        }
        filtre_katsayilari[i] *= 0.54 - 0.46 * cos(2 * PI * i / (katsayilarin_sayisi - 1)); // Hamming window
    }
}
```

Hamming penceresi yöntemi, filtre tasarımında kullanılan birçok farklı pencere fonksiyonundan biridir. Bu pencere fonksiyonu, belirli bir aralıkta seçilen örneklerin, filtrelenmeden önce çarpıldığı bir fonksiyondur. Hamming penceresi, düşük yan lob ve yüksek ana lob genişliği ile karakterizedir. Bu nedenle, FIR filtrelerin tasarımında sıkça kullanılır. Hamming penceresi, aşağıdaki formülle ifade edilebilir:

$$w(n) = 0.54 - 0.46\cos((2\pi n)/(N-1))$$

Burada $w(n)$, Hamming penceresi fonksiyonudur ve N , pencere uzunluğunu temsil eder. Bu formüldeki \cos fonksiyonu, pencerenin ana lob genişliğini kontrol etmek için kullanılır. Hamming penceresi, FIR filtre tasarımında, ideal filtre karakteristiği kullanılarak filtre katsayıları hesaplandıktan sonra, katsayılarla çarpılarak elde edilir.

Konvolüsyon işlemi

Konvolüsyon işlemi, giriş örnekleri ve filtre katsayıları arasında gerçekleştirilen bir çarpma işlemidir. Bu işlem, giriş örnekleri ile filtre katsayıları arasında bir çarpma işlemi gerçekleştirir ve sonuçlarını toplar. Bu toplama işlemi, çarpma işleminin sonucunu temsil eder. Bu işlem, bir sinyalin belirli bir aralığındaki örneklerin filtrelenmesini sağlar ve sonuçta, filtrelenmiş sinyalin örnekleri oluşur.

```
void konvolusyon(double *filtre_katsayilari, uint32_t katsayilarin_sayisi,
                double *giris_ornekleri, double **cikis_ornekleri, uint32_t ornek_sayisi)
{
    *cikis_ornekleri = (double *)malloc(sizeof(double) * ornek_sayisi);
    uint32_t i, j, k;
    double tmp;
    for (k = 0; k < ornek_sayisi; k++)
    {
        tmp = 0;
        for (i = 0; i < katsayilarin_sayisi; i++)
        {
            j = k - i;
            if (j < ornek_sayisi)
            {
                tmp += filtre_katsayilari[i] * giris_ornekleri[j];
            }
        }
        (*cikis_ornekleri)[k] = tmp;
    }
}
```

Giriş örnekleri, işlemin başlangıç noktasını temsil eder. Bu örnekler, genellikle bir ses veya görüntü işleme uygulamasında toplanan verilerdir. Örneklerin sayısı, işlemi gerçekleştirmek için gereken bellek miktarını belirler.

Bu işlem, birçok sinyal işleme uygulamasında kullanılır. Örneğin, bir ses veya görüntü işleme uygulamasında, bir sinyali filtrelemek ve istenmeyen bileşenleri ortadan kaldırmak için kullanılabilir ama ben bu projede sadece ses işlemek için kullandım.

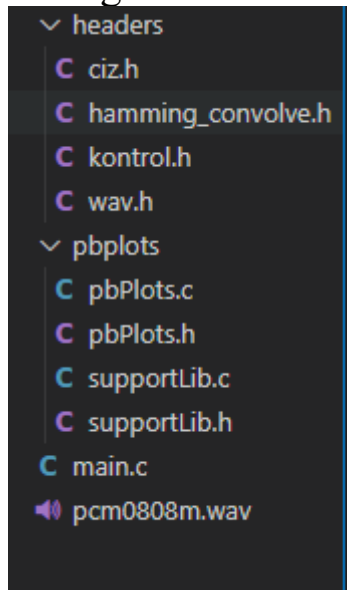
Program Hakkında Bilgiler

Gerekli kütüphaneler ve yazılımlar:

- 1- [pbPlots](#) (C dilinde grafik çizimleri için)

Programın kullanımı:

- 1- Dosyalar aşağıdaki gibi olmalıdır. Üzerinde test yapmak istediğiniz .wav dosyanız aynı konumda olmalıdır.



- 2- C kodunu derleme işlemi için aşağıdaki şekilde komut satırından gcc derleyicisine argüman vermek gerekiyor.

```
gcc main.c pbplots/pbPlots.c pbplots/supportLib.c -o main -lm
```

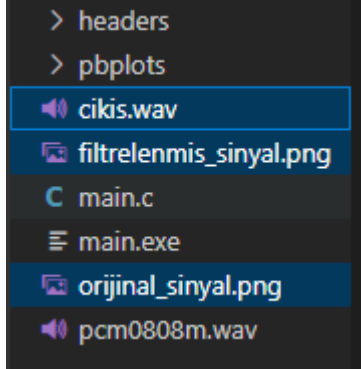
- 3- Daha sonra aşağıdaki şekilde exe dosyasına argüman vererek çalıştırmak gerekiyor.

