Alessio Hu (12345678)

Multimedia Signals Processing

Multimedia Signals Processing
Prof. Marco Marcon
DEIB - Politecnico di Milano

Something:

Something else

A.Y. 2022-2023

Contents

1 Discrete Signals

1.1 Impulse

$$\delta[n] = \begin{cases} 1, \ n = 0 \\ 0, \ n \neq 0 \end{cases}$$

1.2 Discrete unit step

$$u[n] = \begin{cases} 1, \ n \geq 0 \\ 0, \ n < 0 \end{cases}$$

1.3 Discrete exponential step

$$x(n) = a^n u(n)$$

1.4 Discrete periodic signals

$$x(n) = x(n + kT) \qquad \forall\, k \in \mathbb{Z}$$

1.5 Discrete-time sinusoids

From continuous to discrete

$$y(t) = A \cdot \cos(2\pi f_0 t + \phi) \rightarrow y(n) = A \cdot \cos(2\pi f_0 T_s n + \phi)$$

OR

$$\begin{cases} y(t) = A \cdot \cos(2\pi f_0 t + \phi) \rightarrow y(n) = A \cdot \cos(2\pi \tilde{f}_0 n + \phi) \\ \tilde{f}_0 = \frac{f_0}{F_s} = f_0 \cdot T_s \qquad \text{normalized frequency} \\ \tilde{\omega}_0 = 2\pi \tilde{f}_0 = \frac{\omega_0}{F_s} = \omega_0 T_s \end{cases}$$

- $A =$ amplitude
- $f_0 =$ frequency [Hz]
- $\omega_0 = 2\pi f_0 =$ angular frequency [rad/s]
- $\phi =$ phase [rad]
- $\frac{1}{f_0} = \frac{2\pi T_s}{\tilde{\omega}_0} =$ **period of the sinusoid [seconds]**
- $P_{samples} = \frac{1}{f_0 \cdot T_s} = \frac{1}{\tilde{f}_0} = \frac{2\pi}{\tilde{\omega}_0} = \frac{F_s}{f_0}$ **period of the sinusoid [in samples]**
- $t = n \cdot T_s = \left[start : T_s = \frac{1}{F_s} : end = \frac{1}{f_0} - \frac{1}{F_s} \right]$
- $n = 0, 1, 2, \cdots N =$ samples
- $T_s =$ sampling time or sampling period
- $F_s = \frac{1}{T_s} =$ **sampling rate/signal duration [Hz]**
- **Signal as sum of sinusoids**
  - **Principle of superposition**

$$x = \cos(\omega_1 t) + \cos(\omega_2 t) + \cos(\omega_3 t)$$

  - **Global sinusoid period [in samples]**: mcm between the sinusoid periods in samples

2 Linear Time-Invariant Systems

- Linearity = input-output realtionship is linear
- Time invariance = the output does not depend on the particular time the input is applied:

$$x(t) \to y(t) \leftrightarrow x(t-k) \to y(t-k)$$

- The system can be completely characterized by its impulse response $h(t)$

The output:

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

2.1 Convolution

- Flip the second term $h(n) \to h(-k)$
- Shift h by adding n (if positive shift to right, if negative to left) $h(n-k)$
- For output $y(n)$ sum all contributions of $x(k)$ and the shifted flipped h

Last step similar to scalar product. **The length of the convolution is the sum of the two signals-1 (n+m-1) and the support (x-axis from when the signals start to differ from 0) is the sum of the supports until the final support of the two**.

Properties of convolution:

- Commutativity
- Associativity
- Distributivity
- Convolution by pulse $x(n) * \delta(n) = x(n)$
- Convolution by a shifted pulse $x(n) * \delta(n-k) = x(n-k)$

Alternativetely, if $h(n) = h[n]$ =sum of various δ, we can apply the distributive property and sum various convolutions by δ!

2.2 Discrete delay

$$y(n) = x(n-k)$$

So:

$$y(n) = x(n) * h(n) = x(n) * \delta(n-k) = x(n-k)$$

2.3 Moving average

$$y(n) = \frac{1}{M} \sum_{m=0}^{M-1} x(n-m)$$

It is a filter, result of LTI system.

The **impulse response**:

$$y(n) = \frac{1}{M} \sum_{m=0}^{M-1} x(n) * \delta(n-m) = x(n) * \sum_{m=0}^{M-1} \delta(n-m) \rightarrow \sum_{\mathbf{m=0}}^{\mathbf{M-1}} \delta(\mathbf{n}-\mathbf{m})$$

2.4 Cosine signal and generic filter

We will get another cosine with very same frequency, but modified amplitude and phase. **Look at the exercise 20181107-1**

3 Z transform

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n) \cdot z^{-n}$$

$$z = \rho e^{j2\pi f}$$

3.1 Complex Numbers Fundamentals

$$\cos x = \frac{e^{jx} + e^{-jx}}{2} \qquad \sin x = \frac{e^{jx} - e^{-jx}}{2j}$$

$$z = \rho e^{\pm j\theta} = \rho(\cos(\pm\theta) + j\sin(\pm\theta))$$

$$|z| = \sqrt{Re(z)^2 + Im(z)^2} = \rho$$

3.2 Notable transforms

An example

$$x(n) = \delta(n) + \delta(n+1) + 2\delta(n-2)$$

$$X(z) = 1 + z + 2z^{-2}$$

- $Z\{\delta(n-k)\} = z^{-k}$, a delay of k samples
- $Z\{x(n-k)\} = X(z)z^{-k}$
- $Z\{a^n u(n)\} = \frac{1}{1-az^{-1}}$, discrete exponential term
- $Z\{ax(n) + by(n)\} = aX(z) + bY(z)$
- $Z\{x(-n)\} = X(z^{-1})$
- $Z\{nx(n)\} = -z\frac{dX(z)}{dz}$
- $Z\{x(n) * y(n)\} = X(z) \cdot Y(x)$, convolution theorem
- $Z\{-a^n u(n)\} = \frac{1}{1-z^{-1}}$, unitary step
- $Z\{r^n \cos(\omega_0 n)u(n)\} = \frac{1-r\cos(\omega_0)z^{-1}}{1-2r\cos(\omega_0)z^{-1}+r^2z^{-2}}$

3.3 Z transform expressions

1. $$X(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1 z^{-1} + \cdots b_N z^{-N}}{a_0 + a_1 z^{-1} + \cdots + a_D z^{-D}}$$

   Useful to compute the inverse Z transform

2. $$X(z) = z^{D-N}\frac{b_0}{a_0}\frac{\prod_{i=1}^{N}(z - z_i)}{\prod_{i=1}^{D}(z - p_i)}$$

   Useful for filter characterization

3.4 Z transform relationship with LTI systems

Given $x(n)$ and $h(n)$ (impulse response of LTI system)

$$y(n) = x(n) * h(n)$$

The same system can also be described by a linear difference equation with constant coefficients

$$\sum_{k=1}^{D} a_k y(n-k) = \sum_{k=0}^{N} b_k x(n-k)$$

$$y(n) = \underbrace{\sum_{k=0}^{N} b_k x(n-k)}_{\text{Moving Average (FIR)}} - \underbrace{\sum_{k=1}^{D} a_k y(n-k)}_{\text{Autoregressive, feedback (IIR)}}$$

Converting in Z domain

$$\sum_{k=1}^{D} a_k y(n-k) = \sum_{k=0}^{N} b_k x(n-k)$$

$$Y(z) \sum_{k=0}^{D} a_k z^{-k} = X(z) \sum_{k=0}^{N} b_k z^{-k}$$

As $Y(z) = X(z)H(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}}$$

At denominator IIR part

3.5 Inverse Z transform

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}}$$

Given

$$H(z) = \sum_{n=0}^{k} h(n) z^{-n}$$

We can compute its root decomposition

$$H(z) = h_0 \prod_{n=1}^{k} (1 - z_n z^{-1}) \qquad h_0 = H(n=0)$$

z_n are called roots of the polynomial $H(z)$. Thanks to convolution theorem

$$H(z) = h_0 \cdot H_1(z) \cdot H_2(z) \cdots H_k(z)$$

$$\Downarrow$$

$$h(n) = h_0 \cdot h_1(n) * h_2(n) * h_3(n) * \cdots * h_k(n)$$

Where

$$h_i(n) = Z^{-1}\{1 - z_i z^{-1}\} = \delta(n) - z_1 \delta(n-1)$$

3.6 Partial fract expansion for computing Z^{-1}

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} = \sum_{k=1}^{D} \sum_{m=1}^{M} \frac{r_{k_m}}{(1 - p_k z^{-1})^m} + \underbrace{\sum_{k=0}^{N-D} c_k z^{-k}}_{N \geq D}$$

M is the multiplicity of the root (or pole) p_k. The Z transform inversion is the sum of simple inversions (causal):

- $Z^{-1}\left\{\frac{r_{k_1}}{(1-p_k z^{-1})}\right\} = r_{k_1} \cdot (p_k)^n u(n)$
- $Z^{-1}\left\{\frac{r_{k_2}}{(1-p_k z^{-1})^2}\right\} = r_{k_2} \cdot (n+1)(p_k)^n u(n)$
- $Z^{-1}\left\{c_k z^{-k}\right\} = c_k \cdot \delta(n-k)$

3.7 Zeros-Poles factorization

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} = z^{D-N} \frac{b_0}{a_0} \frac{\prod_{i=1}^{N}(z - z_i)}{\prod_{i=1}^{D}(z - p_i)}$$

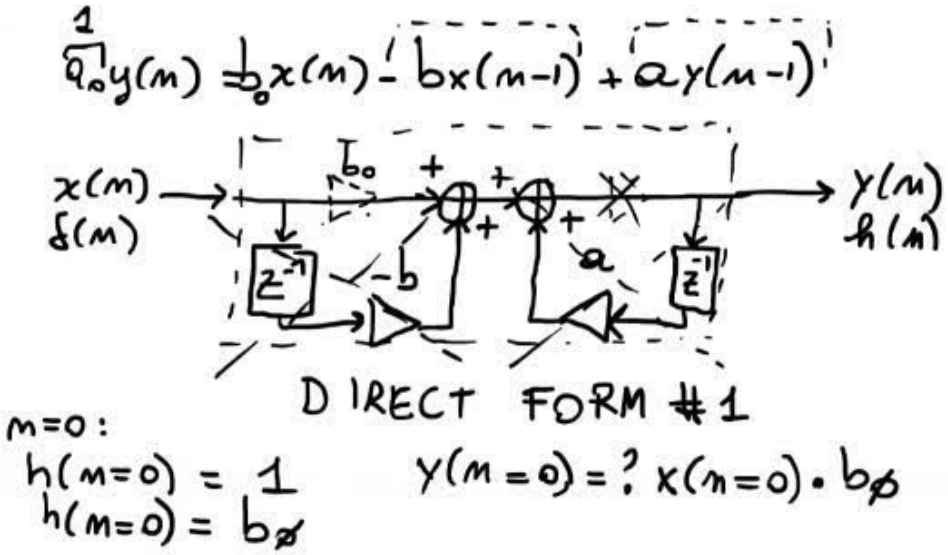
- For a system to be stable

  $$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

  If instable output of the system goes to the infinite
- If all poles of the $H(z)$ are inside the unitary circle ($|p_i| < 1 \; \forall \, i$) the system is stable
- If one positive zero is inside the unitary circle, it is called minimum phase
- If one positive zero is outside the unitary circle, it is called maximum phase
- **It is FIR if there is no denominator**
- **It is IIR if there is feedback** $y = y...$

To move from maximum phase to minimum phase, consider the reciprocals of the zeros.

3.7.1 Transfer function evaluation in time domain: direct form #1



Direct form #1, in this case for $n < 0$ the system for us it does not exist, so zero: so $h(n=0) = b_0 = 1$ in our case

3.7.2 From generic output to transfer function and complex conjugates

Consider

$$y(n) = 0.5x(n) - 2x(n-1) + x(n-2) - 2\rho\cos(\theta)y(n-1) - \rho^2 y(n-2)$$

We can easily find $H(z)$, the coefficients of the y on the right must change sign as they go to the left:

$$H(z) = \frac{0.5 - 2z^{-1} + z^{-2}}{1 + 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2}}$$

Also $h(n=0) = b_0 = 0.5$ and $y(n=0) = x(n=0) \cdot 0.5$

The $1 + 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2}$ represents 2 complex conjugates:

$$1 - 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2} \rightarrow \rho(\cos(\theta) \pm j\sin(\theta))$$

$1 + 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2} \rightarrow -\rho e^{\pm j\theta}$
$1 - 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2} \rightarrow \rho e^{\pm j\theta}$

Their product is a real coefficient!

In general if the numerator and the denominator are expressed with z^{-1}, solve them normally (imposign $z^{-1} = w$ and then invert the results) or exploiting the above formulas then introduce a number of zeros (if solved the denominator) or poles (if solved the numerator) equal to the maximum grade of the polynomial, which will simplify.

