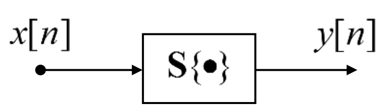
LABORATORY 2
DISCRETE-TIME SYSTEMS

2.1. Discrete-Time Systems

A discrete-time system converts the input sequence $x[n]$ into an output sequence $y[n]$. Denoting by $\mathbf{S}\{\bullet\}$ the system operator, it is described mathematically by the relationship:

$$y[n] = \mathbf{S}\{x[n]\} \tag{2.10}$$

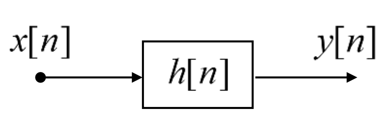
The symbolic representation of a discrete-time system is as follows:



For a linear and time invariant discrete-time system (LTIS), the *weighting function* or *unit impulse response* is defined as the sequence obtained at the output of the system if the input was the impulse unit $\delta[n]$:

$$h[n] = \mathbf{S}\{\delta[n]\} \tag{2.11}$$

Therefore, another symbolic representation of a LTI discrete-time system is:



2. Discrete-Time Systems

The response $y[n]$ of a LTIS for any sequence $x[n]$ can be determined by convolution if the unit impulse response $h[n]$ is known:

$$y[n] = x[n] * h[n] = \sum_k x[k]h[n-k] \tag{2.12}$$

The LTIS can be represented by the *difference equation* with constant coefficients, giving the relation between the input sequence and the output:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{2.13}$$

For the frequency analysis of LTIS, we need the formula for the **Z** transform of a discrete time signal $x[n]$:

$$X(z) = \sum_n x[n] z^{-n} \tag{2.14}$$

For a LTIS, the *system function* or *transfer function* is defined as the ratio between **Z** transforms of the output sequence $y[n]$ and input sequence $x[n]$, and it actually represents the **Z** transform of the impulse response function $h[n]$:

$$H(z) = \frac{Z\{y[n]\}}{Z\{x[n]\}} = \frac{Y(z)}{X(z)} = Z\{h[n]\} \tag{2.15}$$

From the difference equation formula, using the delay property of the **Z** transform ($\mathbf{Z}\{x[n-k]\} = z^{-k} X(z)$), the transfer function $H(z)$ can be expressed as:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{2.16}$$

where we considered $a_0 = 1$.

The roots of the polynomial at the numerator are called *zeros* of the transfer function and the roots of the denominator polynomial are called *poles* of the transfer function.
For a causal system to be stable, the poles must be located within the circle of unity radius (their modulus needs to be below one).

2.2. The impulse response of a *LTIS*

The MATLAB function `impz` allows computing and displaying the impulse response (the weight function) $h[n]$ of a LTIS if we know the coefficients a_k and b_k from the finite difference equation or from the expression of the transfer function $H(z)$.

***Syntax*:**

```
[h,t] = impz(b,a) % for more information type help impz
```

- The vector `b` contains the coefficients b_k (`b` = [b_0, b_1, ..., b_M]) and the vector `a` contains the coefficients a_k (`a` = [1, a_1, a_2, ..., a_N]); it returns a *column* vector `h` that contains the values of the impulse response of the system, $h[n]$, and a *column* vector `t` that contains the moments on the time axis (abscissa), implicitly chosen, in which the previous values were computed; *the output argument* `t` *may be missing from the syntax in the case when we are only interested in the impulse response* `h`.

```
[h,t] = impz(b,a,n)
```

- We state the number `n` of points in which we wish to compute the impulse response `h`; the column vector `t` will contain the values of these points (0, 1, 2, ..., n-1); *the output argument* `t` *may be missing from the syntax in the case when we are only interested in the impulse response* `h`.

```
[h,t] = impz(b,a,n,Fs)
```

- Same as the previous syntax, except that the `n` values from the vector `t` will be distanced with the step 1/Fs (0, 1/Fs, 2/Fs, ..., (n-1)/Fs).

```
[h,t] = impz(b,a,[],Fs)
```

- Same as the previous syntax, except that the number of points in which the impulse response `h` is computed will be chosen implicitly.

```
impz(b,a,...)
```

- Graphical representation of the computed impulse response; any combination of the input parameters from the previous syntaxes may be used.

