Basic DSP Using Matlab

Ching-Hua Lee
11/12/2015
In HW 5 You Will Be Asked to:

- find the DFT and inverse DFT of a discrete time signal.
- find the poles, zeros, the impulse response, etc., of a transfer function.
- find the output of a sinusoidal signal to an LTI system.

※ HW 5 is already on the webpage and is due next Tuesday (11/17).
I. DFT and Inverse DFT

- Matlab implements the computation of the DFT using a much faster algorithm called the “fast Fourier transform (FFT).”

- Use the command: “fft()” to obtain the DFT coefficients.
  
syntax: \( X = \text{fft}(x,N) \);
  
  \( x \): input signal  
  \( X \): DFT coefficients  
  \( N \): N-point DFT

- Use the command: “ifft()” to perform the inverse DFT.
  
syntax: \( x1 = \text{ifft}(X) \);
  
  \( x1 \): inverse DFT coefficients
Example Code 1

\[ x = [1,0,0,0] \]
\[ X = \text{fft}(x,2) \]
\[ x1 = \text{ifft}(X) \]
II. Transfer Function Analysis

\[ H(z) = \frac{2z^2 - z - 1}{z^2 - 5z + 6} \]

- Use \texttt{"tf()"} to create the transfer function:
  
syntax: \texttt{num = [2,-1,-1];}
  \texttt{den = [1,-5,6];}
  \texttt{H = tf(num,den, [],'variable','z');}

- Use \texttt{"pzplot()"} to create pole-zero plot:
  syntax: \texttt{pzplot(H)}
II. Transfer Function Analysis (cont.)

- Use “tf2zp” to factorize the transfer function:
  
  syntax:  \[ [z, p, k] = \text{tf2zp}(\text{num}, \text{den}); \]

  \[ z \text{ – zeros.} \]
  \[ p \text{ – poles.} \]
  \[ k \text{ – gain.} \]

  \[ H(z) = k \frac{(z - z_1)(z - z_2)\ldots(z - z_m)}{(z - p_1)(z - p_2)\ldots(z - p_n)} \]

  \[ z = [1, -0.5] \]
  \[ p = [3, 2] \]
  \[ k = 2 \]

  \[ H(z) = 2 \frac{(z - 1)(z + 0.5)}{(z - 2)(z - 3)} \]
II. Transfer Function Analysis (cont.)

- Use “residuez()” to factorize the transfer function:
  
syntax: \([r,p,d] = \text{residuez}(\text{num},\text{den});\)

- \(r\) – residues.
- \(p\) – poles.
- \(d\) – direct terms.

\[
H(z) = \frac{r_1}{1 - p_1 z^{-1}} + \frac{r_2}{1 - p_2 z^{-1}} + \ldots + \frac{r_n}{1 - p_n z^{-1}} + d_1 + d_2 z^{-1} + \ldots + d_{m-n+1} z^{-(m-n)}
\]

- \(z=[4.6667, -2.5]\)
- \(p=[3, 2]\)
- \(d=[-0.1667]\)

\[
H(z) = \frac{4.6667}{1 - 3 z^{-1}} + \frac{-2.5}{1 - 2 z^{-1}} + (-0.1667)
\]
II. Transfer Function Analysis (cont.)

- Use "impz()" to find the **causal** impulse response:
  
  syntax: `impz(num, den, nsamp)`

  *nsamp*: number of samples
Example Code 2

\[ \text{num} = [2,-1,-1]; \]
\[ \text{den} = [1,-5,6]; \]
\[ \text{H} = \text{tf}(\text{num},\text{den},[],'variable','z'); \]
\[ \text{figure} \]
\[ \text{pzplot}(\text{H}) \]
\[ [\text{z},\text{p},\text{k}] = \text{tf2zpk}(\text{num},\text{den}) \]
\[ [\text{r},\text{p},\text{d}] = \text{residuez}(\text{num},\text{den}) \]
\[ \text{figure} \]
\[ \text{impz}(\text{num},\text{den},10) \]
III. LTI System Output of Sinusoidal Signal

- **Input:** $x[n] = \cos\left(\frac{\pi n}{4}\right)$

- **Transfer function:** $H(z) = \frac{1}{1 + z^{-1} + 0.25z^{-2}}$

- **Output:** $y[n] = ?$

- **Use** “filter()” **to compute the output signal** $y[n]$: syntax: $y = \text{filter}(\text{num}, \text{den}, x)$;
Example Code 3

```matlab
n = 1:100;
x = cos(pi/4*n);
figure
stem(n,x)
num = [1];
den = [1,1,0.25];
y = filter(num,den,x);
figure
stem(n,y)
```