

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



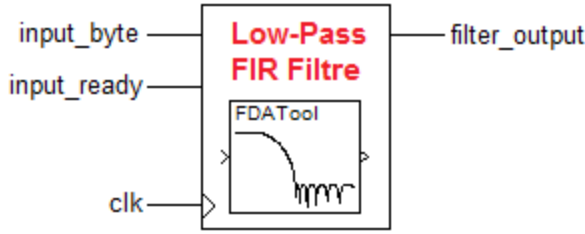
FIR ALÇAK GEÇİREN FİLTRE

HAZIRLAYANLAR

**EMRE AKTÜRK
SERHAN ZEYBEK**

2018-2019 BAHAR DÖNEMİ

Low-Pass Digital FIR Filter



FIR (Finite Impulse Response) filtreler, birim dürtü tepkisi (Unit Impulse Response) sonlu uzunlukta olan filtrelerdir. N. dereceden sayısal bir FIR filtrede birim dürtü tepkisi $h[n]$ de yer alan $N+1$ adet katsayı bulunur. Filtre çıkışı i. katsayı olan b_i ile i birim geciktirilmiş örnek $x[n-i]$ çarpımlarının toplamı olarak ifade edilir.

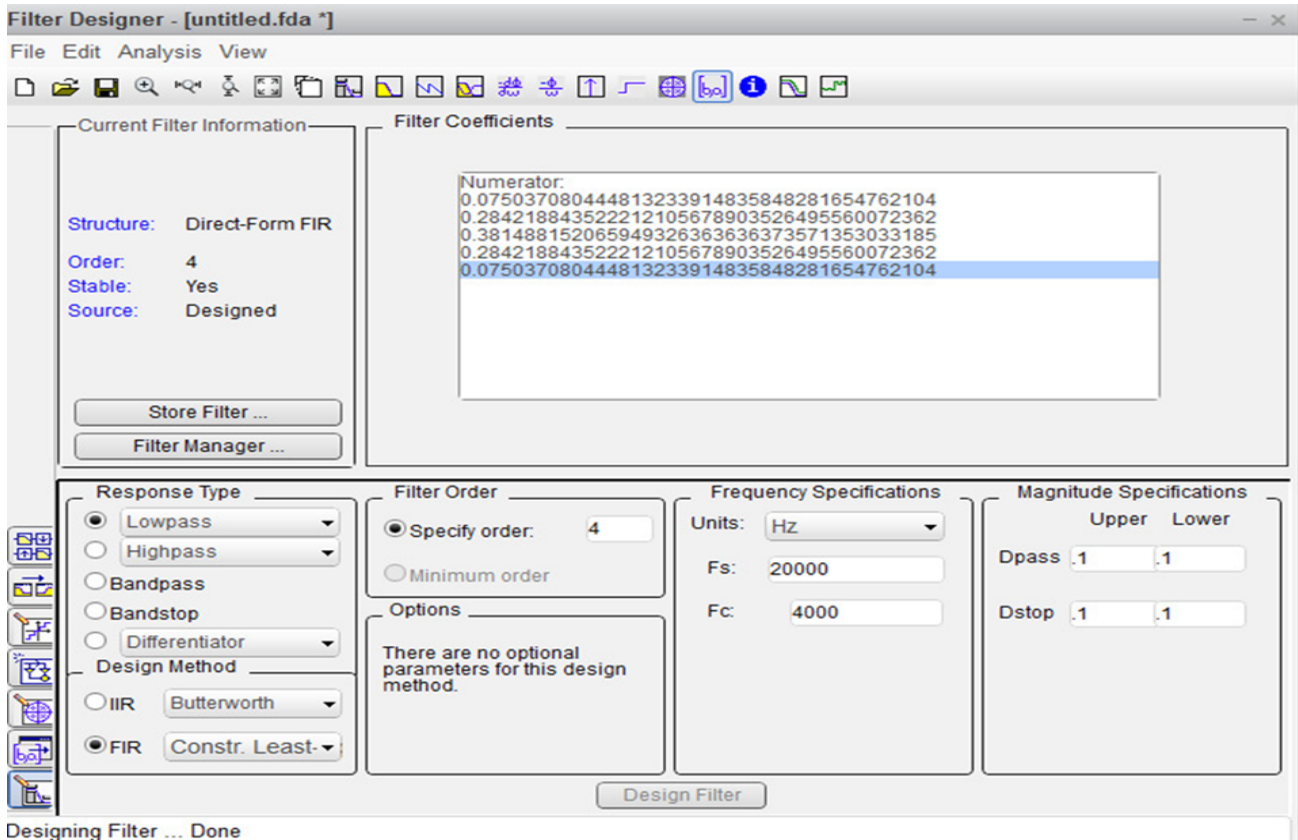
$$y[n] = b_0 x[n] + b_1 x[n-1] + \dots + b_N x[n-N]$$
$$= \sum_{i=0}^N b_i \cdot x[n-i],$$

$h[n]$ FIR filtrenin birim dürtü tepkisi, $x[n]$ herhangi bir giriş sinyali, $y[n]$ filtrenin çıkış sinyali olmak üzere, $y[n]$ convolution yardımıyla şu şekilde hesaplanır:

$$y[n] = x[n] * h[n]$$

$h[n]$ ise, $h[0] = b_0$, $h[1] = b_1$, ..., $h[N] = b_N$ şeklinde b_i katsayılarından oluşur.