***Examples*:**
Determine and plot the impulse response of a LTIS functions defined by:

1. $y[n]-0{,}9y[n-1]=0{,}3x[n]+0{,}6x[n-1]+0{,}6x[n-2]$
2. $H(z)=\dfrac{1+0{,}5z^{-1}}{1-1{,}8\cos(\pi/16)z^{-1}+0{,}81z^{-2}}$

```
b=[0.3,0.6,0.6];
a=[1,-0.9];
```

// the vectors `b` and `a` were defined; they contain the values of the coefficients b_k and a_k from the finite difference equation.

```
[h,t]=impz(b,a);
```

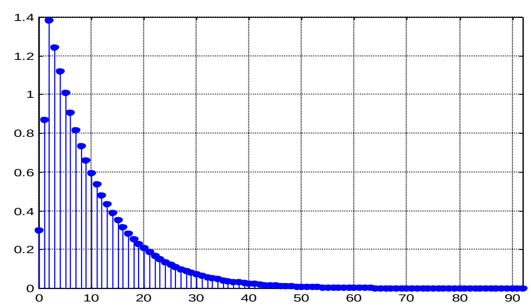
// the impulse response of the system was calculated.

```
size(h)    →    ans = 93    1

size(t)    →    ans = 93    1
```

// we verify the fact that `h` and `t` are column vectors (see the syntax) with 93 elements each.
It can be verified, by typing in the command line `t` and *enter*, that its values are 0, 1, …, 92.
For the graphical representation there are two possibilities:

```
impz(b,a),grid
```




```
stem(t,h),grid
```

// the same graph results.

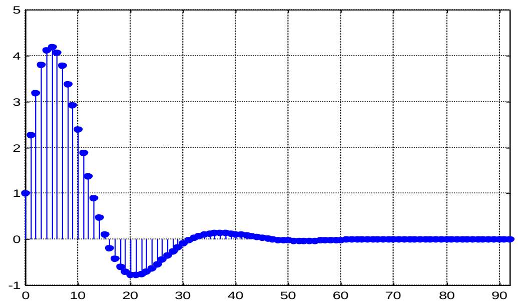
We proceed in a similar way for the second example:

```
b=[1,0.5];
```

```
a=[1,-1.8*cos(pi/16),0.81];
h=impz(b,a);
```

// the column vector h will contain the values of the samples of the impulse response of the system defined by the transfer function $H(z)$ from the example.

```
impz(b,a),grid
```



2.3. The response of a *LTIS* to an input signal

The MATLAB function `filter` allows to determine the response $y[n]$ of a *LTIS* if we know the coefficients a_k and b_k from the finite difference equation or from the expression of the transfer function $H(z)$ and the input signal $x[n]$.

***Syntax*:**

`y = filter(b,a,x)``% for more information type` **`help filter`**

- <u>if `x` is a vector,</u> then it returns a vector `y` of the same dimension with the vector `x`; the vector `x` contains the values of the input signal of the filter (the excitation); the vector `b` contains the coefficients b_k (`b` = [$b_0, b_1, ..., b_M$]) and the vector `a` contains the coefficients a_k (`a` = [1, $a_1, a_2, ..., a_N$]); *if the first element of the vector* `a` *is different than* 1, *then the function* `filter` *normalizes the coefficients* a_k *of the system to the value of the first element of the vector* `a` (*in this way the first element of* `a` *becomes* 1); the obtained vector `y` represents the response of the system defined by the coefficients from the vectors `b` and `a`, if at the input we apply the sequence defined by the vector `x`.
- <u>if `x` is a matrix,</u> then it returns a matrix `y` of the same dimension with the matrix `x`; the function `filter` will in this case operate on columns: column *k* from matrix `y` represents the response of the system defined by the coefficients from the vectors `b` and `a`, if at the input we apply column *k* from matrix `x`.