```
>main pcm0808m.wav 200 500 101
```

(main <giris_dosyasi.wav> <alt_frekans> <ust_frekans> <katsayilarin_sayisi>)

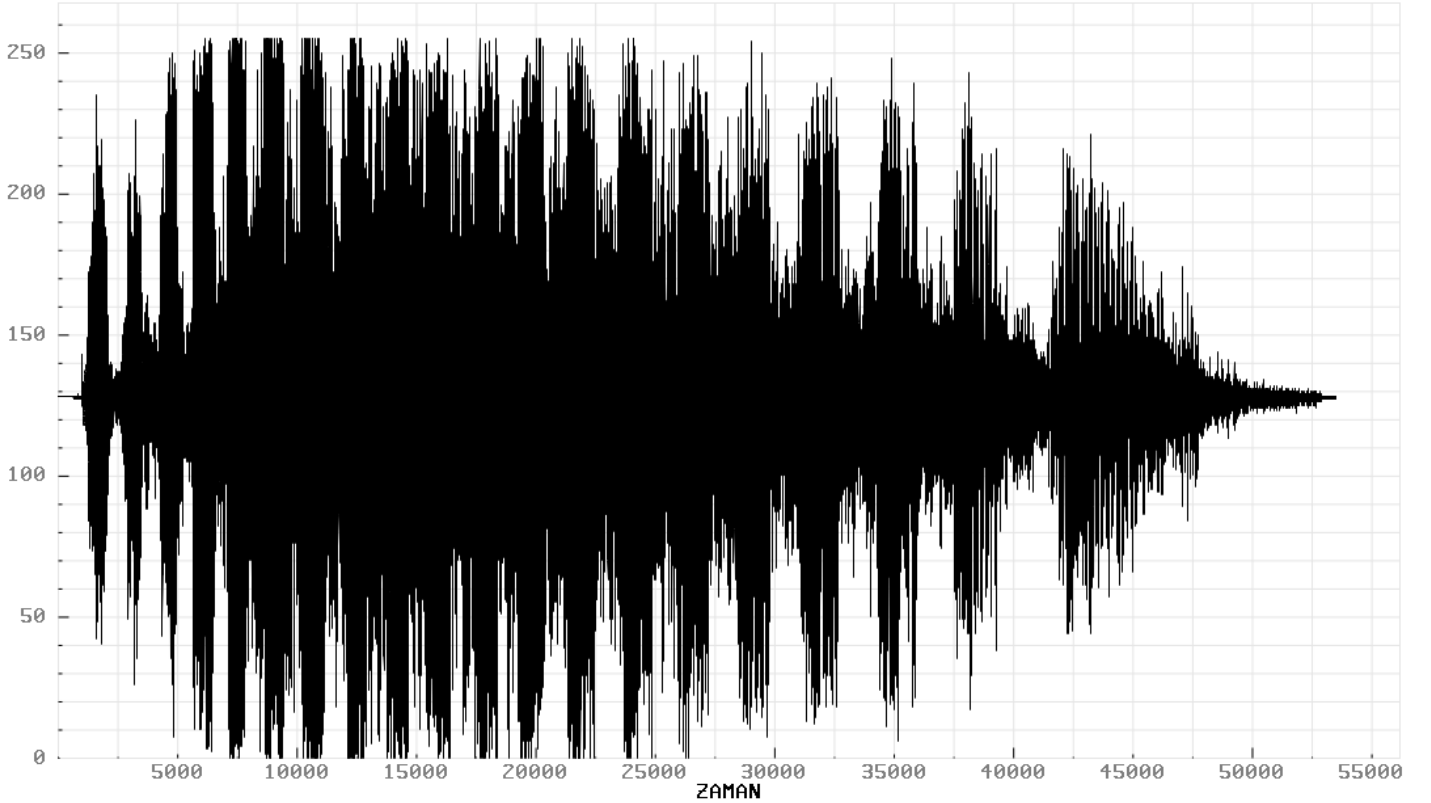
Not: giriş dosyası olarak 8KHz 8 bit wav dosyası kullandım. Daha yüksek frekanslı dosyalarda çalışmayabilir.

- 4- Exe dosyası çalıştıktan sonra dosya dizinine giriş wav dosyasının grafiği, filtrelenmiş wav dosyasının grafiği ve ayrıca filtrelenmiş wav dosyası üretilecektir.



Orijinal Sinyal:

ORIJINAL SINYAL



Filtrelenmiş Sinyal:

FILTRELENMİS SINYAL