3.7.3 Maximum phase to minimum phase

$$H(z) = h_0 + h_1 z^{-1} + \cdots + h_N z^{-N} \Leftrightarrow G(z) = z^{-N} H^*(z^{-1})$$

The G version will have the same coefficients but in reverse order and **exact same magnitude**

The roots that are maximum phase will become minimum phase by choosing their reciprocal in conjugate position, an example:

$$H_M(z) = \frac{\left(1 - 2\sqrt{2}z^{-1} + 4z^{-2}\right)\left(1 + 2\sqrt{2}z^{-1} + 4z^{-2}\right)}{4 + z^{-2}}$$

The numerator (only it) introduce maximum phase, write their G version:

$$H_m(z) = \frac{\left(4 - 2\sqrt{2}z^{-1} + z^{-2}\right)\left(4 + 2\sqrt{2}z^{-1} + z^{-2}\right)}{4 + z^{-2}}$$

Alternatively As $\rho = 2$, now $\rho = \frac{1}{2}$:

$$H_m(z) = A\frac{\left(1 - \frac{\sqrt{2}}{2}z^{-1} + \frac{1}{4}z^{-2}\right)\left(1 + \frac{\sqrt{2}}{2}z^{-1} + \frac{1}{4}z^{-2}\right)}{4 + z^{-2}}$$

3.8 All-pass filter

It has this form:

$$H(z) = \frac{c + z^{-1}}{1 + cz^{-1}}$$

Has magnitude constant=1, keeps magnitude and changes the phase. To achieve it place the pole in a reciprocal conjugate position w.r.t. zero (if zero in 2, put pole in $\frac{1}{2}$)

$$y(n) = cx(n) + x(n-1) - cy(n-1)$$

More in general it has this form:

$$H(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_{n-1} z^{n-1} + a_n z^{-n}}{a_n + a_{n-1} z^{-1} + \cdots + a_1 z^{n-1} + a_0 z^n}$$

4 Discret Time Fourier Transform (DTFT)

Discrete time sequence **non periodic** and I compute a continuous frequency signal (continuous in frequency domain). Defined as

$$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn}$$

- f is the normalized frequency in $[0,1)$ or $[-0.5,0.5)$
- $\omega = 2\pi f$ is the normalized angular frequency in $[0,2\pi)$ or $[-\pi,\pi)$
- DTFT is periodic with period = 1 in normalized frequency, period = 2π in normalized omega, period = Fs in frequency [Hz]

4.1 Properties

- DTFT$\{x(n) * y(n)\} = X(f) \cdot Y(f)$
- DTFT$\{x(n-k)\} = X(f) \cdot e^{-j2\pi fk}$
- If $x(n)$ is real, $X(f) = X^*(-f)$, symmetric behavior
- Relationship with Z-transform: $X(f) = X(z)\Big|_{|z|=1}$

Given

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} \Rightarrow H(f) = H(z)\Big|_{|z|=1}$$

- The amplitude is

$$|H(f)| = |H(z)|\Big|_{|z|=1} = \frac{\left|\sum_{k=0}^{N} b_k z^{-k}\right|}{\left|\sum_{k=0}^{D} a_k z^{-k}\right|}\Bigg|_{|z|=1}$$

- The phase (depends on the zeros, not on the poles) is

$$\angle(H(f)) = \angle(H(z))\Big|_{|z|=1} = \angle\left(\sum_{k=0}^{N} b_k z^{-k}\right) - \angle\left(\sum_{k=0}^{D} a_k z^{-k}\right)\Bigg|_{|z|=1}$$

5 Discret Fourier Transform (DFT)

Discrete time sequence **periodic**. Discrete both in time and frequency, finite number/summation:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n}$$

$X(k)$ with $k = [0, N-1]$. While DTFT is continuous in the frequency domain

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n}$$

In the DFT we assume that the period is signal w.r.t. provided N

5.1 Relationship with DTFT

- While in DTFT $f \in [0,1)$, with DFT $f = \left[0, \frac{1}{N}, \cdots, \frac{N-1}{N}\right] \rightarrow$ length of N
- The inverse (IDFT):

  $$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{2j\pi \frac{k}{N} n}$$

  Length of N
- **DFT==DTFT only if FIR**, if IIR we are truncating when applying DFT, losing info
- DTFT continuous over frequencies, DFT discrete
  - Sampling in Fourier = periodicity in time
  - Sampling in time = periodicity in frequency/Fourier
- Both DFT and IDFT are periodic with a period of N samples:

$$X(k + tN) = X(k) \qquad \forall\, t \in [-\infty, \infty]$$

$$x(n) = IDFT(X(k)) = x(n + tN) \qquad \forall\, t \in [-\infty, \infty]$$

5.2 DFT as a matrix product

The DFT of a sequence $x(n)$ can be computed by:

$$X = W_N \cdot x$$

Where:

$$W_N = \begin{bmatrix} W_i^0 & W_i^0 & W_i^0 & W_i^0 & \cdots & W_i^0 \\ W_i^0 & W_i^1 & W_i^2 & W_i^3 & \cdots & W_i^{N-1} \\ W_i^0 & W_i^2 & W_i^4 & W_i^6 & \cdots & W_i^{2(N-1)} \\ W_i^0 & W_i^3 & W_i^6 & W_i^9 & \cdots & W_i^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ W_i^0 & W_i^{N-1} & W_i^{2(N-1)} & W_i^{3(N-1)} & \cdots & W_i^{(N-1)^2} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j2\pi/N} & e^{-j4\pi/N} & \cdots & e^{-j2\pi(N-1)/N} \\ 1 & e^{-j4\pi/N} & e^{-j8\pi/N} & \cdots & e^{-j4\pi(N-1)/N} \\ 1 & \cdots & \cdots & \cdots & \cdots \\ 1 & e^{-j2\pi(N-1)/N} & e^{-j4\pi(N-1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix}$$

$$W_i = e^{-j\frac{2\pi}{i}}$$

Or

$$W_N = e^{-j\frac{2\pi}{N}kn}$$
$$\underline{k} = [0,1,\cdots,N-1]$$
$$\underline{n} = [0,1,\cdots,N-1]$$

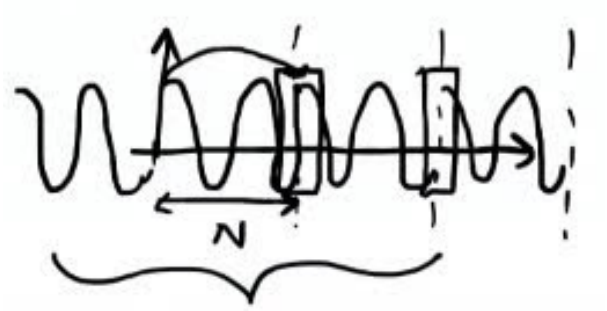
The inverse of $W: N \times N$ **is its conjugate transpose divided by** N.

$$y(n) = W^{-1}Y(k)$$

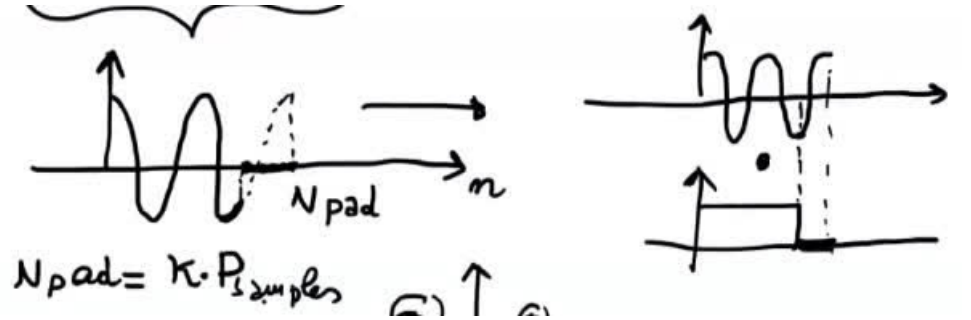
If using matrices, $Y(k) = X(k) \cdot H(k)$ **element-wise product**.

5.3 Zero padding

Zero padding means windowing, which will result in a sinc behavior in the Fourier domain. When we compute the DFT, if the repetition of the signal results in discountinuities:



We introduce zero till we get a signal that repeated will not have discountinuities. This new signal obtained is made of the product of two signals: itself and a rectangle that ends where the previous signal ended (will become 0 onward, so N_{pad} must be multiple of period)



In the Fourier/frequency domain so we will get the **convolution** of transform of those two signals: the transform of the first will be two pulses (δ) at the peaks, while the transform of the rectangle will be a sinc.
Convolving the two, we will get two sincs due to the effect of the window, each one the peaks (in the example peaks at $2Hz$ as frequency of sinusoid $f_0 = 2Hz$):

But when discretize the DFT, we should be careful to put the peak x in the axis/in the interval plotted:

$$k\frac{F_s}{N_{pad}} = f_0$$

$$N_{pad} = k\frac{F_s}{f_0} = k \cdot P_{samples}$$

Same conclusion as before.

But if **strong density** in the spectrum (a lot of padded zeros), introducing a sinc and the peak will not be exactly at f_0, small overlap on the maximum due to tails, introduced errors.

5.3.1 Conditions

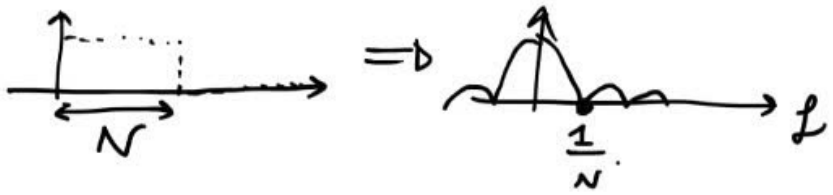
1. $N_{pad} = k \cdot P_{samples}$. **If 2+ signals summed, the global period in samples is the least common multiple (mcm) of the periods**. What if one of the periods is not an integer?

$$\begin{cases} P_1 = 25 \\ P_2 = \frac{50}{2.2} = \frac{250}{11} \end{cases}$$

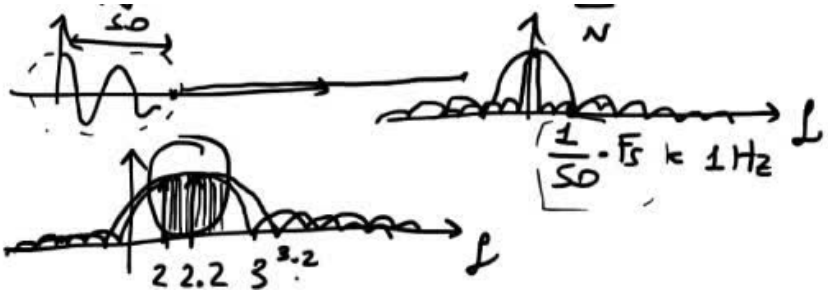
$$lcm(25, \frac{250}{11}) = lcm(25, 250) = 250$$

5.3.2 Alternative to zero-padding for discountinuities in DFT

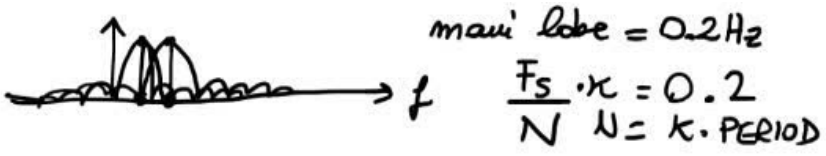
When we are windowing we have a rectangular behavior. If its length is N, its first zero is $\frac{1}{N}$



Suppose a signal with 50 samples, the size of the main lobe we are introducing in the frequency (the first zero) is $\frac{1}{50}$ in the normalized domain or $\frac{1}{50}F_s$ in the Hz domain. Suppose $F_s = 50$:



Two lobes sum up, so using zero-padding useless. We have to reduce size of main lobe of the sinc: increasing number of samples.



So the Alternative is to redefine the signals in time domain with samples equal to the global period in samples immediately: **more measurement, add more samples**.

5.4 Discontinuity

In the Fourier domain introduces high frequency content

5.5 DFT of sinusoids: pulses and peaks

A sinusoid (a cosine) with frequency f_0 will have two δ's in the frequency domain, one at f_0 the other at $-f_0 = F_s - f_0$ (mirrored w.r.t. $F_s/2$ Nyquist, if normalized frequencies, it is at $1 - \tilde{f}_0$, $\pm\tilde{\omega}_0$). If we add a second signal with a different frequency f_0', we will have two additions δ's at f_0' and $-f_0'$. In the magnitude/amplitude plot, the two pulses will have $\frac{N}{2}$ height

Given a sinusoid with F_s [Hz] and f_0 [Hz], its DFT will have two peaks, one in f_0, the other one in $F_s - f_0$, symmetric w.r.t. the Nyquist frequency. For example, given:

$$x(t) = A * \cos(2\pi f_0 t) \qquad F_s = 50Hz \qquad f_0 = 2Hz$$

$$x(n) = A * \cos\left(\frac{2\pi}{25}n\right)$$

Its DFT is:

$$|X(\omega)| = \frac{N}{2}\delta\left(\omega - \frac{2\pi}{25}\right) + \frac{N}{2}\delta\left(\omega - \left(2\pi - \frac{2\pi}{25}\right)\right)$$

5.6 DFT and IDFT periodicty

Given a non periodic sequence $x(n)$ defined over N samples, its N-periodic extension is defined as:

$$\tilde{x}(n) = \sum_{t=-\infty}^{\infty} x(n - tN)$$

Same with DFT:

$$\tilde{X}(k) = \sum_{t=-\infty}^{\infty} X(k - tN)$$

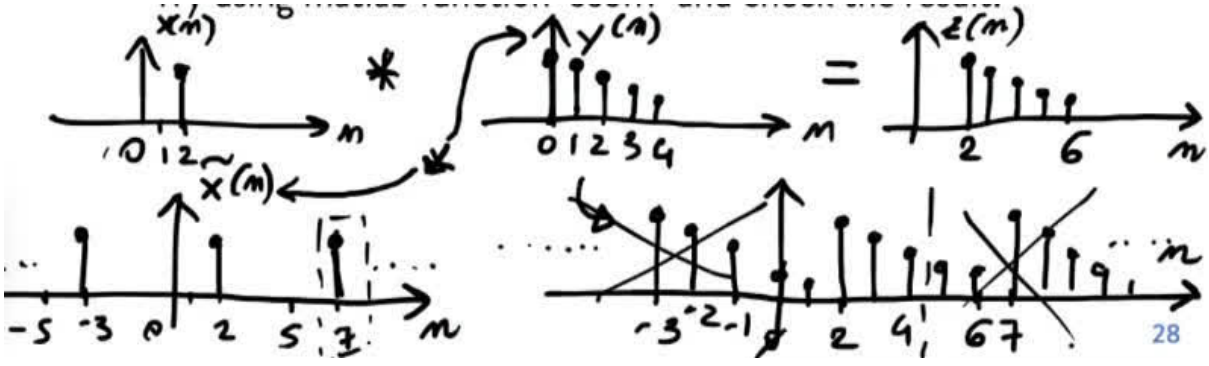
5.6.1 Cyclic/circular convolution

For the DFT:

$$X_1(k) \cdot X_2(k) \neq x_1(n) * x_2(n)$$

$$IDFT\left\{X_1(k) \cdot X_2(k)\right\} = x_1(n) \circledast x_2(n) = \sum_{m=0}^{N-1} x(m)\tilde{y}(n-m)$$