***Examples*:**

Determine and plot the response of a system defined by:

1. $y[n]-0{,}9y[n-1]=0{,}3x[n]+0{,}6x[n-1]+0{,}6x[n-2]$

2. $H(z)=\dfrac{1+0{,}5z^{-1}}{1-1{,}8\cos(\pi/16)z^{-1}+0{,}81z^{-2}}$

to the input signal $x_1[n]=u[n]-u[n-10]$, for $0\le n\le 40$.

```
b=[0.3,0.6,0.6];
a=[1,-0.9];
```

// vectors `b` and `a` contain the values of the coefficients b_k and a_k.

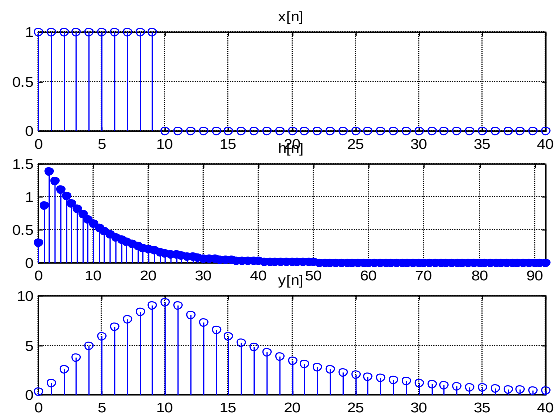
```
x=treapta(0,40,0)-treapta(0,40,10);
```

// the vector `x` corresponding to the input sequence is defined.

```
y=filter(b,a,x);
```

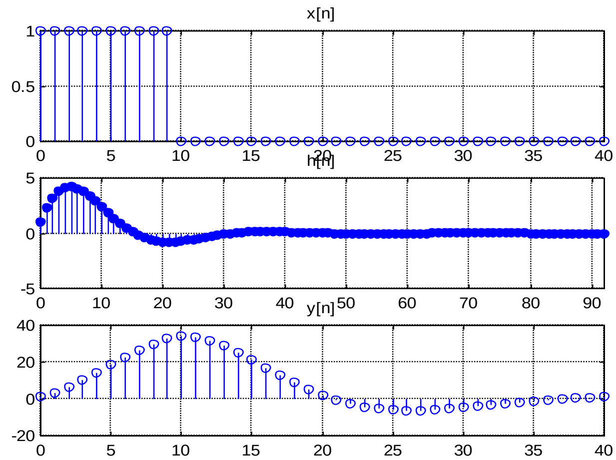
// the response of the system to the input sequence defined through the vector `x` was computed.

```
n=0:40;
subplot(3,1,1),stem(n,x),grid,title('x[n]')
subplot(3,1,2),impz(b,a),grid,title('h[n]')
subplot(3,1,3),stem(n,y),grid,title('y[n]')
```



For the second example, we proceed in a similar way:

```
b=[1,0.5];
a=[1,-1.8*cos(pi/16),0.81];
y=filter(b,a,x);
subplot(3,1,1),stem(n,x),grid,title('x[n]')
subplot(3,1,2),impz(b,a),grid,title('h[n]')
subplot(3,1,3),stem(n,y),grid,title('y[n]')
```



2.4. The frequency response of a *LTIS*

The MATLAB function `freqz` allows to determine the frequency response of a *LTIS* if we know the coefficients a_k and b_k from the finite difference equation or from the expression of the transfer function. If the function is called without the output parameters, it will plot the amplitude-frequency and phase-frequency characteristics of the respective *LTIS*.

***Syntax*:**

`[H,W] = freqz(b,a,n)``% for more information type` **`help freqz`**

- The vector `b` contains the coefficients b_k (`b` = [$b_0, b_1, ..., b_M$]) and the vector `a` contains the coefficients a_k (`a` = [1, $a_1, a_2, ..., a_N$]) and `n` is the number of points in which the frequency response `H` is computed; the vector `W` will contain the values of these `n` points (the values will be between 0 and π); it is recommended to choose `n` as a power of 2 (in order to allow an efficient computation using a FFT algorithm); *if* `n` *is not specified it will be chosen implicitly* 512.

