Digital Signal Processing
Lab 4: Transfer Functions in the Z-domain

A very important category of LTI systems is described by difference equations of the following type

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

From which, through Z-transform we obtain

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \quad \text{and} \quad Y(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} X(z)$$

where *H(z)* is the transfer function of the system.

In Matlab notation, as indexes must start from 1, if we consider the vectors a and b of the coefficients of the polynomials at numerator and denominator, after posing nb=length(b), na=length(a), we will have a representation of *H(z)* according to

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(nb)z^{-(nb-1)}}{a(1) + a(2)z^{-1} + \ldots + a(na)z^{-(na-1)}} X(z)$$

The transfer function *H(z)* is represented by means of the vectors a and b in several Matlab functions, as described in the following.

<u>zplane</u>

The function zplane creates a plot of the positions of zeros and poles in the plane of the complex variable z, with the unit circle for reference, starting from the coefficients a and b.

Each zero is represented with a 'o' and each pole with a 'x' on the plot. Multiple zeros and poles are indicated by the multiplicity number shown to the upper right of the zero or pole.

The function is called as: zplane(b,a)

where b and a are row vectors. It uses the function roots to calculate the roots of numerator and denominator of the transfer function.

<u>Example-1</u>: $H(z) = \frac{2+2z^{-1}+z^{-2}}{1-0.8z^{-1}}$

```
b=[2 2 1];
a=[1 -0.8];
zplane(b,a);
```

impz

The function impz computes the impulse response of a system starting from the coefficients b and a.

[h,t]=impz(b,a) produces the impulse response in vector h and the time axis in vector t.

If the output arguments h and t are omitted, a plot of the impulse response is directly displayed.

If the impulse response is of infinite length, only its initial part is computed.

Example 2: $H(z) = \frac{1}{1-0.9z^{-1}}$ (system with a pole in z=0.9)

```
b=1;
a=[1 -0.9];
[h,t]=impz(b,a);
stem(t,h);
```

freqz

The function freqz is used to compute the frequency response of systems expressed by difference equations or rational transfer functions.

[H,w]=freqz(b,a,N);

where N is a positive integer, returns the frequency response H and the vector w with the N angular frequencies at which H has been calculated (i.e. N equispaced points on the unit circle, between 0 and π). If N is omitted, a default value of 512 is assumed. If no output argument is specified, the amplitude plot and the phase plot of the frequency response are directly displayed.

[H,w]=freqz(b,a,w);

where w is a vector of frequencies (in radians, e.g. w=-pi:1/100:pi;) computes the frequency response at the frequencies specified by w. This function can be used to evaluate the DTFT of a sequence x on any desired set of frequencies w, e.g. with the command [X,w]=freqz(x,1,w);

See help freqz for a complete reference.

Example 3: $H(z) = \frac{1}{1-0.9z^{-1}}$, i.e. a system with exponentially decaying impulse response $h[n] = (0.9)^n u[n]$

```
b=1;
a=[1 -0.9];
[H,w]=freqz(b,a);
subplot(211)
plot(w/pi, 20*log10(abs(H))); % amplitude plot in decibel
```

```
xlabel('frequency in \pi units'); ylabel('Magnitude in dB');
title('Magnitude Response')
subplot(212)
plot(w/pi, angle(H)/pi);      % phase plot
xlabel('frequency in \pi units'); ylabel('Phase in radians/\pi');
title('Phase Response')
```

Example 4: h[n]=u[n]-u[n-L]

```
L=11;
b=ones(1,L);
w=-pi:1/100:pi;
[H,w]=freqz(b,1,w);
subplot(211)
plot(w/pi,abs(H));         % amplitude plot
subplot(212)
plot(w/pi, angle(H)/pi);      % phase plot
```

Note that, if L is odd, h[n+(L-1)/2] is a real and symmetric sequence, therefore...

```
subplot(311)
plot(w/pi,real(H.*exp(j*w*(L-1)/2)))    % it is actually real
subplot(312)
plot(w/pi,imag(H.*exp(j*w*(L-1)/2)))    % should be zero but...
subplot(313)
plot(w/pi, angle(H.*exp(j*w*(L-1)/2))/pi); % phase plot
```

Example 5: $H(z) = \frac{1-0.9^8 z^{-8}}{1-0.9z^{-1}}$, i.e. a system with truncated exponentially decaying impulse response $h[n]=0.9^n$ (u[n]-u[n-8])

```
b=[1 0 0 0 0 0 0 0 -(0.9)^8];
a=[1 -0.9];
[H,w]=freqz(b,a);
subplot(211)
plot(w/pi, 20*log10(abs(H))); % amplitude plot in decibel
subplot(212)
plot(w/pi, angle(H)/pi);      % phase plot
```

Check also: impz(b,a); zplane(b,a);

filter

The function filter implements the filtering of an input sequence x, starting from a transfer function H(z) expressed as ratio between polynomials in z^{-1} with coefficients given by vectors b (numerator) and a (denominator).