The cyclic/circular convolution holds

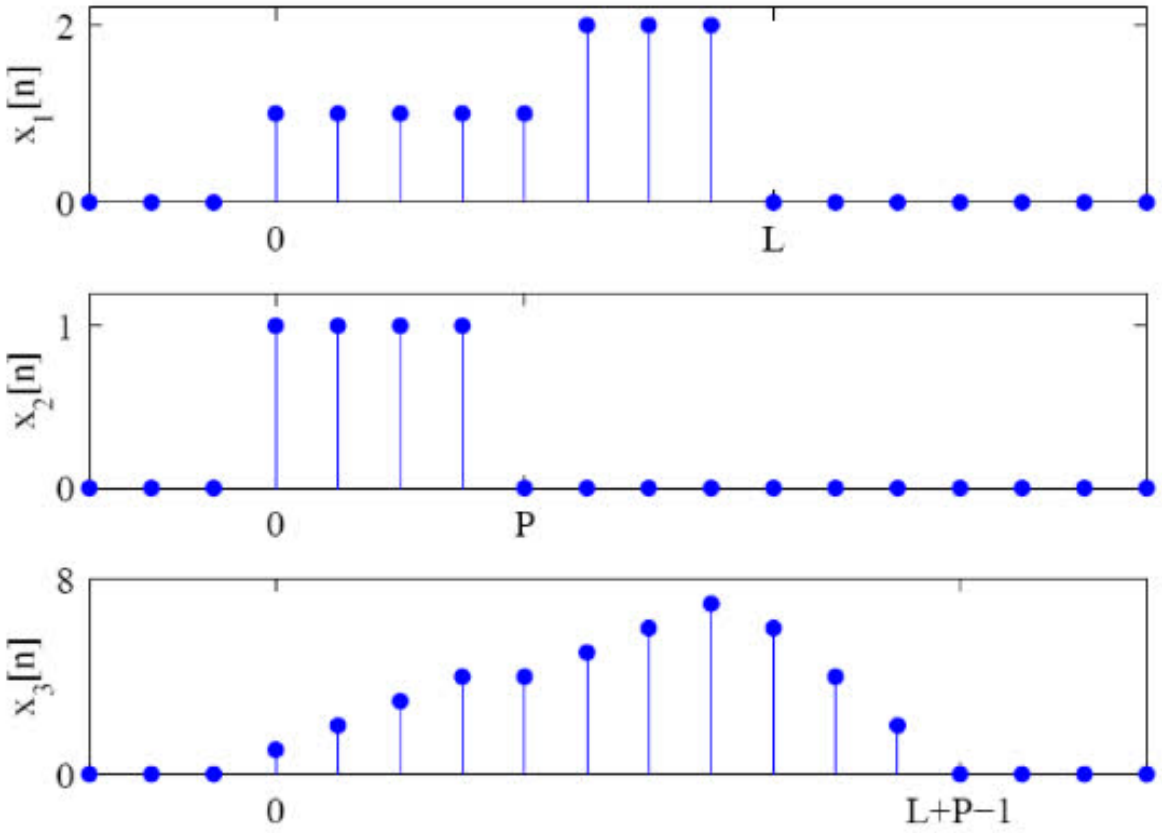


So to compute circular convolution I can use the DFT: compute product of DFTs and then invert.

But LTI systems such as communication, radar, audio and image systems all work with linear convolution, so to compute linear convolution in DFT how do we do?

5.6.2 Linear and cyclic convolution

Given a sequence with length L and a sequence with length P, the maximum length of the linear convolution is $N_c = L + P - 1$.



The cyclic convolution computed over a generic number N of samples is equal to the linear convolution if N is long enough to avoid periodic artifacts. Therefore

$$N \geq L + P - 1$$

Pad with zeros the two sequences until reaching N samples, we will not introduce artifacts and the result of the cyclic convolution will be equal to that of the normal convolution.

5.7 Long convolution

The input sequence $x(n)$ can be very long, zero-padding can be unfeasible if it is unknown as well (real-time applications), so we can segment the signal into smaller blocks and process them separately

5.7.1 Overlap and add

- Let the impulse resposne $h(n)$ long P, decompose the input $x(n)$ into **non-overlapped** blocks with length L
- For each block compute the output as

$$IDFT(X_n(k) \cdot H(k)), k \in [0, N = L + P - 1)$$

- Sum the overlapping portions between the results

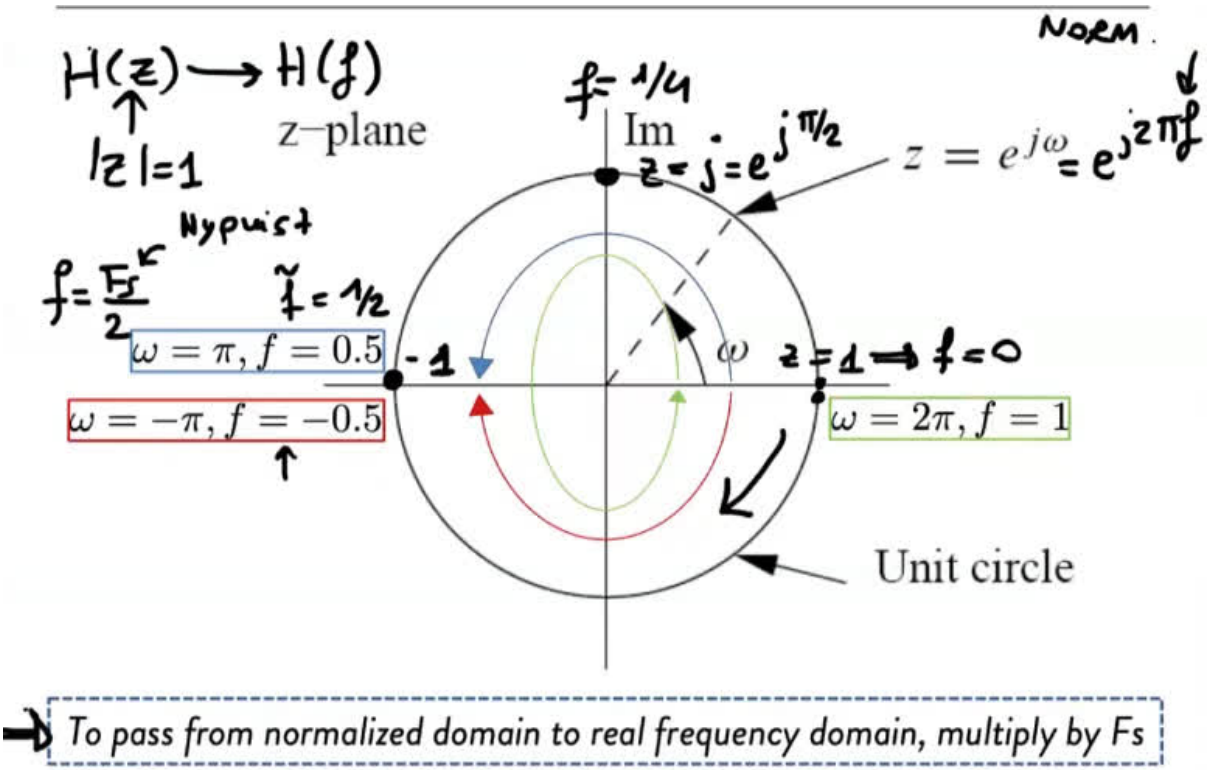
5.7.2 Overlap and save

- Let the impulse resposne $h(n)$ long P, decompose the input $x(n)$ into **overlapped** blocks with length $L > P$ with overlap $P-1$
- The circular convolution evaluated over L sampels is different from the linear convolution only in the first $P-1$ samples (periodic artifacts)
- Pad the first block with $P-1$ zeros at the beginning
- Overlap blocks by $P-1$ samples
- For each block compute the circualr convolution ofer L samples and discard the first $P-1$ samples

$$IDFT(X_n(k) \cdot H(k)), k \in [0, N = L+P-1)$$

- Sum the overlapping portions between the results

6 1D Digital filters



If asks for Hz domain, multiply by F_s. Reminder, a filter:

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} = z^{D-N} \frac{b_0}{a_0} \frac{\prod_{i=1}^{N}(z - z_i)}{\prod_{i=1}^{N}(z - p_i)} = \frac{b_0}{a_0} \frac{\prod_{i=1}^{N}(1 - z_i z^{-1})}{\prod_{i=1}^{N}(1 - p_i z^{-1})}$$

- The poles
  - Are associated with the autoregressive part, they generate IIR filter $z \to p_i$, in the frequency the filter will increase $H(f)$
  - The filter amplitude response enhances frequencies which are near the poles
  - If poles are outside the unit circle and the filter is causal, the system is unstable
- The zeros
  - Are associated with the moving average of the filter, they generate FIR filters
  - The filter amplitude response attenuates frequencies which are near the zero
  - Zeros influence also the phase of the filter (they do not influence stability)
    - Minimum phase zeros if $z < 1$, inside unit circle
    - Maximum phase zeros if $z \geq 1$, outside unit circle

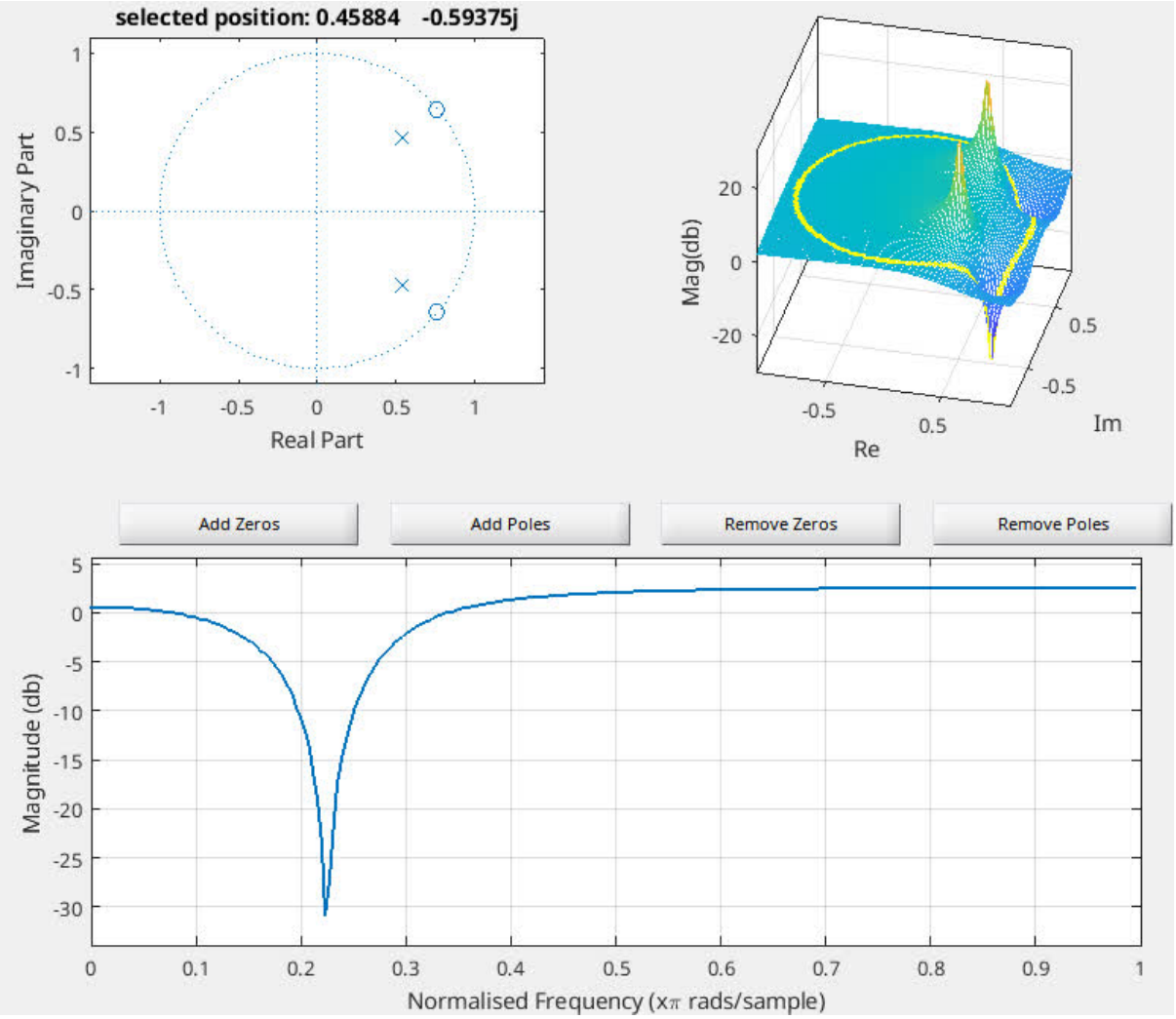
6.1 Filter design: remarkable LTI filters

- Place poles close tot he unit circle in frequencies that must be emphasized
- Place zeros according to the desired phase response, the closer are to the unit circle the higher frequency attenuation

6.1.1 Notch

I want to delete a specific frequency:

1. Put two complex zeros on the unit circle
2. Put two poles with absolute value less than 1, close to unit circle



6.1.2 Magnitude square function

The magnitude response of a LTI system is:

$$M(f) = |H(f)|^2 = H(f) \cdot H^*(f) = H(z) \cdot H^*(z^{-1})\big|_{|z|=1}$$

Given a generic rational transfer function

$$H(z) = \frac{b_0}{a_0} \frac{\prod_{i=1}^{N}(1 - z_i z^{-1})}{\prod_{i=1}^{N}(1 - p_i z^{-1})}$$

$$\Downarrow$$

$$M(z) = H(z)H^*(z^{-1}) = \frac{|b_0|^2}{|a_0|^2} \frac{\prod_{i=1}^{N}(1 - z_i z^{-1})(1 - z_i^* z)}{\prod_{i=1}^{D}(1 - p_i z^{-1})(1 - p_i^* z)}$$

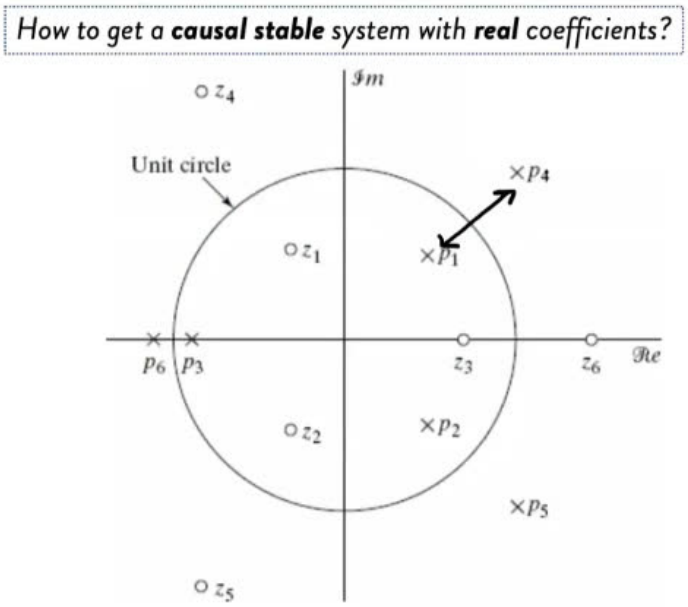
Where $H^*(z^{-1})$ means $H(z)$ with complex conjugates evaluated in z^{-1}, star in the coefficients