`[H,F] = freqz(b,a,n,Fs)`

- This syntax allows specifying a value for the sampling frequency `Fs` (in Hz); the vector `F` will contain the values of the `n` points in which the frequency response `H` is computed (in this case the values will be between 0 and `Fs`/2);

`[H,W] = freqz(b,a,n,'whole')`

- Same as the previous syntax, except that the values of the `n` points, contained by the vector `W`, will be between 0 and 2π; *if* `n` *is not specified it will be chosen implicitly* 512.

`[H,F] = freqz(b,a,n,'whole',Fs)`

Same as the second syntax, except that the values of the `n` points, contained by the vector `F`, will be between 0 and `Fs`; *if* `n` *is not specified it will be chosen implicitly* 512.

`H = freqz(b,a,W)`

- The frequency response `H` is computed at the frequencies specified in the vector `W`; the values of these frequencies must be between 0 and 2π; *if* `W` *is not specified*, 512 *values of the frequency will be implicitly chosen*.

`H = freqz(b,a,F,Fs)`

- The frequency response `H` is computed at the frequencies specified in the vector `F`; the values of these frequencies must be between 0 and `Fs` (the sampling frequency in Hz).

`freqz(b,a,...)`

- Plots the amplitude-frequency and phase-frequency characteristics of the computed frequency response; any combination of the input parameters from the previous syntaxes may be used.

***Examples*:**

Determine the frequency response of a *LTIS* defined by:

1. $y[n]+0{,}9y[n-1]=0{,}3x[n]+0{,}6x[n-1]+0{,}3x[n-2]$
2. $H(z)=\dfrac{0.634-0.634z^{-2}}{1-0.268z^{-2}}$

Plot the amplitude-frequency and phase-frequency characteristics of the computed frequency response.

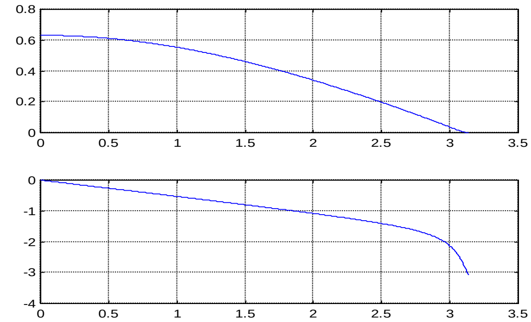
```
b=[0.3,0.6,0.3];
a=[1,0.9];
[H,W]=freqz(b,a);
```

// the frequency response `H` was computed in 512 frequency points between $[0,\pi]$.

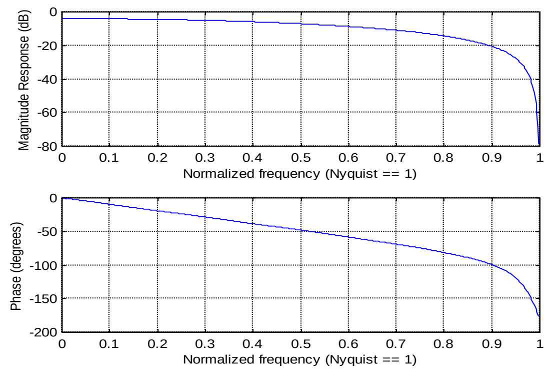
The requested graphical representations can be done in two ways:

```
figure(1)
```

```
subplot(2,1,1),plot(W,abs(H)),grid
subplot(2,1,2),plot(W,angle(H)),grid
```



```
figure(2)freqz(b,a)
```



We proceed in a similar way for the second example:

```
b=[0.634,0,-0.634];
a=[1,0,-0.268];
[H,W]=freqz(b,a);
freqz(b,a)
```

2.5. Zero-pole diagram for the transfer function

The MATLAB function `zplane` allows to display the zero-pole diagram in the case of a *LTIS* function if we know the values of the poles and zeros *or* if we just know the coefficients a_k and b_k from the finite difference equation or from the expression of the transfer function $H(z)$.