Fs: 20 KHz, Fc: 4 KHz için dördüncü dereceden (4th order) yani 5 katsayılı bir Low-pass FIR katsayılarının Matlab fdatool aracı bulunması:



h[n] değerleri:

```
h[0] = 0.075037080444813233914835848281654762104
h[1] = 0.284218843522212105678903526495560072362
h[2] = 0.381488152065949326363636373571353033185
h[3] = 0.284218843522212105678903526495560072362
h[4] = 0.075037080444813233914835848281654762104
```

Parametre olarak verilen genlik ve frekansa sahip sinüsten N adet örnek üreten fonksiyon:

```
// Generate Sinus Wave
double* sinWave(int N, double amp, double freq)
{
    double *data = new double[N];
    for(int i=0; i < N; i++)
        data[i] = amp * sin((2 * M_PI * freq) * ((double)i / SAMPLE_RATE));
    return data;
}
```

Convolution işlemini gerçekleştiren fonksiyon:

```
// The Input Side Algorithm
double* conv(double *s, int s_size, double *h, int h_size)
{
    int y_size = s_size + h_size - 1;
    double *y = new double[y_size];
    memset(y, 0, y_size*sizeof(double));

    for(int i=0; i<s_size; i++)
    {
        for(int j=0; j<h_size; j++){
            y[i+j] += s[i] * h[j];
        }
    }
    return y;
}
```

Low-pass FIR filtrenin test edilmesi:

```
int main()
{
    int N    = 100;
    int amp  = 5;
    int freq = 5000;

    int s_size = 100;
    int h_size = 5;
    int y_size = s_size + h_size - 1 ;

    double *s = sinWave(N, amp, freq);
    //double s[] = {10, 10, 10, 10, 10, 10,10, 10, 10, 10};
    double h[5];
    double *y;

    h[0] = 0.075037080444813233914835848281654762104;
    h[1] = 0.284218843522212105678903526495560072362;
    h[2] = 0.381488152065949326363636373571353033185;
    h[3] = 0.284218843522212105678903526495560072362;
    h[4] = 0.075037080444813233914835848281654762104;

    y = conv(s, s_size, h, h_size);

    for(int i=0; i<100; i++)
        printf("sin[%d] = %f \n",i, s[i]);

    for(int i=0; i<y_size; i++)
        printf("y[%d] = %f \n",i, y[i]);
    return 0;
}
```

Solda 1 KHz frekansa sahip 10 birim genlikli sinüs giriş sinyalinin filtre çıkışı, sağda ise 5 KHz frekansa sahip 10 birim genlikli sinüs giriş sinyalinin filtre çıkışı görülmektedir:

| | |
|--------------------|-------------------|
| y[36] = -9.924437 | y[36] = 0.000000 |
| y[37] = -10.435171 | y[37] = -2.314140 |
| y[38] = -9.924437 | y[38] = -0.000000 |
| y[39] = -8.442230 | y[39] = 2.314140 |
| y[40] = -6.133639 | y[40] = -0.000000 |
| y[41] = -3.224645 | y[41] = -2.314140 |
| y[42] = -0.000000 | y[42] = -0.000000 |
| y[43] = 3.224645 | y[43] = 2.314140 |
| y[44] = 6.133639 | y[44] = -0.000000 |
| y[45] = 8.442230 | y[45] = -2.314140 |
| y[46] = 9.924437 | y[46] = 0.000000 |
| y[47] = 10.435171 | y[47] = 2.314140 |
| y[48] = 9.924437 | y[48] = 0.000000 |
| y[49] = 8.442230 | y[49] = -2.314140 |
| y[50] = 6.133639 | y[50] = -0.000000 |
| y[51] = 3.224645 | y[51] = 2.314140 |
| y[52] = 0.000000 | y[52] = -0.000000 |
| y[53] = -3.224645 | y[53] = -2.314140 |
| y[54] = -6.133639 | y[54] = -0.000000 |
| y[55] = -8.442230 | y[55] = 2.314140 |
| y[56] = -9.924437 | y[56] = -0.000000 |
| y[57] = -10.435171 | y[57] = -2.314140 |
| y[58] = -9.924437 | y[58] = -0.000000 |
| y[59] = -8.442230 | y[59] = 2.314140 |
| y[60] = -6.133639 | y[60] = 0.000000 |