The filter is applied to an input sequence x with the Matlab command
y=filter(b,a,x);

Example 6: $H(z) = \frac{1}{1-0.9z^{-1}}$, (corresponding to h[n]= $(0.9)^n$ u[n])

```
x=[1 zeros(1,100)];   % represents a delta pulse
b=1;
a=[1 -0.9];
y=filter(b,a,x);
stem(y);    % we get the impulse response
```

tf2zp

The command [z,p,k]=tf2zp(b,a) finds zeros, poles and gain of the transfer function associated to coefficients b and a

zp2tf

The command [b,a]=zp2tf(z,p,k)
finds the coefficients b and a of the associated transfer function, given a set of zero locations in vector z, a set of pole locations in vector p, and a gain in scalar k.

Example 7: Design a low pass filter using pole-zero placement and then:

- Convert the pole-zero representation to a rational transfer function representation
- Make a plot of the desired magnitude and phase response

```
p=[0.5;0.45+0.5i;0.45-0.5i];  % poles
z=[-1;i;-i];                  % zeros

zplane(z,p)     % check positions in the plane of z
pause
k=1;
[num, den]=zp2tf(z,p,k);
[H,w]=freqz(num,den);
subplot(2,1,1)
```

```
plot(w/pi,abs(H));
title('\midH(e^j^\omega)\mid');
subplot(2,1,2);
plot(w/pi,angle(H));
title('arg(H(e^j^\omega))')
```

Example 8: Design of a simple all-pass filter.

```
% Define a simple pair of conjugate poles:
z0 = 0.9*exp(j*0.1*pi); z1 = z0';
% poly() gives the polynomial coefficients for these roots
a = poly([z0 z1])
%a = 1.0000   -1.7119    0.8100
% For an allpass filter, numerator coefficients are simply the reverse
b = fliplr(a);

zplane(b,a);
pause

% Look at the response:
[H,w]=freqz(b,a);
subplot(211)
plot(w/pi,abs(H));
title('\midH(e^j^\omega)\mid');
subplot(212);
plot(w/pi,angle(H));
title('arg(H(e^j^\omega))')
```

From the properties of the Z-transform we know that the time-domain convolution operation corresponds to a multiplication between the transforms in the Z domain.

Example 9: If we have two polynomials $X_1(z) = 2 + 3z^{-1} + 4z^{-2}$ and $X_2(z) = 3 + 4z^{-1} + 5z^{-2} + 6z^{-3}$, their product $X_3(z) = X_1(z)X_2(z)$ can be obtained by means of the convolution of the sequences corresponding to the inverse Z-transforms, i.e. $x_1=\{\underline{2}, 3, 4\}$ and $x_2=\{\underline{3}, 4, 5, 6\}$:

```
x1 = [2,3,4]; x2 = [3,4,5,6];
x3 = conv(x1,x2)

>> x3 = 6 17 34 43 38 24
```

Hence $X_3(z) = 6 + 17z^{-1} + 34z^{-2} + 43z^{-3} + 38z^{-4} + 24z^{-5}$

Partial fraction expansion

Given a rational transfer function

$$X(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \frac{B(z)}{A(z)}$$

X(z) can be expressed, by means of partial fraction expansion, as

$$X(z) = \underbrace{\sum_{k=0}^{M-N} C_k z^{-k}}_{if\ M \geq N} + \underbrace{\sum_{k=1}^{N} \frac{A_k}{\left(1 - p_k z^{-1}\right)}}_{if\ \text{all the poles are distinct}}$$

We can derive such a partial fraction expansion by means of the Matlab function residuez. [A,p,C]=residuez(b,a) computes the constants on the numerator (A_k, known as also residues), poles (p_k), and direct terms (C_k) of *X(z)*.
The returned column vector A contains the residues, column vector p contains the pole locations, and row vector C contains the polynomials terms when M≥N.

Example 10: Derive analytically and plot the impulse response of the system with transfer function

$$H(z) = \frac{z^{-1}}{3 - 4z^{-1} + z^{-2}} = \frac{0 + z^{-1}}{3 - 4z^{-1} + z^{-2}} \qquad |z|>1$$

```
b = [0,1]; a = [3,-4,1]; [A,p,C] = residuez(b,a)

A =
    0.5000
   -0.5000
p =
    1.0000
    0.3333
C =
     []
```

which leads to the following partial fraction expansion:

$$H(z) = \frac{\frac{1}{2}}{1 - z^{-1}} - \frac{\frac{1}{2}}{1 - \frac{1}{3}z^{-1}}$$

The corresponding impulse response is $h[n] = 0.5u[n] - 0.5(1/3)^n u[n]$

```
n=0:30;
h=A(1)*p(1).^n + A(2)*p(2).^n;
stem(n,h);        % compare with impz(b,a)
```

Example 11: The z-transform of a signal x[n] is given as $X(z) = \frac{1 - 0.64z^{-2}}{1 - 0.2z^{-1} - 0.08z^{-2}}$. Determine the partial fraction expansion.

```
b = [1, 0, -0.64]; a = [1, -0.2, -0.08]; [A,p,C] = residuez(b,a)

A =
-2
-5
p =
 0.4
-0.2
C =
8
```

which leads to the following partial fraction expansion: $X(z) = 8 + \frac{-2}{1 - 0.4z^{-1}} + \frac{-5}{1 + 0.2z^{-1}}$. Derive analytically x[n] considering the possible ROCs.