***Syntax*:**

```
zplane(z,p) % for more information type help zplane
```

- if `z` and `p` are two column vectors containing the values of the zeros and poles of the transfer function $H(z)$ respectively, then the zero-pole diagram is displayed, marking the zeros with ‘o’ and the poles with ‘x’; if there are multiple poles or zeros, their multiplicity order will be written near the respective symbol.

- if `z` and `p` are two matrices, the zero-pole diagram will be displayed for each column in a different figure.

`zplane(b,a)`

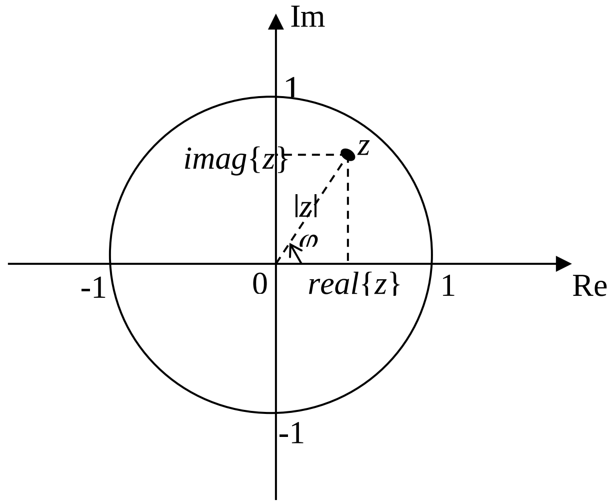
- if `b` and `a` are two line vectors containing the values of the b_k and a_k then the zero-pole diagram of the transfer function $H(z)$ will be displayed (computing the roots of the polynomials from the numerator and the denominator of the system function).

The representation of poles and zeros of a function system (pole-zeros diagram) is done in the ***Z*** plane. So, as complex numbers, these values can be expressed in polar form or cartesian form.

Any complex number *z* can be expressed in *polar form* as follows:

$$z = |z| \cdot e^{j\varphi} \tag{2.17}$$

where: - $|z|$ = magnitude of the complex number z ;
- φ = angle (phase) of the complex number z ;



As *cartesian form* the complex number z is expressed as:

$$z = real(z) + j \cdot imag(z) \tag{2.18}$$

If we have the coefficient values b_k and a_k, and we want to determine poles and zeros of the system transfer function, we can use MATLAB function `roots`, that calculates the roots of a polynomial if coefficients are known.

If we want to determine the coefficient values b_k and a_k from the values of poles and zeros of the system function, we can use MATLAB function `poly`, which calculates the coefficients of a polynomial if its roots are specified.

Check the syntax of the two functions using the help command.

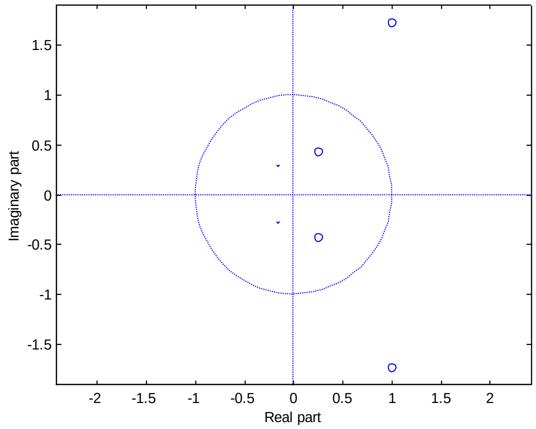
***Examples*:**

1. The transfer function of a LTIS has a zero $r=\frac{1}{2}e^{j\frac{\pi}{3}}$. The transfer function has also zeros in r^*, $\frac{1}{r}$, $\frac{1}{r^*}$ and has two poles in $q=\frac{1}{3}e^{j\frac{2\pi}{3}}$ and q^*. Plot the pole-zero diagram and determine the transfer function (determine the coefficients b_k and a_k).

```
r=1/2*exp(j*pi/3);
q=1/3*exp(j*2*pi/3);
z=[r;conj(r);1/r;1/conj(r)];
p=[q;conj(q)];
```