- For each zero z_i of $H(z)$ there is another zero at $\frac{1}{z_i^*}$

$$\frac{1}{z_i^*} = \frac{1}{\rho_i e^{-j\theta_i}} = \frac{1}{\rho_i} e^{j\theta_i}$$

  The angle/phase is the same, only distance from origin changes

- For each pole p_i of $H(z)$ there is another pole at $\frac{1}{p_i^*}$
- $M(z)$ presents poles and zeros in conjugate reciprocal pairs



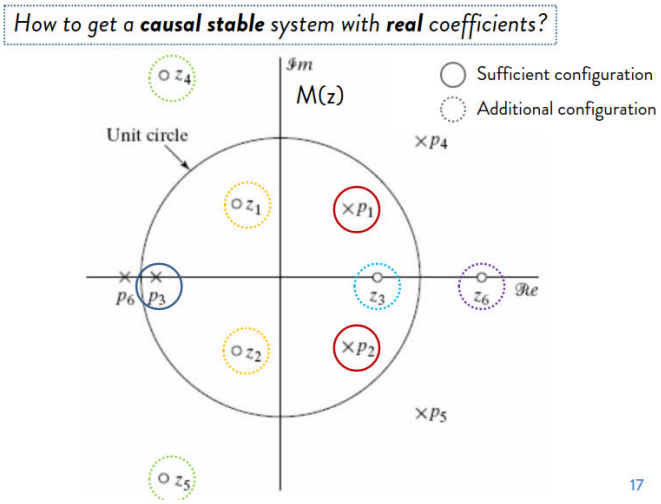
Given a magnitude response requirement $M(z)$ for $H(z)$, given stability and causality requirements for $H(z)$:

- The poles of $H(z)$ are those of $M(z)$ inside the unit circle anda re uniquely identified (for stability constraints)
- The zeros of $H(z)$ are not uniquely identified (as zeros influence only the phase and we have no constraints on phase), so the zeros can be both inside and outside the unit circle
- Given a causal FIR filter $H(z)$ of order N, it has the same magnitude response $M(z)$ of the causal FIR filter:

$$G(z) = z^{-N} H^*(z^{-1})$$

  Minimum to maximum phase filter and viceversa

From the previous example, in order to have causal and stable system we select the following zeros and poles:



6.1.3 Allpass filters

Flat behavior in the frequency, which means constant gain and any phase:

$$|H_{ap}(f)| = |H_{ap}(z)||_{|z|=1} = 1 \qquad \forall f$$

Given the previous considerations, a generic causal allpass filter is:

$$H_{ap}(z) = z^{-K} e^{j\phi} \frac{A(z)}{\tilde{A}(z)}$$

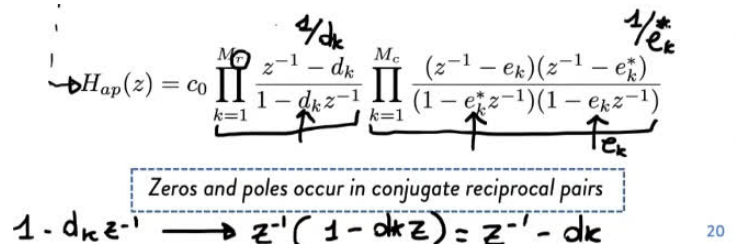
Where

$$\begin{cases} A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N} \\ \tilde{A}(z) = z^{-N} A^*(z^{-1}) = a_N^* + a_{N-1}^* z^{-1} + \cdots + a_2^* z^{2-N} + a_1^* z^{1-N} + z^{-N} \end{cases}$$

The denominator is the G version of the numerator, so their magnitudes cancel out.

$$H_{ap}(z) = z^{-N} e^{j\theta} \frac{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}}{a_N^* + a_{N-1}^* z^{-1} + \cdots + a_2^* z^{2-N} + a_1^* z^{1-N} + z^{-N}}$$

A general form to represent an allpass real valued impulse response is:



We introduced a coefficient c_0 in order to guarantee magnitude of 1. Basically

$$H(z) = \left| c_0 \frac{B_{ap}(z)}{A_{ap}(z)} \right|_{|z|=1} = 1$$

We pass to the frequency domain. As that fraction is flat in the frequency domain, we choose frequency $f = 0, z = 1$:

$$|c_0| \frac{|B_{ap}(z=1)|}{|A_{ap}(z=1)|} = 1$$

$$\Rightarrow c_0 = \pm \frac{|A_{ap}(z=1)|}{|B_{ap}(z=1)|}$$

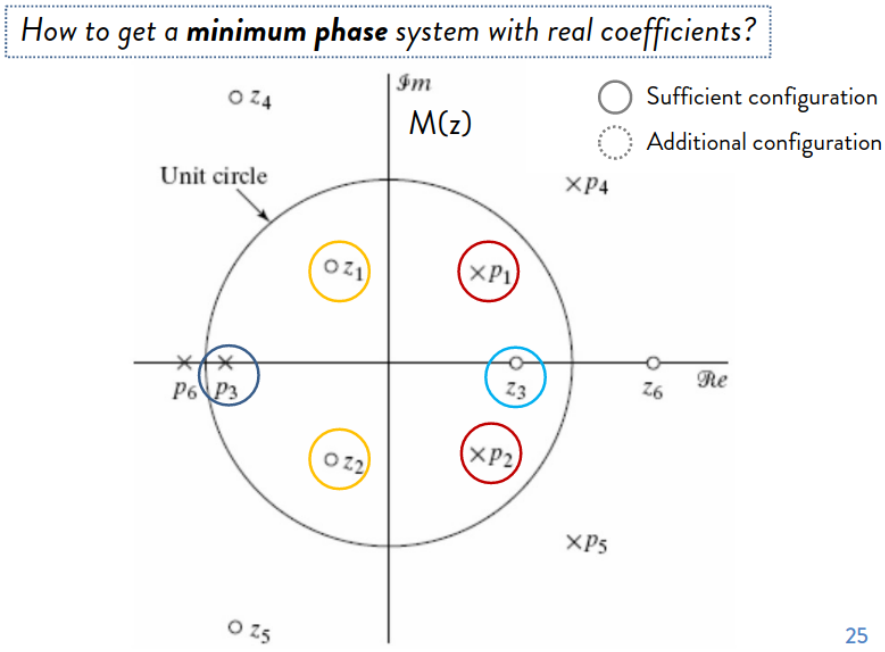
So to find c_0 just sum all the coefficients and then compute the division.

Zeros and poles are in conjugate reciprocal pairs. Properties:

- Cascade of two allpass filters is again an allpass filter
- Each pole of an allpass system is associated with a conjugate reciprocal zero
- The magnitude of many cascaded allpass filters is always the same

6.1.4 Minimum phase filters

- Minimum phase filters are such that both $H(z)$ and $\frac{1}{H(z)}$ are stable and causal
- The poles must be inside the unit circle
- The zeros must be inside the unit circle
- Given a square magnitude response $M(z)$, there is a unique system whose zeros and poles are inside the unit circle and it is called minimum phase system



Properties of allpass: **Any rational causal stable system can be decomposed into the multiplication of a minimum phase system and an allpass system**

$$H(z) = H_{min}(z)H_{ap}(z)$$

Causal and stable, so all poles are inside the unit circle!
$H_{min}(z)$ contains:

- The poles and the zeros of $H(z)$ that lie inside the unit circle
- Zeros that are conjugate reciprocals of the zeros $H(z)$ lying outside the unit circle

$H_{ap}(z)$ contains:

- All the zeros of $H(z)$ that lie outside the unit circle
- Poles that are conjugate reciprocals of the zeros of $H(z)$ lying outside the unit circle

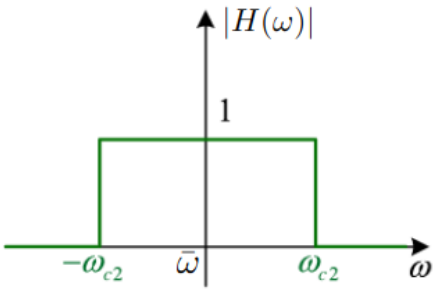
6.2 Digital filters' design

1. Specify always the characteristics of the filter in frequency domain, not in time domain (e.g. lowpass, highpass, bandpass...)
2. Approximate these properties using a discrete-time system, find the filter coefficients
3. Realize the system using finite precision arithmetic

6.2.1 Ideal filters

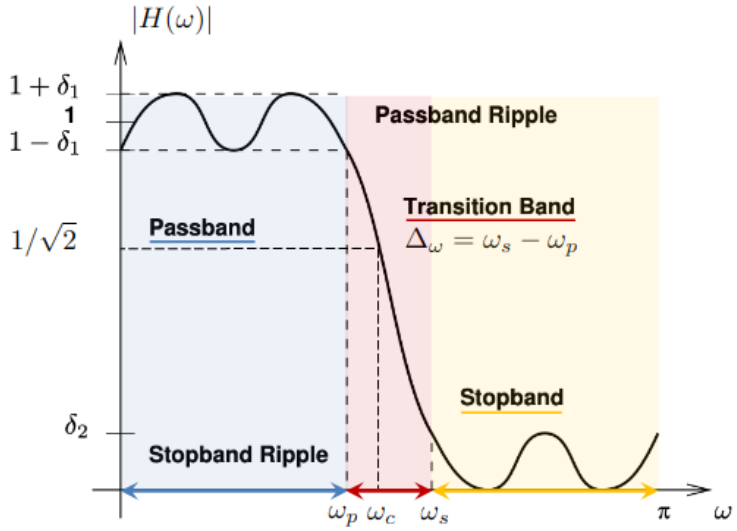
Ideal filter:

- low-pass if $\bar{\omega} = 0$
- band-pass if $0 < \bar{\omega} < \pi$
- high-pass if $\bar{\omega} = \pi$



Outside the rectangle is called **stopband**. The impulse response of this filter is $\approx$ the sinc function. It is **non-causal (sinc has on the left non-causal part)** with an **infinite delay**, **real systems can only approximate**.

6.2.2 Real filters



δ_1 Peak passband ripple

$1 - \delta_1$ Minimum passband gain

δ_2 Peak stobband ripple

ω_c 3dB or cut-off frequency

Oscillations both in passband and in stopband. Cutoff frequency is the frequency in which I pass from the passband to the stopband.

As I don't want ripples and we want transition band to 0:

- Peak ripple $\to 0$
- Transition band $\to 0$

We consider FIR and IIR cases:

FIR:

- Only zeros
- Always stable
- Can be linear phase
- It should be high order for best performances

IIR:

- Poles and zeros
- May be unstable
- Difficult to control phase
- Lower order (1/10-th of FIR) for high performances

Allpass must be FIR.

6.2.3 FIR filter design: windowing method

I want a rectangle in the frequency, but in the time I have a sinc: obtain a FIR filter by truncating an infinite duration impulse response

- Given the ideal $h_i(n)$, build $h(n) = h_i(n)w(n)$
- $w(n)$ is a finite duration window, in frequency domain it becomes convolution
- $H(f)$ is a blurred version of the ideal filter $H_i(f)$

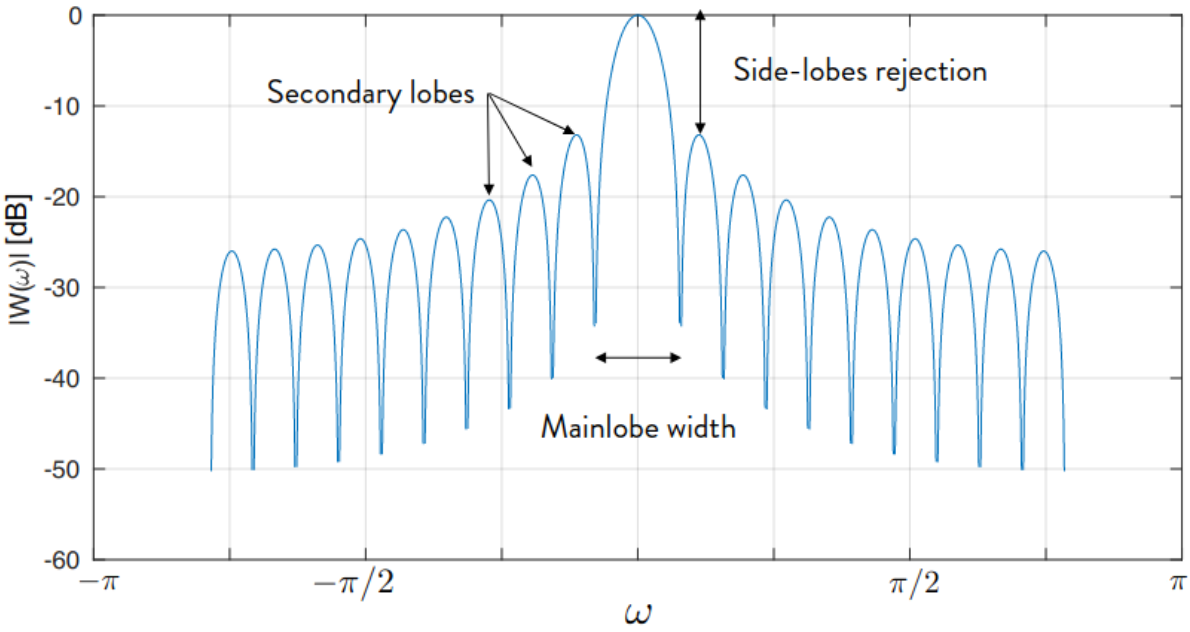
Windowing means multiplying in the time, so in the frequency we have convolution.