Additional Exercises

Exercise 12. Given a causal system specified by the difference equation

$$y[n] = x[n-1] - 1.2x[n-2] + x[n-3] + 1.3y[n-1] - 1.04y[n-2] + 0.222y[n-3]$$

- plot the frequency response (amplitude and phase) using freqz
- check zero and pole positions using zplane
- determine zeros and poles using roots, or using [z,p,k]=tf2zp(b,a)
- plot the impulse response using impz

Exercise 13. Given the difference equation

$$y[n] = -0.4y[n-1] + 0.12y[n-2] + x[n] + 2x[n-1]$$

- compute:
  - the corresponding transfer function H(z)
  - the partial-fraction expansion of H(z), using [r,p,k]=residuez(b,a)
  - the consequent analytic expression of the impulse response h[n]
- compare the plot of the computed h[n] with the one produced by means of impz
- compare also with the sequence y produced by x=[1 zeros(1,N)]; y=filter(b,a,x);

Exercise 14. Compute and plot the frequency response of

$$H(z) = \frac{0.15(1 - z^{-2})}{1 - 0.5z^{-1} + 0.7z^{-2}}$$

for $0 \le \omega \le \pi$.

What type of filter does it represent? Check what changes if instead we use

$$H_1(z) = \frac{0.15(1 - z^{-2})}{0.7 - 0.5z^{-1} + z^{-2}}$$

Make use of freqz, zplane, impz

The following script shows an animation based on the geometrical representation of the effect of poles and zeros of a given transfer function into magnitude and phase responses, as a function of the frequency. The number and positions of zeros and poles can be easily changed to investigate different configuration. The parameter of the function pause can be modified to adjust the speed of the animation

```
% Zeros/poles and Transfer Function animation
% script adapted from code by prof. Dan Ellis, Columbia University, USA

% Define the poles/zeros
% Two-pole, two-zero example
zz = [0.8*exp(j*pi*0.3) 0.8*exp(j*pi*-0.3)].';
pp = [0.9*exp(j*pi*0.3) 0.9*exp(j*pi*-0.3)].';

ymax = 2;

bb = poly(zz);
aa = poly(pp);

% Steps around the top half of the unit circle
ww = [0:200]/200*pi;
```

```
% Lay out the display
subplot(121);
zplane(pp,zz);

% fixed axes for magnitude plot
subplot(222)
fax = [0 1 0 ymax];
axis(fax)
grid

% fixed axes for phase plot (pi-normalized)
subplot(224)
pax = [0 1 -1 1];
axis(pax);
grid


GG = polyval(bb,exp(j*ww))./polyval(aa,exp(j*ww));
HH = abs(GG);
PP = angle(GG);

% plot the frames
for i = 1:length(ww);

  w = ww(i);
  z = exp(j*w);

  % Evaluate the z transform at this point
  Gz = polyval(bb,z)./polyval(aa,z);

  HH(i) = abs(Gz);
  PP(i) = angle(Gz);

  % Make the plots
  subplot(121)
  zplane([0],[]);
  hold on
  plot(real(z),imag(z),'sg');
```

```
  % Add omega parameters
  for www = -0.8:0.2:0.8
    ejw = exp(j*www*pi);
    ll = sprintf('%.1f\\pi',www);
    text(real(ejw), imag(ejw), ll);
  end

  % Plot and connect to all the poles and zeros
  for r = pp.'
        plot(real(r),imag(r), 'xr');
        plot([real(r), real(z)], [imag(r), imag(z)], '-r');
  end
  for r = zz.'
        plot(real(r),imag(r), 'ob');
        plot([real(r), real(z)], [imag(r), imag(z)], '-g');
  end
  hold off

  subplot(222)
  plot(ww/pi, HH, w/pi, HH(i), 'sg',[w/pi w/pi], [0 HH(i)], '-g');
  axis(fax)
  grid
  title('magnitude');

  subplot(224)
  plot(ww/pi, PP/pi, w/pi, PP(i)/pi, 'sg');
  axis(pax)
  grid
  title('phase');
  xlabel('\omega / \pi');
  set(gca, 'YTick', [-1 -0.5 0 0.5 1])
  set(gca, 'YTickLabel', [' -pi ';'-pi/2';'  0  ';' pi/2'; '  pi '])

  pause(0.1);  % adjust the speed
end
```