IIR DIGITAL FILTER DESIGN
BY
BILINEAR TRANSFORMATION. MATLAB APPLICATION

Exercise 7.

Function, operators, special characters	Description of function, operators and special characters
[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 filter.
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.
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)).
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.
pi	3.1415926535897....

Function, operators, special characters	Description of function, operators and special characters
[B,A] = butter(N,Wn)	Function designs an Nth order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors B and A.
[N, Wn] = buttord(Wp, Ws, Rp, Rs)	Function returns the order N of the lowest order digital Butterworth filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.
[B,A] = cheby1(N,R,Wn)	Function designs an Nth order lowpass digital Chebyshev filter with R decibels of peak-to-peak ripple in the passband. CHEBY1 returns the filter coefficients in length N+1 vectors B and A.
[N, Wn] = cheb1ord(Wp, Ws, Rp, Rs)	Function returns the order N of the lowest order digital Chebyshev Type I filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.
[B,A] = cheby2(N,R,Wn)	Function designs an Nth order lowpass digital Chebyshev filter with the stopband ripple R decibels down and stopband edge frequency Wn. CHEBY2 returns the filter coefficients in length N+1 vectors B and A.
[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs)	Function returns the order N of the lowest order digital Chebyshev Type II filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.
[B,A] = ellip(N,Rp,Rs,Wn)	Function designs an Nth order lowpass digital elliptic filter with Rp decibels of peak-to-peak ripple and a minimum stopband attenuation of Rs decibels. ELLIP returns the filter coefficients in length N+1 vectors B and A.
[N, Wn] = ellipord(Wp, Ws, Rp, Rs)	Function returns the order N of the lowest order digital elliptic filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.

Tasks

1. Design IIR filters by bilinear transform method (butter, cheby1, cheby2, ellip).
2. Compare frequency response (magnitude response, phase response, group delay) of the different filters of the same order.
3. Compare frequency response (magnitude response, phase response, group delay) of the different filters of the different order.
4. Cascade structures for the implementation of IIR filters (the second order section applications).
5. IIR parameters quantozing.