How to choose the window? Tradeoff

- As short as possible (in time) to minimize the cost of the FIR filter
- As narrow as possible in frequency to approach the ideal filter (a narrow in the time means very large in the frequency domain)

I want a short signal in the time, so an impulse, $W(f)$ should look like a $\delta(f)$, without considering the filter cost

- Its energy must be concentrated around $f = 0$
- $W(f)$ should decay fast as frequency increases



We want a big ration of $\frac{width(MainLobe)}{width(SecondaryLobes)}$. Every window is characterized by:

- Main-lobe width: it decreases as the window length increases
- Side-lobes rejection: ratio between the main-lobe peak and 1° secondary lobe peak [dB], if I want it high for example the blackman window is quite good
- Side-lobes roll off: asymptotic decay of the side-lobe peaks vs frequency octave [dB/octave] or frequency decade [dB/decade]

In matlab set cutoff = double of desired one, if I want cutoff of 0.5 I will put in matlab 1.

Filter order is always #filter samples-1, e.g. if filter samples = 64, filter order = 64-1 = 63.

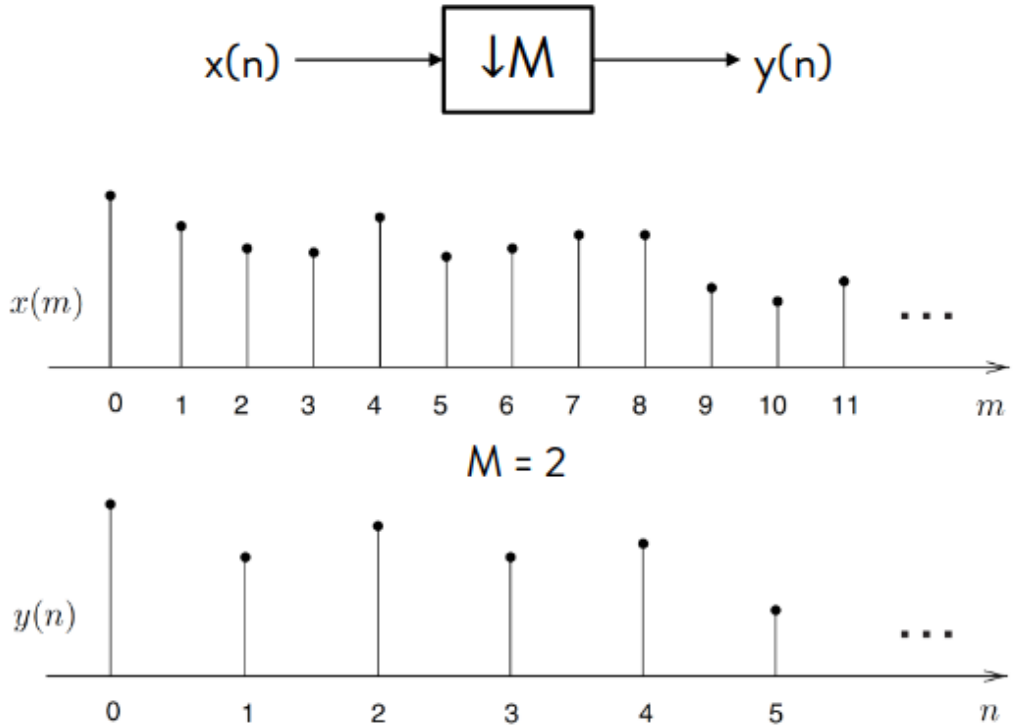
6.3 Aspects to remember

- **Phase is always correlated with a delay in the time**
- **Real filters always introduce a delay, a phase term, hence LPF's introduce delays**
- **FIR are always stable**, the numerator has greater grade than denominator
- **Only poles at the origin means FIR**
- **Denominator = 1 means FIR**
- **Maximum phase** will result in **jumps in the phase plot**
- **Minimum phase has the greatest part of energy of the filter concentrated in the first part of the temporal samples**, which means I am introducing very small delay
- **Allpass MUST be FIR**
- **A causal stable allpass system has always maximum phase zeros**, they are the reciprocal complex conjugate of the poles therefore the **phase has jumps =** 2π
- **DFT of sinusoid will have peak in the freq/normalized freq depending on representation of frequency domain (either** f_0 $[Hz]$ **or** $\tilde{f}_0$**) and another one symmetric w.r.t. Nyquist frequency, and in presence of consine with magnitude** A**, in DFT domain we will have a pulse of magnitude** $A/2$

7 Multirate Processing

7.1 Downsampling

$$y(n) = x(nM)$$



In the frequency domain, the downsampled is:

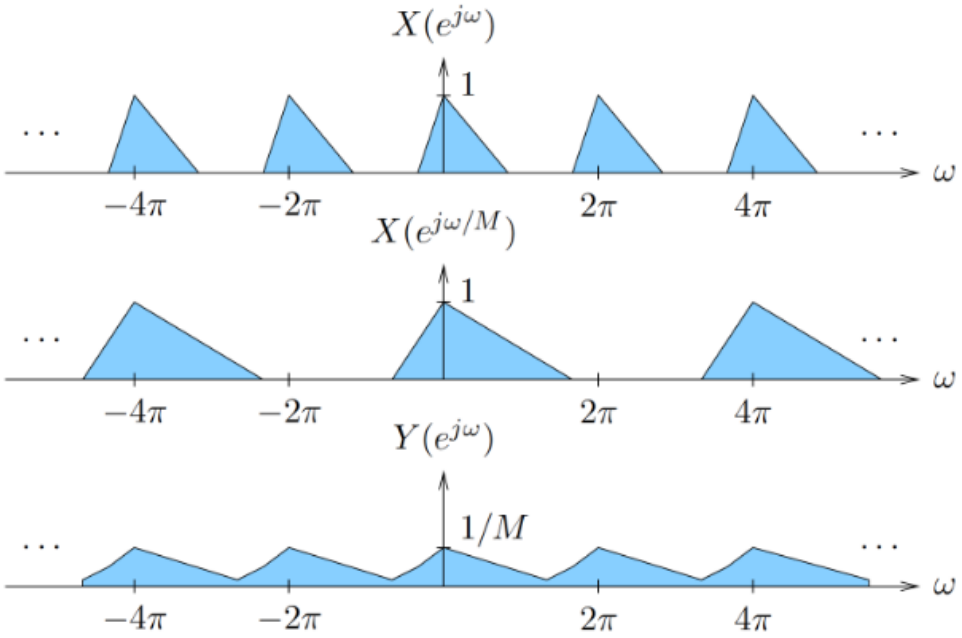
$$Y(f) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(\frac{f-k}{M}\right)$$

The DTFT if $y(n)$ is composed of copies of the DTFT of $x(n)$ **expanded by M and repeated with period 1** in normalized frequency (or F_s in Hertz, or 2π in angular frequencies)

The gain is reduced by a factor of M.

In practice the DFT of a downsampled sinusoid will be the pulses evaluated at $\tilde{\omega} \cdot M$ with a gain reduced by M: $A \to \frac{A}{M}$, remember to add the symmetric pulse for each!

7.1.1 Aliasing



Aliasing in frequency occurs if the DTFT of x(n) is not limited to 1/(2M) (or π/M, or Fs/(2M))

Note that in the picture it is not a real signal, not symmetric.

There isn't overlap if

$$BM \leq \frac{1}{2} \leftrightarrow B \leq \frac{1}{2M} \leftrightarrow B \leq \frac{\pi}{M} \leftrightarrow B \leq \frac{F_s}{2M}$$

Where B is the first right zero in frequency diagram, the original band of my signal. To avoid aliasing, we introduce a lowpass filter.

7.1.2 Decimation

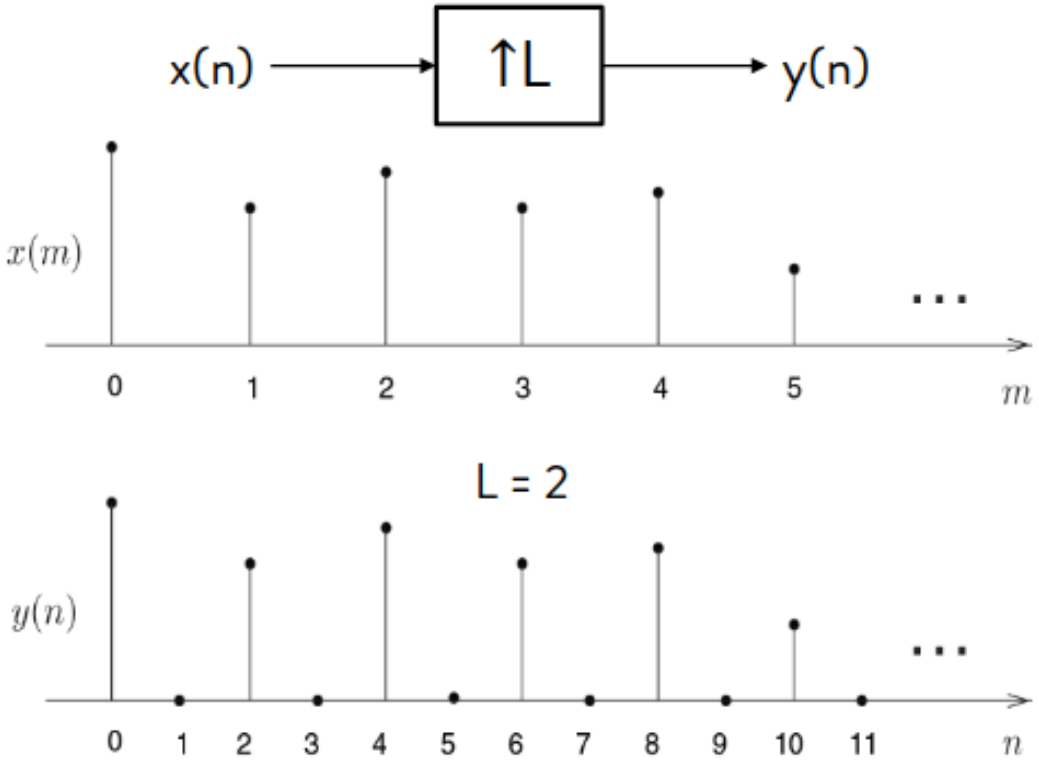
Put a lowpass filter, then downsample. We use a lowpass with cutoff of $\frac{1}{2M}$:

$$H(f) = \begin{cases} 1 & |f| \leq \frac{1}{2M} \\ 0 & \text{otherwise} \end{cases}$$

7.2 Upsampling

Upsampling of a factor L means to insert L -1 zeros between the input signal samples

$$y(n) = \begin{Bmatrix} x(n/L) & n = kL \\ 0 & \text{otherwise} \end{Bmatrix}$$



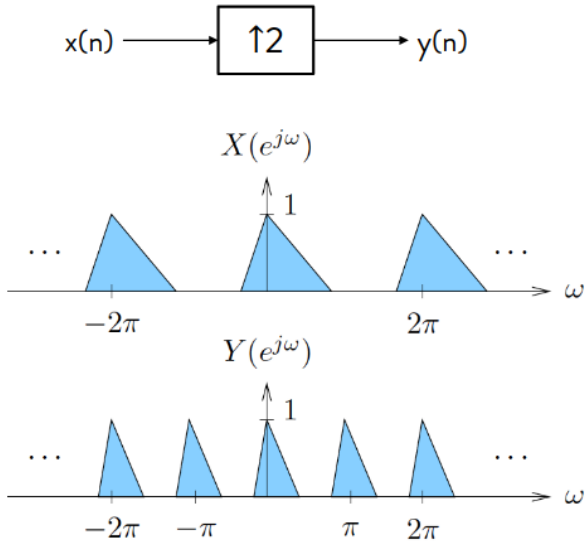
In the frequency domain:

$$Y(f) = X(fL)$$

Upsampling compresses the DTFT by a factor of L, **if we upsampled a sinusoid, its DFT will repeat the pulses every L, same goes for the symmetric pulses**.

In practice the DFT of an upsampled sinusoid will be the pulses evaluated at $\frac{\tilde{\omega}+2\pi k}{L}$, where $k = 0, 1, \cdots, L-1$, remember to add the symmetric pulse for each!

7.2.1 Replicas

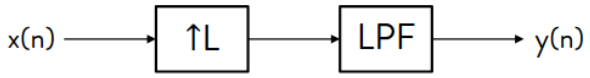


Spectral replicas do not overlap: upsampling just causes a compression of the spectrum, which has a new period of 1/L (or 2π/L, or Fs/(L))

7.2.2 Interpolation

Put a lowpass filter **after** upsampling. We use a lowpass with cutoff of $\frac{1}{2L}$, in this way we remove replicas:

$$H(f) = \begin{cases} L & |f| \leq \frac{1}{2L} \\ 0 & \text{otherwise} \end{cases}$$



After lpf, MULTIPLY AGAIN THE SIGNAL BY L

7.3 Rational sampling rate conversion: noninteger factor

Cascade interpolator wih a decimator

The lowpass dilter is:

$$H(f) = \begin{cases} L & |f| \leq \min\left\{\frac{1}{2L}, \frac{1}{2M}\right\} \\ 0 & \text{otherwise} \end{cases}$$

8 Matlab

```
close all
clearvars
clc
```

8.1 Sinusoid Signals

8.1.1 Sinusoid signal continuous, in time and in samples plot

A sinusoid in continuous domain is

$$x(t) = A \cdot \cos(2\pi f t + \phi)$$

Its sampled version is:

$$\begin{cases} x(n) = A \cdot \cos(2\pi \tilde{f} n + \phi) \\ \tilde{f} = \frac{f}{F_s} \end{cases}$$

Given:

$$\begin{cases} t = [0, 0.5] \\ F_s = 1000\,Hz = \frac{1}{T_s} & \text{sampling rate} \\ A = 0.8 & \text{amplitude} \\ f = 50\,Hz & \text{frequency, not normalized} \\ \phi = 30^\circ = \frac{\pi}{6} & \text{phase} \end{cases}$$

In matlab:

```
A = 0.8;
f = 50;
Fs = 1e3; %1000;
Ts = 1/Fs;
duration = 0.5:
t = 0:Ts:duration;
t = 0:1/Fs:duration;
phi = 30; % deg

% conversion
% phi_deg:phi_rad = 180: pi
phi_rad = phi*pi/180;

x = A*cos(2*pi*f*t + phi_rad);

%% Plot the signals as a function of time and as a function of samples

figure(1);
plot(t, x);
hold on;
plot(t, x1, '--');
title('Signals as a function of time');

figure(2);
plot(x); %plot(1:length(x), x);
hold on;
plot(1:length(x1), x1, '--');
title('Signals as a function of samples');
```

8.1.2 Sum of multiple signals continuous, lcm

Build a signal $x(n)$ as the sum of three different sinusoids at the normalized angular frequencies $\tilde{\omega}_1 = \pi/5$, $\tilde{\omega}_2 = \pi/8$, $\tilde{\omega}_3 = \pi/4$.
The sampling period is $T_s = 0.3$ seconds, and the signal is defined for t in $[0, 100]$ seconds.