// the column vectors `z` and `p` contain zeros and poles of the transfer function
// and the pole-zero diagram is ploted

```
zplane(z,p)
```




```
b=poly(z)  →  b =
              1.0000   -2.5000   5.2500   -2.5000   1.0000
a=poly(p)  →  a =
              1.0000   0.3333   0.1111
```

// the transfer function coefficients b_k and a_k are computed. The resulted transfer function expression is:

$$H(z)=\frac{1-2{,}5z^{-1}+5{,}25z^{-2}-2{,}5z^{-3}+z^{-4}}{1-0{,}3333z^{-1}+z^{-2}}$$

2. Represent the pole-zero diagrams for discrete-time systems defined by:

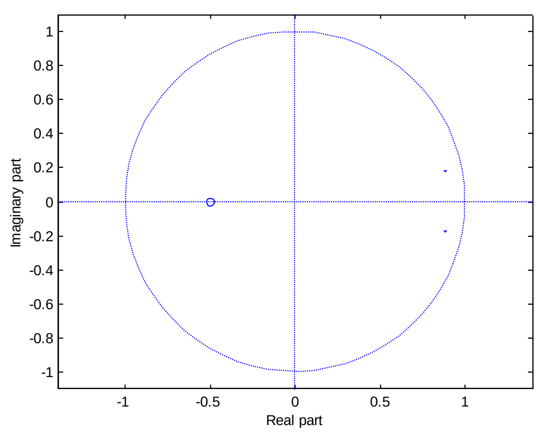
a. transfer function: $H(z)=\dfrac{1+0{,}5z^{-1}}{1-1{,}8\cos(\pi/16)z^{-1}+0{,}81z^{-2}}$

b. transfer function: $H(z)=\dfrac{(1+0{,}5z^{-1})^2}{1-1{,}8\cos(\pi/16)z^{-1}+0{,}81z^{-2}}$

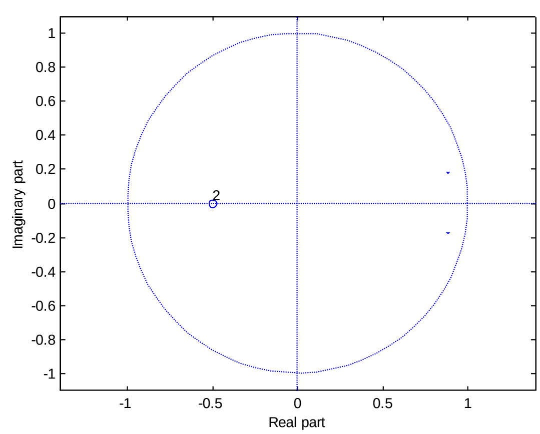
c. difference equation:

$$y[n]+0.1y[n-1]+0.1y[n-3]+0.1y[n-4]=0{,}3x[n]+0{,}6x[n-1]+0{,}6x[n-2]$$

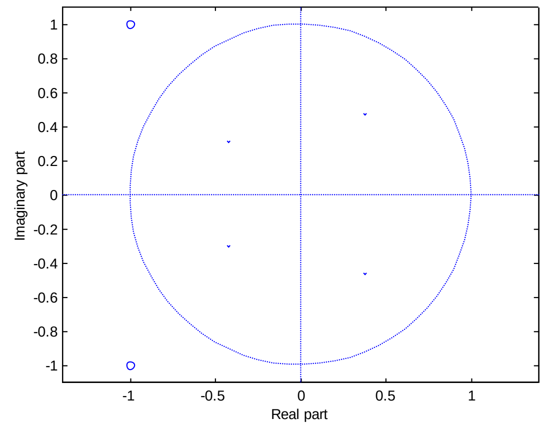
```
b=[1,0.5];
a=[1,-1.8*cos(pi/16),0.81];
zplane(b,a)
```




```
b=[1,1,0.25];
zplane(b,a)
```



```
b=[0.3,0.6,0.6];
a=[1,0.1,0,0.1,0.1];
zplane(b,a)
```



