# Design and implementation of finite impulse response ( FIR ) filter.

Objective : FIR filters can provide linear phase response , sharp cutoff and stability at the expense of process time and memory storage .

In this experiment , FIR filter is designed based on window method where ideal impulse response

of the filter hd ( n ) is multiplied by a smooth window function w ( n ) of finite duration to alleviate excess ripples .

\* plot hamming , han and kaiser window functions used in design of a FIR filter .

CODE:

% plot hamming , han and kaiser window functions used in design of a FIR filter .

% window function used in design of a FIR filter

beta = 5.2 ;

N = 20 ;

n = 1 : 1 : 20 ;

y = hamming ( N ) ;

y1 = hann ( N ) ;

y2 = kaiser ( N , beta ) ;

plot ( n , y , 'k^-' , n , y1 , 'kd:' , n , y2 , 'k\*:' ) ;

xlabel ( 'n' )

ylabel ( 'h(n)' )

legend ( 'Hamming' , 'Hann' , 'Kaiser' )

grid on



%Design a bandpass FIR filter with upper and lower cutoff frequency of 275 and 125 (use both Hamming and Kaiser window function). consider sampling frequency of 2kHZ.

Fs = 2000 ; % Sampling frequency

Fn = Fs / 2 ;

N = 100 ; % Number of coefficients

beta = 5.65 ; % beta parameter for kaiser window

fc1 = 125 / Fn ; % normalized cutoff frequency of lower side

fc2 = 275 / Fn ; %normalized cutoff frequency of upper side

Fc = [ fc1 fc2 ] ;

hn = fir1 ( N-1 , Fc , kaiser ( N , beta ) ) ; % FIR coefficients

[ H , f ] = freqz ( hn , 1 , 512 , Fs ) ; % Frequency response

mag = 20 \* log ( abs ( H ) ) %conversion in dB

subplot ( 2 , 1 , 1 )

plot ( f , mag ) , grid on

xlabel ( 'Frequency in Hz' )

ylabel ( 'Magnitude in dB' )

title ( 'Using kaiser window' )

hn = fir1 ( N-1 , Fc , hamming ( N ) ) % Fir coefficient

[ H , f ] = freqz ( hn , 1 , 512 , Fs ) ; % frequency response of Fir

mag = 20 \* log ( abs ( H ) ) ; % conversion in dB

subplot ( 2 , 1 , 2 )

plot ( f , mag ) , grid on

xlabel ( 'Frequency in Hz' )

ylabel ( 'Magnitude in dB' )

title ( 'Using Hamming window' )



# In frequency sampling method, the FIR filter is response by desired frequency response instead of impuslse response. The coefficient of an FIR filter is evaluated as,

h(n) = IDFT{H(k)}; k= 0,1,2,3,...N-1

=

here, H(k) is the desired frequency response (Normalized form) of the filter of N

samples taken at intervals of kFs/N.

# Let us consider a low pass FIR filter of passband : 0-5 kHz , sampling frequency , Fs = 18 kHz and the number of samples, N = 9.

Now, kFs / N = k \* 18 / 9 = 2 kHz. For the passband of 0-5 kH , k = 0 , 1 and 2 . For stopband k = 4 , 5 , 6 , 7 and 8 .

So, H ( k ) = { 1 ; k = 0, 1 , 2

{ 0 , k = 4 , 5 , 6 , 7 , 8

In matlab Syntax of FIR filter based on frequency sampling technique is , hn = fir2 ( N-1 , F , H) ; where H is the vector of desired magnitude response at the corresponding frequency points of the vector F.

The frequency points of the vector F is in normalized form in the range of 0 to 1 .

Fs = 18 ;

N = 9 ;

fts = [ 0 1 2 3 4 5 6 7 ] / 7 ;

Hk = [ 1 1 1 0 0 0 0 0 ] ;

b = fir2 ( N-1 , fts , Hk ) ;

% b = fir2(N,F,A) designs an Nth order FIR digital filter with the frequency response

%specified by vectors F and A and return the filter coefficients in length N+! vector b.

[ h , f ] = freqz ( b , 1 , 512 , Fs ) ;

plot ( f \* ( Fs / N ) , abs ( h ) ) ;

xlabel ( 'Frequency in kHz' )

ylabel ( 'Magnitude' )

grid on