We know that

$$\tilde{\omega} = 2\pi\tilde{f} = \frac{\omega}{F_s} = \omega T_s$$

```
T = .3;
end_duration = 100;
time = 0:T:end_duration;

% normalized omegas
omega_1_n = pi/5;
omega_2_n = pi/8;
omega_3_n = pi/4;

% omegas
omega_1 = omega_1_n / T;
omega_2 = omega_2_n / T;
omega_3 = omega_3_n / T;

% principle of superposition
A_1 = 1; % not specified so 1
A_2 = 1; % not specified so 1
A_3 = 1; % not specified so 1
x = A_1*cos(omega_1*time) + A_2*cos(omega_2*time) + A_3*cos(omega_3*
    time);
```

For the period of sinusoid in time and samples we have:

$$\frac{1}{f} = \frac{2\pi T_s}{\tilde{\omega}} \qquad \text{Period in seconds}$$

$$\frac{1}{f \cdot T_s} = \frac{1}{\tilde{f}} = \frac{2\pi}{\tilde{\omega}} = \frac{F_s}{f} \qquad \text{Period in samples}$$

```
%% compute the period of the sinusoids (seconds)

P_1 = 2*pi*T/omega_1_n;
P_2 = 2*pi*T/omega_2_n;
P_3 = 2*pi*T/omega_3_n;

% If you work with matrices: NB: here you have to put ./ otherwise
    MATLAB reports an error.
% P = 2*pi*T ./ Omega_n;s

%% compute the period of the sinusoids (samples)

P_1_samples = P_1 /T; % 10
P_2_samples = P_2 /T; % 16
P_3_samples = P_3 /T; % 8

%% period of x = lcm among (10, 16, 8) = 2^4 * 5

P_x_samples = lcm(lcm(P_1_samples, P_2_samples), P_3_samples);
P_x = P_x_samples * T;
```

8.1.3 Sum of multiple signals discrete, lcm

The signal $w(n)$ is defiend as the sum of 3 cosinusoidal signals, where the first signal has frequency f_0, the second $f_0/2$, the third $f_0/4$. Define $w(n)$ such that it repeats periodically every 10ms, knowing that it is sampled with $F_s = 1kHz$

```
Fs = 1e3;
P = 0.01;
P_samples = P*Fs;

% find the least common multiple among 1/f0, 2/f0, 4/f0
% --> 4/f0 = P_samples --> f0 = 4/P_samples
f0 = 4/P_samples;
f1 = f0/2;
f2 = f0/4;
N = 10*P_samples;
n = 0:N-1;
w = cos(2*pi*f0*n) + cos(2*pi*f1*n) + cos(2*pi*f2*n);
```

8.2 Discrete Signals (non-sinusoidal)

8.2.1 Shifting

Signal $x(n) = 0.8^n u(n)$ in $n = 1:20$. Generate the signals $y_1(n) = x(n-5)$ and $y_2(n) = x(n+5)$ always in $n = 1:20$

```
% Generate the signal x(n) = (0.8)^n u(n), n = 1:20
% Generate the signal y1(n) = x(n-5),  n = 1:20
% Generate the signal y2(n) = x(n+5), n = 1:20
n = 1:20;
x = (0.8).^n;

% initialize the two signals
y1 = zeros(size(n));
y2 = zeros(size(x));

y1 = circshift(x, 5); % x(n-5)
y1(1:5) = 0;

y2 = circshift(x, -5); % x(n+5)
y2(end-5:end) = 0;
```

8.2.2 Periodicity

Generate the signal $x(n) = u(n-5) - u(n-10)$ considering $n = 1:15$, then generate the periodic version $x_p(n)$ wih period $N = 15$ considering $n = 1:200$. Then plot the periodic signal considering only 8 periods.

```
% generate signal x(n)
N = 15; % period
n = 1:N;
x = zeros(1,N);
x(n >= 5 & n < 10) = 1;
stem(n, x);
hold on;

% Generate the periodic signal xp(n) with period N = 15,
% considering n = 1:200

n_max = 200;
N_p = ceil(n_max/N);
x_p = repmat(x,1,N_p);
x_p = x_p(1:n_max);

stem(1:15*8, x_p(1:15*8));
```

8.2.3 Convolution

Given:

$x(n) = [3,11,7,0,-1,4,2] \qquad n \in [-3,3]$
$h(n) = [2,3,0,-5,2,1] \qquad n \in [-1,4]$

Compute the convolution

```
n_x = -3:3;
n_h = -1:4;

x = [3, 11, 7, 0, -1, 4, 2];
h = [2, 3, 0, -5, 2, 1];
```

```
y = conv(x, h);
supp = n_x(1) + n_h(1):n_x(end) + n_h(end); % always like this

figure;
stem(supp, y);
```

8.2.4 Shifting through convolution

Given $x(n) = [3,11,7,0,-1,4,2]$, $n \in [-3,3]$, create $y(n) = x(n-5)$ $n \in [0,10]$ without using circshift or for loops

```
% y(n)=x(n-5)=x(n)*\delta(n-5)
n_x = -3:3;
x = [3, 11, 7, 0, -1, 4, 2];

n_h = 0:10;
delta_5 = zeros(size(n_h));
delta_5(n_h == 5) = 1;

% support of the convolution
n_conv = n_x(1) + n_h(1):n_x(end) + n_h(end);

y = conv(delta_5, x); % as commutative

% but we want only from 0:10
y = y(n_conv >= 0 & n_conv <= 10);
stem(0:10, y);
```

Create $y(n) = \frac{1}{3}\sum_{m=0}^{2} x(n-m)$ $n \in [0,10]$ without using circshift or for loops

```
% define the filter, which is 1/3 (delta(n) + delta(n-1) + delta(n-2))
n_h = 0:10;
h = zeros(size(n_h));
h(1:3) = 1/3;

% support of the convolution:
n_conv = n_x(1) + n_h(1): n_x(end) + n_h(end);

y = conv(x, h);

% we wanted y defined only for n = 0:10
y = y(n_conv>=0 & n_conv <= 10);
```

8.3 Z-transform

8.3.1 Convolution as product in frequency domain

We can use conv here as well, even though the z-transform of a convolution is the product, the coefficients of the resulting polynomial are the same of the coefficients of the time domain convolution

```
n_x = -2:2;
x = [3, 2, 1, 0, 1];

n_h = 0:4;
h = [1, 3, 2.5, 4, 2];

% support of the convolution -2:6
n_y = n_x(1) + n_h(1):n_x(end) + n_h(end);

y = conv(h, x); % as commutative

% Write the expression of H(z)
% 1\delta(n)+3\delta(n-1)+2.5\delta(n-2)+4\delta(n-3)+2\delta(n-4)
% H(z) = 1+3z^-1+2.5z^-2+4z^-3+2z^-4
H_z = h; % we just insert the coefficients

% analog way for X
% X(z)=3z^2+2z+1+z^-2
X_z = x;

% convolution in z-domain is just product, how do we do it?
% Just ignore polynomial product
% Due to convolution theorem and the fact that final support
% Y(z)=y0z^2+...+yNz^-6
% starts from 2 as support in -2:6.*-1 = 2:-6
% Where those coefficients are the same found from conv, so:
Y_z = y;
```

8.3.2 Filter cascade

$$H(z) = h_0 \cdot H_1(z) \cdot H_2(z) \cdots H_k(z)$$

$$\Downarrow$$

$$h(n) = h_0 \cdot h_1(n) * h_2(n) * h_3(n) * \cdots * h_k(n)$$

We express the filter cascade in the frequency domain

```
n_h = 0:4;
h = [1, 3, 2.5, 4, 2];
h_roots = roots(H_z);

% For h0 just put when support == 0
h_0 = h(n_h == 0); % [1 0 0 0 0]

% Now for every root 1-z_iz^{-1}
for r = 1:length(h_roots)
    h_r = [1, -h_roots(r)]; % coefficient of this sub H_i
    % The support will change, as we go on convolution contribtion
    if r == 1
        h_cascade = h_r;
        sup_cascade = [0, 1];
    else
        % convolve by the cascade
        h_cascade = conv(h_r, h_cascade);
        % new support of the cascade
```

```
            sup_cascade = sup_cascade(1) + 0:sup_cascade(end) + 1;
        end
    end

    % final cascade
    h_cascade = h_0 * h_cascade;
    y = conv(x, h_cascade);
```

8.3.3 Partial fract expansion of z-transform

Givan a LTI system:

$$H(z) = \frac{z^{-5} + z^{-4} - 3z^{-3} - 8z^{-2} + 7z^{-1} + 9}{z^{-3} - 2z^{-2} - z^{-1} + 2}$$

Find its partial fract expansion and then find $h(n)$ as the sum of the elementary filters found with the partial fract expansion, $n = 0:100$.

A reminder:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{D} a_k z^{-k}} = \sum_{k=1}^{D} \sum_{m=1}^{M} \frac{r_{k_m}}{(1 - p_k z^{-1})^m} + \underbrace{\sum_{k=0}^{N-D} c_k z^{-k}}_{N \geq D}$$

- $Z^{-1}\left\{\frac{r_{k_1}}{(1-p_k z^{-1})}\right\} = r_{k_1} \cdot (p_k)^n u(n)$
- $Z^{-1}\left\{\frac{r_{k_2}}{(1-p_k z^{-1})^2}\right\} = r_{k_2} \cdot (n+1)(p_k)^n u(n)$
- $Z^{-1}\left\{c_k z^{-k}\right\} = c_k \cdot \delta(n-k)$

```
    A_z = [2 -1 -2 1];
    B_z = [9 7 -8 -3 1 1];

    [r, p, c] = residuez(B_z, A_z); % residuez!!!

    n = 0:100;
    h_partial = zeros(size(n));

    h_partial(1:length(c))= c;

    for r_i = 1:length(r)
        h_el = r(r_i)*p(r_i).^n;
        h_partial = h_partial+h_el;
    end

    stem(n, h_partial);
```

8.3.4 From z transfer function to time domain

Givan a LTI system:

$$H(z) = \frac{z^{-5} + z^{-4} - 3z^{-3} - 8z^{-2} + 7z^{-1} + 9}{z^{-3} - 2z^{-2} - z^{-1} + 2}$$

Find $h(n)$ in $n = 0:100$

```
    A_z = [2 -1 -2 1];
    B_z = [9 7 -8 -3 1 1];
    n = 0:100;

    delta = zeros(1, length(n));
    delta(1) = 1;
    h = filter(B_z, A_z, delta);
```

8.3.5 Zero-pole plot

Given $y(n) = x(n) - bx(n-1) + ay(n-1)$, find the expression of $H(z)$ and plot it in the z-plane

```
B_z = [1 -b];
A_z = [1 -a];
zplane(B_z, A_z);
```

8.4 DFT

8.4.1 DFT H(f) (and H(z)) amplitude and phase

We are given:

$$y(n) = -2y(n-1) - y(n-2) + x(n) + 2\rho\cos(\theta)x(n-1) + \rho^2 x(n-2)$$

With $\rho = 0.9$, $\theta = \pi/8$ and sequence defined for n $[0, N-1]$, $N = 1000$

```
A_z = [1 2 1]; % denominator
B_z = [1 2 * rho * cos(theta) rho^2];
N = 1000;
[H_omega_2pi, omega_2pi] = freqz(B_z, A_z, N, 'whole');

% plot from -pi to pi, magnitude/amplitude NOT normalized
figure;
plot(omega_2pi - pi, fftshift(abs(H_omega_2pi)));
xlabel('\omega'); title('Amplitude'); grid

% amplitude in normalized frequencies in [0,1)
figure;
plot(omega/(2*pi), abs(Hap));

% amplitude in normalized frequencies in [-0.5,0.5)
plot(omega/(2*pi) - 0.5, fftshift(abs(H)));

% plot from -pi to pi, phase
figure;
plot(omega_2pi - pi, fftshift(angle(H_omega_2pi)));
xlabel('\omega'); title('Phase'); grid
```

8.4.2 DFT by hand with matrix

Given an h, compute the DFT as $H = W_N \cdot h$, sequence defined for n $[0, N-1]$, $N = 1000$

```
N = 1e3;
n = 0:N - 1;

A_z = [1 2 1]; % denominator
B_z = [1 2 * rho * cos(theta) rho^2];
%% h(n)
delta = zeros(size(n));
delta(1) = 1;
h = filter(B_z, A_z, delta);

W = exp(-1i * 2 * pi / N * (n' * n));
H_k = W * h';
```

8.4.3 DFT/FFT with function

```
H = fft(h); % careful, if filter has feedback (is IIR) this is not
    really DFT, we are computing a truncated version
h = ifft(H);
```

8.4.4 DFT plot: normalized frequency

Given FIR filter $h(n) = 1$, n in $[0, 19]$, visualize the DFT in normalized frequency $[-0.5, 0.5)$

```
N = 19;
n_h = 0:N - 1;
h = ones(size(n_h));

H_k = fft(h);

freq_axis = 0:N - 1; % k in [0,1/N...,(N-1)/N]
freq_axis = freq_axis / N; % or we can define it as 0:1/N:1 - 1/N;
freq_axis = freq_axis - 0.5; % just subtract to shift it, as f
    normalliy in [0, 1)
figure;
stem(freq_axis, fftshift(abs(H_k)));
grid
```

8.4.5 DFT plot: frequency (in Hz)

Given a sinusoidal with frequency $2Hz$ and duration $1.3s$, sampled with $F_s = 50$

```
f0 = 2;
duration = 1.3;
Fs = 50;
time = 0:1 / Fs:duration;

s = cos(2 * pi * f0 * time);
N = 50;

% FFT of the first 50 samples
S_f = fft(s(1:N));
freq_axis = 0:Fs / N:Fs - Fs / N;
figure;
stem(freq_axis, abs(S_f));

% DFT of entire signal
S_complete = fft(s);
N_complete = length(s);
freq_axis_complete = 0:Fs / N_complete:Fs - Fs / N_complete;
% if shift needed, do something like freq_axis = freq_axis-Fs/2;
figure;
stem(freq_axis_complete, abs(S_complete));
```

