MATLAB: AN INTRODUCTION II
SIGNAL PROCESSING TOOLBOX

Exercise 2-3.

Function, operators, special characters	Description of function, operators and special characters
b = fir1(N,Wn)	FIR1 FIR filter design using the window method. b = fir1(N,Wn) designs an N'th order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B. The cut-off frequency Wn must be between 0 < Wn < 1.0, with 1.0 corresponding to half the sample rate.
W = BOXCAR(N)	W = BOXCAR(N) returns the N-point rectangular window. See also BARTLETT, BLACKMAN, CHEBWIN, HAMMING, HANN, KAISER and TRIANG.
[h,w] = freqz(b,a,n)	Digital filter frequency response. freqz returns the N-point complex frequency response vector H and the N-point frequency vector W in radians/sample of the filte:
abs(h)	ABS(X) is the absolute value of the elements of X. When X is complex, ABS(X) is the complex modulus (magnitude) of the elements of X.
angle(h)	ANGLE(H) returns the phase angles, in radians, of a matrix with complex elements.
unwrap(x)	Unwrap phase angle. UNWRAP(P) unwraps radian phases P by changing absolute jumps greater than pi to their 2*pi complement. It unwraps along the first non-singleton dimension of P. P can be a scalar, vector, matrix, or N-D array.
[Gd,W] = grpdelay(B,A,N)	Group delay of a digital filter. [Gd,W] = GRPDELAY(B,A,N) returns length N vectors Gd and W containing the group delay and the frequencies (in radians) at which it is evaluated. Group delay is -d{angle(w)}/dw.
filter(b,a,x)	One-dimensional digital filter. Y = FILTER(B,A,X) filters the data in vector X with the filter described by vectors A and B to create the filtered data Y.
roots(x)	Find polynomial roots. ROOTS(C) computes the roots of the polynomial whose coefficients are the elements of the vector C. If C has N+1 components, the polynomial is C(1)*X^N + ... + C(N)*X + C(N+1).
poly(x)	POLY(V), when V is a vector, is a vector whose elements are the coefficients of the polynomial whose roots are the elements of V . For vectors, ROOTS and POLY are inverse functions of each other, up to ordering, scaling, and roundoff error.
polar(theta, rho)	Polar coordinate plot. POLAR(THETA, RHO) makes a plot using polar coordinates of the angle THETA, in radians, versus the radius RHO. POLAR(THETA,RHO,S) uses the linestyle specified in string S. See PLOT for a description of legal line styles.
i, j	Imaginary unit. As the basic imaginary unit SQRT(-1), i and j are used to enter complex numbers.

Function, operators, special characters	Description of function, operators and special characters
pi	3.1415926535897....
conj(x)	Complex conjugate. CONJ(X) is the complex conjugate of X. For a complex X, CONJ(X) = REAL(X) - i*IMAG(X).
exp(x)	EXP(X) is the exponential of the elements of X, e to the X. For complex Z=X+i*Y, EXP(Z) = EXP(X)*(COS(Y)+i*SIN(Y)).
poly(x)	POLY(V), when V is a vector, is a vector whose elements are the coefficients of the polynomial whose roots are the elements of V . For vectors, ROOTS and POLY are inverse functions of each other, up to ordering, scaling, and roundoff error.
max(x)	Largest component. For vectors, MAX(X) is the largest element in X.
min(x)	Smallest component. For vectors, MIN(X) is the smallest element in X.
ginput(N)	Graphical input from mouse. [X,Y] = GINPUT(N) gets N points from the current axes and returns the X- and Y-coordinates in length N vectors X and Y. The cursor can be positioned using a mouse (or by using the Arrow Keys on some systems). data points are entered by pressing a mouse button or any key on the keyboard except carriage return, which terminates the input before N points are entered. [X,Y] = GINPUT gathers an unlimited number of points until the return key is pressed.
gtext('text')	Place text with mouse. GTEXT('string') displays the graph window, puts up a cross-hair, and waits for a mouse button or keyboard key to be pressed. The cross-hair can be positioned with the mouse (or with the arrow keys on some computers). Pressing a mouse button or any key writes the text string onto the graph at the selected location.
B = FIR2(N,F,A)	FIR arbitrary shape filter design using the frequency sampling method. B = FIR2(N,F,A) designs an N'th order FIR digital filter with the frequency response specified by vectors F and A, and returns the filter coefficients in length N+1 vector B. Vectors F and A specify the frequency and magnitude breakpoints for the filter such that PLOT(F,A) would show a plot of the desired frequency response. The frequencies in F must be between 0.0 < F < 1.0, with 1.0 corresponding to half the sample rate. They must be in increasing order and start with 0.0 and end with 1.0.

Tasks

1. Design FIR filters by windowing method (FIR1).
2. Window functions.
3. Frequency response computation. Magnitude response, phase response, group delay of a digital filter.
4. Plot of magnitude response, phase response, group delay of a digital filter.
5. Zero plot of FIR filters computation.
6. Filtering of signals. Example.
7. Design FIR filters by windowing method (FIR2).

Example 1.
Design a low-pass filter with pass-band cut off frequencies $f_1 = 20\,kHz$ of the order $N = 20$. Frequency sampling is $f_S = 80\ kHz$. For the filter design the different kind of the window should be used. Plot the magnitude response, phase response, group delay function and zero-plot of the designed filters. Compare the attenuation in the stop-band and ripple in the pass-band of the designed filters corresponding by the application of the different windows.

Example 2.
Design a low-pass filter with pass-band cut off frequencies $f_1 = 20\,kHz$ of the order $N = 11, 20, 30, 50, 100, 150, 200$. Frequency sampling is $f_S = 80\ kHz$. It is desired to apply rectangular and Bartlett window at the design. Plot the magnitude response, phase response, group delay function and zero plot of the designed filters. Analyze the influence of the length of the impulse response and the applied window function on the ripple of the magnitude response in the pass-band (so-called Gibss phenomenon).

Example 3.
Design a low-pass filter with pass-band cut off frequencies $f_1 = 20\,kHz$ of the order $N = 11, 20, 30, 50, 100$. Frequency sampling is $f_S = 160\,kHz$. It is desired to apply rectangular and Bartlett window at the design.

Example 4.
Design a high-pass filter with pass-band cut off frequencies $f_1 = 20\,kHz$ of the order $N = 11, 20, 30, 50, 100$. Frequency sampling is $f_S = 160\,kHz$. It is desired to apply rectangular and Bartlett window at the design.

Example 5.
Design a band-pass filter with pass-band cut off frequencies $f_1 = 20\,kHz$ and $f_2 = 40\,kHz$ of the order $N = 11, 20, 30, 50, 100$. Frequency sampling is $f_S = 160\,kHz$. It is desired to apply rectangular and Bartlett window at the design.

Example 6.
By the impulse response truncation method (by the windowing method at rectangular window application) design a Hilbert transformer of the order $N = 11, 20, 30, 50, 100$.

Example 7.
By the windowing method at Hann window application design a differentiator of the order $N = 11, 20, 30, 50, 100$.

Example 8.
Design a stop-band filter with pass-band cut off frequencies $f_1 = 20\,kHz$ and $f_2 = 40\,kHz$ of the order $N = 11, 20, 30, 50, 100$. Frequency sampling is $f_S = 160\,kHz$. It is desired to apply rectangular and Bartlett window at the design.