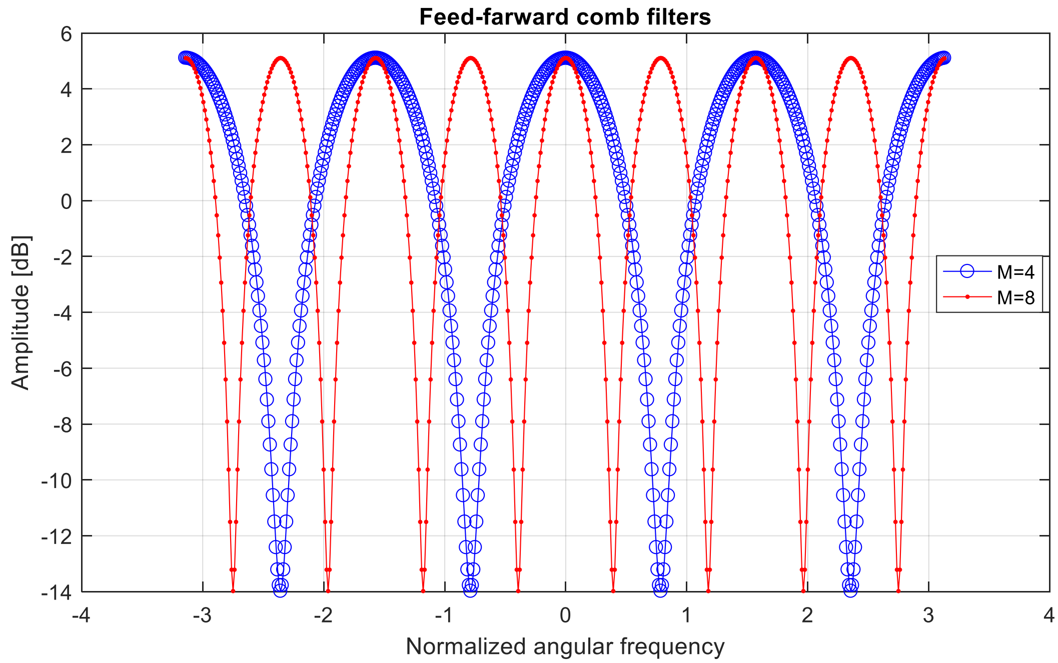
***Examples*:**

2. The transfer function of a feed-forward comb filter is: $H(z) = 1 + 0.8z^{-M}$. Represent the amplitude-frequency characteristic of this system for an order $M = 4$ and $M = 8$. Represent the pole-zero diagrams associated with these systems.