8.4.6 Zero padding

Same signal of before

```
% Pad the array h(n) with zeros until reaching 100 samples
% Evaluate the DFT of the padded h(n) and visualize it
% Are the two DFTs equal? Comment on the results

num_zeros = 100 - N;
h_pad = padarray(h, [0, num_zeros], 'post');
H_pad = fft((h_pad));

N_pad = length(h_pad);
freq_axis_pad = 0:N_pad - 1;
freq_axis_pad = freq_axis_pad / N_pad;
freq_axis_pad = freq_axis_pad - 0.5;

figure;
stem(freq_axis_pad, fftshift(abs(H_pad)));
grid
```

```
% Why are the two DFTs different?
% In the second case we have more samples (we added more samples)
% When we add zeros it means we are windowing a signal with a rectangle
% Windowing = multiplying in the time
% In the Fourier domain we are convolving with a sort of sinc
% As we have a delta convolved with a signal, we get the signal
% Also as we introduce sinc behavior, due to summation of tails
% of the sinc we don't have peaks exactly at expected points
```

8.4.7 DFT-periodic signals, sinusoidal

Given a sinusoidal with frequency $2Hz$ defined over $N = 50$ samples, sampled with $F_s = 50$ and a second sinusoid with frequency $2.2Hz$, equal duration and sampling rate, compute the DFT.

```
Fs = 50;
N = 50;
time = 0:1/Fs:1/Fs*(N - 1);
f0 = 2;
f1 = 2.2;
s0 = cos(2 * pi * f0 * time);
s1 = cos(2 * pi * f1 * time);
s = s0 + s1;

% FFT of s
S_f = fft(s);
freq_axis = 0:Fs / N:Fs - Fs / N;

figure; stem(freq_axis, abs(S_f));

% It will show two peaks, not what we expected, we wanted 4 peaks
% this is due to discontinuities
% to see the correct peaks use zero padding
% To pad N_pad=k*Period
% P_samples_1 = Fs/f1 = 25
% P_samples_2 = Fs/f0 = 50/2.2 which is NOT AN INTEGER
% but no worries, we can find the global period as well with lcm of

N_tot = lcm(25, 250);
num_pad = N_tot - N;
s_pad = padarray(s, [0, num_pad], 'post');

S_pad = fft(s_pad);
freq_axis_pad = 0:Fs / N_tot:Fs - Fs / N_tot;

figure; stem(freq_axis_pad, abs(S_pad));

% But the plot seems a single sinusoid, zero padding does not help here
    !
% Solution: increase #samples, that must be a multiple of the two
    periods
% P_samples_1 = Fs/f1 = 25
% P_samples_2 = Fs/f0 = 50/2.2

N = 250; % == lcm(25,250)
time = 0:1/Fs:1/Fs*(N - 1);
s0 = cos(2 * pi * f0 * time);
s1 = cos(2 * pi * f1 * time);
s = s0 + s1;
S_f = fft(s);
freq_axis = 0:Fs/N:Fs-Fs/N;

figure; stem(freq_axis, abs(S_f));
```

8.4.8 Cyclic convolution

```
x = [0 0 1 0 0]; n_x = 0:4;
y = [5 4 3 2 1]; n_y = 0:4;
% amount of samples over which we want to compute the cyclic conv
N = length(x);

% linear conv
z = conv(x, y);
n_z = n_x(1) + n_y(1):n_x(end) + n_y(end);

% cyclic conv
z_c = cconv(x, y, N);
stem(0:N - 1, z_mat, '--');
```

8.5 Filters

8.5.1 A and B from z-transform filter: conv

Given the filter:

$$H(z) = \frac{2(1-z^{-1})(1+0.5z^{-1})}{(1-0.8e^{j\pi/4}z^{-1})(1-0.8e^{-j\pi/4}z^{-1})}$$

Derive $A(z)$ and $B(z)$

```
B = conv(2*[1, -1], [1, 0.5]);
A = conv([1, -0.8*exp(1i*pi/4)], [1, -0.8*exp(-1i*pi/4)]);
```

8.5.2 Allpass filter

Define an allpass function

```
function [z_out, p_out, b_out, a_out] = allpass(b, a)
    % Input: b, a = numerator and denominator of H(z)
    % Output: z_out, p_out, b_out, a_out = zeros, poles, numerator,
    % denominator of the allpass transfer function related to H(z)

    %%  first possibility: create Hap as H(z) / tilde{H(z)}

    % do this for both numerator and denominator
    a_tilde = fliplr(conj(a));
    b_tilde = fliplr(conj(b));

    b_out = conv(b, a_tilde);
    a_out = conv(a, b_tilde);

    % usually, every filter is normalized such that a_0 = 1
    % (but it's just a scaling operation)
    b_out = b_out / a_out(1);
    a_out = a_out / a_out(1);

    z_out = roots(b_out);
    p_out = roots(a_out);
end
```

8.5.3 Stability check

For a filter to be stable, all poles must be inside the unit circle

```
stability_check = all(abs(p_ap) < 1);
% or (~ is logical complement)
stability_check = ~any(abs(p_ap) > 1);
```

8.5.4 Minimum and maximum phase check

For a filter to be maximum phase, at least one zero is outside the unit circle

```
function [filter_type] = typeOfFilter(b, a)
    % compute zeroes and poles
    zeroes = roots(b);
    poles = roots(a);

    % stability is related to poles only

    if any(abs(poles) > 1)
        filter_type = -1;
```

```
        else
            % the system is stable
            % check if it is minimum phase
            % also zeros must be inside the unit circle
            if any(abs(zeroes) > 1)
                % the system is not minimum phase
                filter_type = 0;
            else
                % the system is minimum phase
                filter_type = 1;
            end
        end
    end
```

8.5.5 Minimum phase-allpass decomposition

Properties of allpass: **Any rational causal stable system can be decomposed into the multiplication of a minimum phase system and an allpass system**

$$H(z) = H_{min}(z)H_{ap}(z)$$

Causal and stable, so all poles are inside the unit circle!
$H_{min}(z)$ contains:

- The poles and the zeros of $H(z)$ that lie inside the unit circle
- Zeros that are conjugate reciprocals of the zeros $H(z)$ lying outside the unit circle

$H_{ap}(z)$ contains:

- All the zeros of $H(z)$ that lie outside the unit circle
- Poles that are conjugate reciprocals of the zeros of $H(z)$ lying outside the unit circle

```
    B = [1, -1.98, 1.77, -0.17, 0.21, 0.34];
    A = [1, 0.08, 0.40, 0.27];

    % all-pass minimum phase decomposition
    zeroes = roots(B);
    poles = roots(A);

    % zeros and poles which are already minimum phase
    z_min = zeroes(abs(zeroes) <= 1);
    p_min = poles(abs(poles) <= 1);

    % check if there are zeroes outside the unit circle, maximum phase
    if any(abs(zeroes) > 1)
        % allpass decomposition
        z_ap = zeroes(abs(zeroes) > 1); % extract the maximum phase zeroes
        p_ap = 1 ./ conj(z_ap); % poles of allpass compensate for the zeros

        % minimum phase part
        % include other zeros = p_ap in the minimum phase filter
        z_min = [z_min; p_ap];

        % define the polynomials related to the all pass filter
        B_ap = poly(z_ap);
        A_ap = poly(p_ap);
    else %system is minimum phase
        B_ap = 1;
        A_ap = 1;
    end
```

```
% define the polynomials related to the minimum phase filter
B_min = poly(z_min);
A_min = poly(p_min);

% multipy by c_0 if want amplitude of allpass = 1
c_0 = sum(A_ap) / sum(B_ap);
B_ap = c_0 * B_ap;

% divide Hmin by c_0, so to compensate the multiplication before
B_min = B_min / c_0;

%% check on zplane
figure; zplane(B_min, A_min); grid; title("Minimum phase component");
figure; zplane(B_ap, A_ap); grid; title("Allpass component");
```

8.5.6 FIR filter to attenuate signal and conjugate zero/pole

You are given two zeros with absolute value equal to 2 in complex conjugate position, build a FIR filter in order to attenuate a signal with frequency $f_1 = \cdots$

```
z1 = 2*exp(1i*2*pi*f1);
z2 = conj(z1);

A = 1; % it is a FIR
% If we have two zeroes which are complex conjugate, we can
% always define the polynomial related to the zeros as
% 1 - 2*rho*cos(theta)z^{-1} + rho^2 z^{-2}
B = [1, -2*cos(2*pi*f1), 4];
y = filter(B,A,x);
```

8.5.7 Windowing

Given x as a consine wave sampled at $F_s = 8kHz$, defined from 0 to 1 second, amplitude 1.5, frequency $1.1kHz$, phase $45deg$. Compute y as x filtered with a low-pass filter with normalized cutoff frequency of 0.4 and 64 samples.

```
Fs = 8e3; duration = 1; A = 1.5; f0 = 1.1e3; phase = pi/4;
time = 0:1/Fs:duration;

N_filter = 64;
filter_order = N_filter - 1;
cutoff = 0.4;
cutoff_filter = cutoff * 2; % matlab cutoff always twice of desired

h = fir1(filter_order, cutoff_filter);
y = filter(h, 1, x); % as FIR, we put 1 as denominator
```

Apply a Hanning window to select the first 512 samples of y, then plot the DFT of the windowed y vs frequency in Hz, defined in $[-F_s/2, F_s/2]$

```
Nw = 512;
w = window(@hann, Nw); % or hann(Nw);
y_w = y(1:Nw) .* w'; % multiply element wise

%% we can try to exam some common windows behaviour
w1 = rectwin(64);
w2 = bartlett(64); % triangular
w3 = hamming(64);
w4 = blackman(64);
```

```
% open window tool
wvtool(w1, w2, w3, w4);

% compute DFT
Yw = fft(y_w);
freq_axis = 0:Fs/Nw:Fs-Fs/Nw;
freq_axis = freq_axis-Fs/2;
figure;
plot(freq_axis, Yw);
```

8.6 Multirate

8.6.1 Downsampling and decimation

Given a signal $x(n)$, downsample with factor $M = 4$, decimate it with a decimation factor of $M = 4$, using a FIR filter with order 64.

```
Fs = 500; f_0 = 50; f_1 = 100; duration = 3; time = 0:1 / Fs:duration;
x = cos(2 * pi * f_0 * time) + cos(2 * pi * f_1 * time);

M = 4;

% LPF
cut_off = 1 / (2 * M);
cut_off_filter = 2 * cut_off;
lpf = fir1(64, cut_off_filter);

x_f = filter(lpf, 1, x); % FIR

% decimate
x_decimated = x_f(1:M:end); % just specify a step size of M

% As original pulse in 50 Hz, after downsampling with M = 4
% we will find a pulse/peak in 4*50=200. Same goes for 100*4 = 400
% Without decimation we have aliasing at freq 100 (and its symmetric
    400)
% With lpf we will not have contributions of 100 and 400, as cutoff was
% 1/(2M) = 1/8. BUT as we are in Hz, 1/8*Fs=1/8*500 = 62.5, so any freq
% higher than that removed (100 > 62.5 removed, so its symmetric
    removed as well)
% BM < Fs/2 -> 50*4 < 250 yes, but 100*4 < 250 NO
```

8.6.2 Upsampling and interpolation

Given the downsampled signal from before, without lpf first, decimate it with a decimation factor of $M = 4$, using a FIR filter with order 64.

```
% Starting signal, from before
Fs = 500; f_0 = 50; f_1 = 100; duration = 3; time = 0:1 / Fs:duration;
x = cos(2 * pi * f_0 * time) + cos(2 * pi * f_1 * time);
M = 4;
x_downsampled = x_f(1:M:end); % starting signal, applied downsample
    without lpf

% Upsample it WITHOUT applyting lpf at the end
L = 4;
x1 = zeros(1, length(x_downsampled) * L);
x1(1:L:end) = x_downsampled;
```

Given the same downsampled signal, but this time with lpf applied, upsample with interpolation

```
% Starting signal, downsampled signal with lpf this time
cut_off = 1 / (2 * M);
cut_off_filter = 2 * cut_off;
lpf = fir1(64, cut_off_filter);
x_f = filter(lpf, 1, x);
x_decimated = x_f(1:M:end);

% Upsample it and then apply lpf
L = 4;
x2 = zeros(1, length(x_decimated) * L);
x2(1:L:end) = x_decimated;
```

```
lpf = fir1(64, 1/L); % cut_off = 2*(1/(2*L))=1/L
% remember to put the gain L in interpolation
x2 = L*filter(lpf, 1, x2); % MULTIPLY AGAIN BY L

% at the end, I WILL NOT OBTAIN THE ORIGINAL SIGNAL, i lost info
% as I downsampled to avoid aliasing (aliasing makes me lose
    information)
% At the end only in f0 we will find a sinusoidal contribution
% we also will find a delay at the beginning (many zeros at the
    beginning)
% This is because every filter in real life is a causal filter, so in
    the final
% output will be delay (phase related with delay) -> lpf's introduce
    delay
% x(n) -> [LPF*LPF] -> y(n)
% The convolution of those two LPF's introduce delay
% If we inspect the maximum peak in freq domain of the convolution/
    mutiplication
% as the peak is not in 0, but e.g. 65, a delay will be introduced
% To find the delay:
% [max_filter, filter_delay] = max(conv(lpf,lpf));
% filter_delay is the position of the max, so delay
% So to delete delay:
% actual_signal = cos(2*pi*f*(time(1:N)-time(filter_delay)));

[max_filter, filter_delay] = max(conv(lpf, lpf)); % lpf_up == lpf_down
    in this case
actual_signal = cos(2*pi*f_0 * (time(1:N) - time(filter_delay)));
```

