

**WHITING
SCHOOL OF
ENGINEERING**

JOHNS HOPKINS UNIVERSITY

520.435 Digital Signal Processing

Hassan Mohy-ud-Din

Course Page : http://cyphynets.lums.edu.pk/index.php/ECE_520.435_-_DSP_for_Matlab

My Webpage: http://cyphynets.lums.edu.pk/index.php/Hassan_Mohy-ud-Din

Z Transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$x[n] = \frac{1}{2\pi j} \oint_C X(z)z^{n-1} dz$$

$$= \sum [\text{Residues of } X(z) \cdot z^{n-1} \text{ at the poles inside the contour } C]$$

Pole-Zero Plot

$$H(z) = K \frac{(z - z_1)(z - z_2) \cdots (z - z_n)}{(z - p_1)(z - p_2) \cdots (z - p_n)}$$

Transfer Function
Polynomial
Coefficients

$$H(z) = K \frac{Num(z)}{Den(z)}$$

$$G(z) = \frac{1 - 24z^{-1} + 288z^{-2}}{1 - 08z^{-1} + 064z^{-2}}$$

[Z,P,K]=tf2zpk(Num, Den)

Discrete-time Transfer Function to Zero-Pole Conversion

Plotting in Z-Plane

`zplane(z,p)`

Plots the zeros and poles with the Unit circle as reference
*z-row vector of zeros of $H(z)$ or **roots of Num***
*p-row vectors of columns of $H(z)$ or **roots of Den***

Digital Filter Frequency Response

$$H(e^{j\omega}) = \frac{B(e^{j\omega})}{A(e^{j\omega})} = \frac{b(1) + b(2)e^{-j\omega} + \dots + b(m+1)e^{-jm\omega}}{a(1) + a(2)e^{-j\omega} + \dots + a(n+1)e^{-jn\omega}}$$

$[H, w] = \text{freqz}(\text{Num}, \text{Den}, N)$

H: N-point complex frequency response vector

W: N-point frequency vector, w , in radians/sample

N: resolution of frequency response

Z-Transform Partial Fraction

$$\frac{B(z)}{A(z)} = \frac{r(1)}{1 - p_1 z^{-1}} + \dots + \frac{r(n)}{1 - p_n z^{-1}} + K_1 + K_2 z^{-1} \dots$$

number of poles, $n = \text{length}(\text{Den}) - 1 = \text{length}(R) = \text{length}(P)$

if $\text{length}(\text{Num}) < \text{length}(\text{Den})$, then $K = []$,

$\text{length}(K) = \text{length}(\text{Num}) - \text{length}(\text{Den}) + 1$

if $P(j) = \dots = P(j + m - 1)$ is a pole of algebraic multiplicity, m , then :

$$\frac{R(j)}{1 - p_j z^{-1}} + \frac{R(j+1)}{(1 - p_j z^{-1})^2} + \dots + \frac{R(j+m-1)}{(1 - p_j z^{-1})^m}$$

Residue Computation

Reversible
Function

[R,P,K] = residuez(Num,Den)

R: Residues of partial fraction expansion of $H(z)$

K: Direct Terms of partial fraction expansion of $H(z)$

P: Poles of $H(z)$

Residuez(.) is a discrete equivalent for computing the partial fraction expansion of $H(z)$ and we **do not** need to divide by **z** initially to obtain the coefficients.

Residue(.) is a continuous equivalent for computing the partial fraction expansion of $H(z)$ and we **do** need to divide by **z** initially to obtain the coefficients

Inverse Z-Transform

$$X(z) = \frac{1}{(1 - 0.9z^{-1})^2 (1 + 0.9z^{-1})}, \quad |z| > 0.9$$

Computing the Num and Den and then applying Residue Theorem.

Q. But MATLAB is supposed to reduce work, can I calculate the Den coefficient directly?

A. Yes, you can. Using poly(.)

System Representation in Z-Domain

$$H(z) \triangleq Z[h(n)] = \sum_{n=-\infty}^{\infty} h(n)z^{-n}; \quad R_{h-} < |z| < R_{h+}$$

$$Y(z) = H(z)X(z); \quad ROC_y = ROC_h \cap ROC_x$$

$$y(n) + \sum_{k=1}^N a_k y(n-k) = \sum_{l=0}^M b_l x(n-l)$$

$$H(z) = b_o z^{n-m} \frac{\prod_{l=1}^N (z - z_l)}{\prod_{k=1}^N (z - p_k)}$$

Transfer Function Representation

$$H(e^{j\omega}) = b_o e^{j(N-M)\omega} \frac{\prod_{l=1}^M (e^{j\omega} - z_l)}{\prod_{k=1}^N (e^{j\omega} - p_k)}$$

$$|H(e^{j\omega})| = |b_o| \frac{|e^{j\omega} - z_1| \dots |e^{j\omega} - z_M|}{|e^{j\omega} - p_1| \dots |e^{j\omega} - p_N|}$$

$$\angle H(e^{j\omega}) = \underbrace{[0 \text{ or } \pi]}_{\text{constant}} + \underbrace{[(N-M)\omega]}_{\text{linear}} + \underbrace{\sum_{k=1}^M \angle(e^{j\omega} - z_k) - \sum_{k=1}^N \angle(e^{j\omega} - p_k)}_{\text{non-linear}}$$

Solutions to the Difference Equation

One-sided Z-Transform of a sequence $\mathbf{x}[n]$ is:

$$Z^+[x(n)] \triangleq Z[x(n)u(n)] \triangleq X^+[z] = \sum_{n=0}^{\infty} x(n)z^{-n}$$

$$Z^+[x(n-k)] = x(-1)z^{1-k} + x(-2)z^{2-k} + \dots + x(-k) + z^{-k}X^+(z)$$

The above equation can be used to solve the difference equation :

$$1 + \sum_{k=1}^N a_k y(n-k) = \sum_{m=0}^M b_m x(n-m), \quad n \geq 0$$

subject to the initial conditions :

$$\{y(i), i = -1, \dots, -N\} \text{ and } \{x(i), i = -1, \dots, -M\}$$

filter(b, a, x, x_initial_conditions)