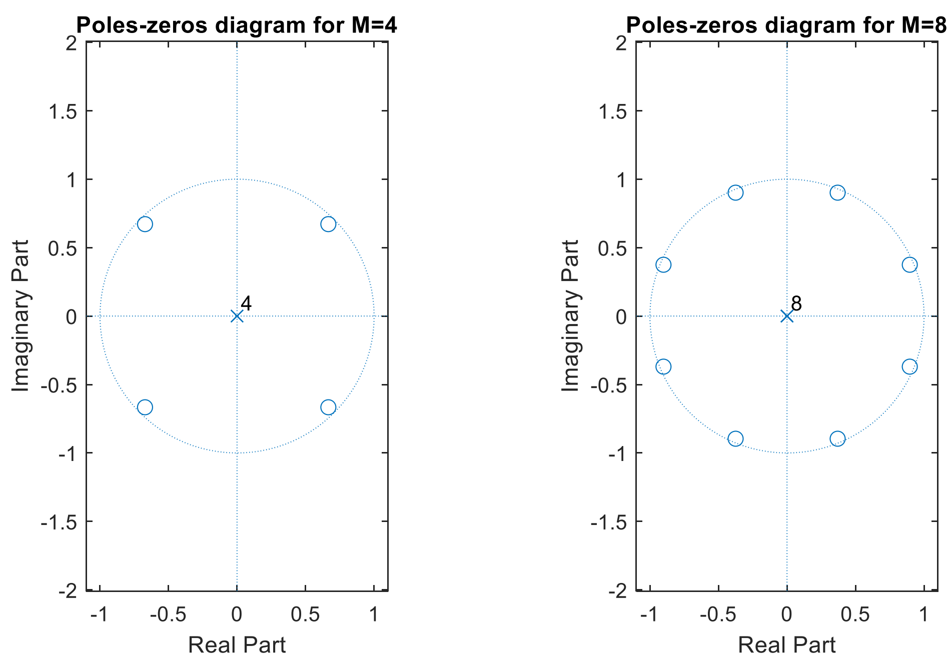
// the difference equations for such systems is: $y(n) = x(n) + 0.8 \cdot x(n - M)$, so the output is a linear combination of the current input sample and an attenuated M-tap delayed sample from the input signal.

```
b8 = [1, 0, 0, 0, 0, 0, 0, 0, 0.8];
b4 = [1 0 0 0 0.8];
a = [1];
[h4,t] = impz(b4,a);
H4 = fft(h4,512);
[h8,t] = impz(b8,a);
H8 = fft(h8,512);
w = -pi: 2*pi/512 :pi-2*pi/512;
figure(1),plot(w,20*log10(fftshift(abs(H4))),'o-b',...
    w,20*log10(fftshift(abs(H8))),'.-r'),grid
```

The resulting frequency response for both systems is presented in the fallowing figure. The appearance of a “comb” with exactly 4 teeth corresponds to the filter of order 4, while the filter of order 8 displays 8 teeth.



The pole-zero diagrams of these discrete-systems contain exactly M zeros displayed symmetrically on the unity circle, as shown in the figure below:



Remark! Filtering an audio signal through this comb filter leads to a single discrete acoustic effect of an echo.

E1. ***Exercises*:**

Let the following LTIS defined by:

1. $y[n] = x[n] - 1{,}27x[n-1] + 0{,}81x[n-2] - 0{,}5x[n-3] + 0{,}125x[n-4]$
2. $y[n] + 0{,}9y[n-1] = x[n]$
3. $y[n] + 0{,}13y[n-1] + 0{,}52y[n-2] + 0{,}3y[n-3] =$
   $= 0{,}16x[n] - 0{,}48x[n-1] + 0{,}48x[n-2] - 0{,}16x[n-3]$
4. $y[n] + 0{,}9y[n-2] = 0{,}3x[n] + 0{,}6x[n-1] + 0{,}3x[n-2]$
5. $H(z) = \dfrac{1 - 0{,}5z^{-1} + 0{,}125z^{-2} - 0{,}075z^{-3}}{1 - 0{,}8z^{-1} + 0{,}64z^{-2} - 0{,}4z^{-3} + 0{,}024z^{-4}}$
6. $H(z) = \dfrac{1}{1 - 0{,}77z^{-1} + 0{,}44z^{-3}}$
7. $H(z) = 1 - 1{,}27z^{-1} + 0{,}81z^{-2} - 0{,}5z^{-3} + 0{,}125z^{-4} - 0{,}3z^{-7}$

The input signals that can be applied can be any of the following:

I. $x_1[n] = \delta[n]$ for $0 \le n \le 40$

II. $x_2[n] = u[n]$ for $0 \le n \le 40$

III. $x_3[n] = \begin{cases} n & ,0 \le n \le 10 \\ 20 - n & ,11 \le n \le 20 \end{cases}$

IV. $x_4[n] = \sin\left(\dfrac{n\pi}{5}\right)$ for $0 \le n \le 20$

a) Determine the response of these systems to input signals I-IV and plot in the time domain (using `subplot`) the input signal, the impulse response and the output signal.
b) For each case analyzed, plot in the frequency domain (using subplot) the input signal spectrum and the output signal spectrum. And plot the magnitude and phase of the transfer function of the system.
c) Represent the pole-zero diagrams associated systems 1-7.