8.6.3 Rational sampling rate conversion: noninteger factor

Given the signal $x(t) = A_1\cos(2\pi f_1 t) + A_2\cos(2\pi f_2 t)$, create the signal $x(n)$ as $x(t)$ from 0 to 0.5 seconds, sampled at $F_s = 8000Hz$, $A_1 = 0.7$, $A_20+ = 0.5$, $f_1 = 1800Hz$, $f_2 = 3600Hz$. Then create the signal $y(n)$ by resampling $x(n)$ with $6000Hz$, suing $N = 64$ filter samples;

```
duration = 0.5; Fs = 8e3; time = 0:1/Fs:duration;
A1 = 0.7; f1 = 1800;
A2 = 0.5; f2 = 3600;

x = A1*cos(2*pi*f1*time) + A2*cos(2*pi*f2*time);

% 8000 -> 24000 -> 6000
L = 3; M = 4; % or use [L, M] = rat(Fs_new / Fs);

% Upsample
x_upsampled = zeros(1, length(x) * L);
x_upsampled(1:L:end) = x;

% LPF with cutoff of min(1/(2L),1/(2M))
cutoff = min([1/(2*L), 1/(2*M)]);
cutoff_filter = 2*cutoff;
h = fir1(64-1, cutoff_filter);
x_f = L*filter(h, 1, x_upsampled);

% Downsample
y = x_f(1:M:end);
```

8.7 Functions Recap

```
%% Shifting discrete signals, a positive value will shift to the right.
    Circular, if shifting right by n, first n values will become the last n
    values
circshift([1 2 3 4 5], 2);
% [4 5 3 2 1]


%% Periodic sequence generation. The first paramater is the matrix, the
    second one is the rows repetition, the third is the cols repetition
repmat([1 2 3],1,3);
% [1 2 3 1 2 3 1 2 3]
repmat([1 2 3],2,3);
% [1 2 3 1 2 3 1 2 3;
%  1 2 3 1 2 3 1 2 3]


%% Returns the convolution of vectors u and v. If u and v are vectors of
    polynomial coefficients, convolving them is equivalent to multiplying
    the two polynomials.
conv([1 1],[1 1]); % (1+x)(1+x)
% [1 2 1], which is (1+x)(1+x)=1+2x+x^2


%% Roots of a polynomial
roots([1 0 1]); %1+x^2
% [0.0000+1.0000i, 0.0000-1.0000i]'


%% Partial fract expansion
A_z = [2 -1 -2 1];
B_z = [9 7 -8 -3 1 1];
[r , p , c ] = residuez(B_z, A_z);


%% Apply the filter of numerator B_z, denominator A_z to input delta
h = filter(B_z, A_z, delta);


%% z-plane, zero-pole plot
zplane([1 -0.5],[1 -0.2]);
% A zero in 0.5 and a pole in 0.2


%% DFT magnitude and phase plot
[H_omega_2pi, omega_2pi] = freqz(B_z, A_z, N, 'whole');
% plot from -pi to pi
plot(omega_2pi - pi, fftshift(abs(H_omega_2pi)));
plot(omega_2pi - pi, fftshift(angle(H_omega_2pi)));


%% DFT/FFT and inverse
H = fft(h);
h = ifft(H)


%% Pad an array
padarray([1 1], [0, 3], 'post');
% [1 1 0 0 0]
padarray([1 1], [2, 1], 'post');
% [1 1 0; 0 0 0; 0 0 0]
```

```
%% Cyclic convolution
cconv(x, y, N);


%% Invert an array
fliplr([1 2 3 4])
% [4 3 2 1]


%% Get numerator and denominator of a fraction
[L, M] = rat(6000 / 8000);
% [3, 4]
```

9 Exercises

9.1 20220217-1

A signal $x[n]$ is filtered using a filter $h[n]$ that is the cascade of two filters, $h_1[n] = \{1,-1\}$ and $h_2[n] = \{1,0,4\}$

9.1.1 Temporal sequence, convolution and z-transform

1. [3 pts] Find the temporal sequence of the filter $h[n] = \cdots$ and its z-transform $H(z) = \cdots$

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

The convolution:

- Flip the second term $h(n) \to h(-k)$
- Shift h by adding n (if positive shift to right, if negative to left) $h(n-k)$
- For output $y(n)$ sum all contributions of $x(k)$ and the shifted flipped h

Last step similar to scalar product. **The length of the convolution is the sum of the two signals-1 and the support (x-axis from when the signals start to differ from 0) is the sum of the supports until the final support of the two**

So in our case the length of the convolution is $2+3-1=4$. We first flip h_2 then

$$h[n=0] = \left\{ \begin{matrix} & & 1 & -1 \\ \cdot & \cdot & \cdot & \cdot \\ 4 & 0 & 1 & \end{matrix} \right\} = 1$$

$$h[n=1] = \left\{ \begin{matrix} & 1 & -1 \\ \cdot & \cdot & \cdot \\ 4 & 0 & 1 \end{matrix} \right\} = -1$$

$$h[n=2] = \left\{ \begin{matrix} 1 & -1 & \\ \cdot & \cdot & \cdot \\ 4 & 0 & 1 \end{matrix} \right\} = 4$$

$$h[n=3] = \left\{ \begin{matrix} 1 & -1 & & \\ \cdot & \cdot & \cdot & \cdot \\ & 4 & 0 & 1 \end{matrix} \right\} = -4$$

So $h[n] = \{1,-1,4,-4\}$. Its z-transform:

$$H(z) = 1 - z^{-1} + 4z^{-2} - 4z^{-3}$$

Alternatively, we apply the distributive property of the convolution. Consider:

$$h_2[n] = \delta[n] + 4\delta[n-2]$$

So

$$h[n] = h_1[n] * \delta[n] + h_1[n] * 4\delta[n-2] = h_1[n] + 4h_1[n-2] =$$
$$\{1,-1,0,0\} + 4\{0,0,1,-1\} = \{1,-1,4,-4\}$$

9.1.2 Pole-zero plot and magnitude

2. [3 pts] Represent the pole-zero plot of $H(z)$ and its magnitude.

We must rewrite $H(z)$ into a fraction:

$H(z) = 1 - z^{-1} + 4z^{-2} - 4z^{-3} = 1 - \frac{1}{z} + 4\frac{1}{z^2} - 4\frac{1}{z^3} =$
$= \cdots \frac{1}{z^3}\left(z^3 - z^2 + 4z - 4\right)$

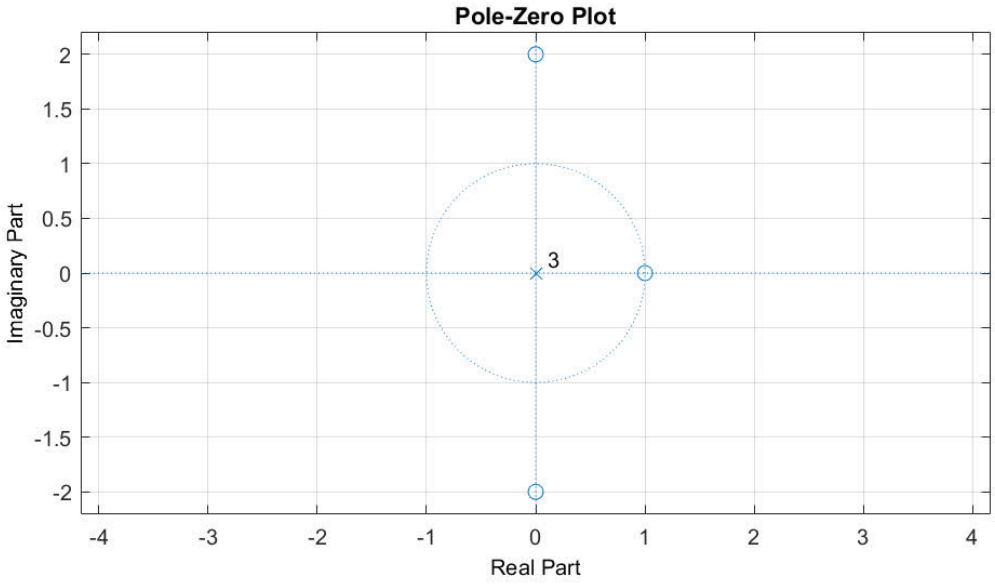
	1	-1	4	-4
1		1	0	4
	1	0	4	0

$= \cdots \frac{(z-1)\left(z^2+4\right)}{z^3}$

So we have:

- **Three poles** in 0
- **A zero** in 1
- **Two complex conjugate zeros** in $\pm 2j$

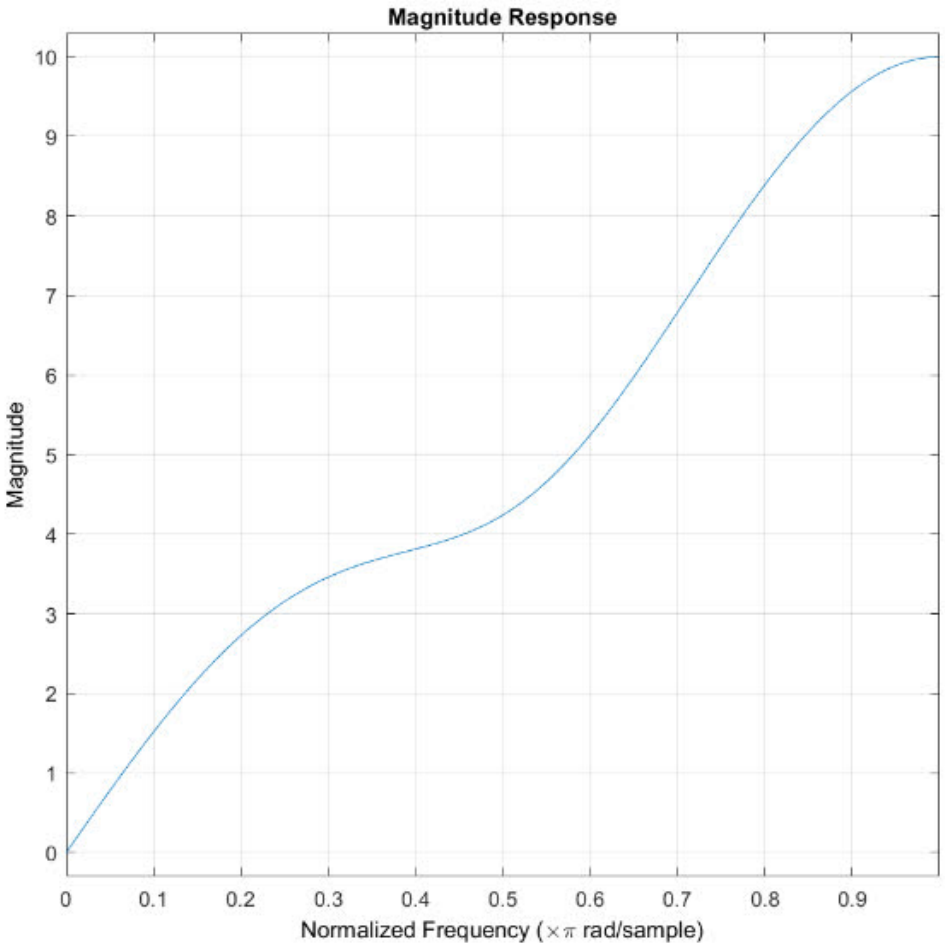
We notice that the filter is FIR, it has only poles at the origin.



For the amplitude we just look at the pole-zero plot and multiply all origin-zero distances while we divide for all origin-pole distances. As all the poles are at the origin, their distance from a point on the circle will always be 1. We plot from 0 to π as $-\pi$ to 0 is symmetric, so normalized in 0 to 1. As $z = \rho e^{j2\pi f} = e^{j\omega}$:

- $\omega = 0$, $z = 1 \rightarrow |H(z)| = 0$
- $\omega = \frac{\pi}{2}$, $z = j \rightarrow |H(z)| = 3\sqrt{2} \sim 4.24$
- $\omega = \pi$, $z = -1 \rightarrow |H(z)| = 2\sqrt{5}\sqrt{5} = 10$

Reaching a zero will lead to a minimum



9.1.3 Define minimum phase filter with fixed amplitude

[2 pts] In case of it is a maximum phase filter provide its minimum phase version with the same magnitude response otherwise, if it is a "minimum phase filter" provide its maximum phase version.

The filter is maximum phase, the two zeros are outside the circle

$$H(z) = 1 - z^{-1} + 4z^{-2} - 4z^{-3} = (z^{-1} - 1)(-4z^{-2} - 1) = (1 - z^{-1})(1 + 4z^{-2})$$

$$\begin{cases} \rho^2 = 4 \\ 2\rho\cos\theta = 0 \end{cases} \Rightarrow \begin{cases} \rho = 2 \\ \theta = \pm\frac{\pi}{2} \end{cases}$$

Write the G version of those that introduce maximum phase:

$$H_m(z) = (1 - z^{-1})(4 + z^{-2}) = 4 - 4z^{-1} + z^{-2} - z^{-3}$$

9.1.4 Output samples: convolution

[3 pts] Working in the time domain evaluate the output $y[n]$ of the signals $x_a[n] = \{1,-1,1,-1\}$ and $x_b[n] = \{1,1,1,1\}$ with filter $h[n]$.

$$h[n] = \delta(n) - \delta(n-1) + 4\delta(n-2) - 4\delta(n-3)$$

$n=$	0	1	2	3	4	5	6
$x_a[n]$	1	-1	1	-1			
$-x_a[n-1]$	0	-1	1	-1	1		
$4x_a[n-2]$	0	0	4	-4	4	-4	
$-4x_a[n-3]$	0	0	0	-4	4	-4	4
$y_a[n]$	1	-2	6	-10	9	-8	4

$n=$	0	1	2	3	4	5	6
$x_b[n]$	1	1	1	1			
$-x_b[n-1]$	0	-1	-1	-1	-1		
$4x_b[n-2]$	0	0	4	4	4	4	
$-4x_b[n-3]$	0	0	0	-4	-4	-4	-4
$y_b[n]$	1	0	4	0	-1	0	-4

9.2 20220217-2

A signal $x(t) = \sin(2\pi 100t) + 2\cos(2\pi 50t)$ is sampled at $400Hz$ and then downsampled by an order of $M = 3$

9.2.1 Downsampling without LPF, DFT of a sinusoid: peaks and pulses

[5pts] What is the output in case of no low pass (antialiasing) filter is adopted? Depict the magnitude of the output in the range $0 - \pi$ (normalized frequencies).

Suppose we have N samples, the sampled signal is

$$x[n] = 2\cos\left(\frac{\pi}{4}n\right) + \sin\left(\frac{\pi}{2}n\right)$$

Computing its DFT, as it is a sinusoid, we will have 4 pulses, 2 at $\frac{\pi}{2}$ and its symmetric w.r.t. the Nyquist frequency, 2 at $\frac{\pi}{4}$ and its symmetric. As the contribution on $\frac{\pi}{2}$ (the cosine) is twice the contribution of the sine, the pulse at $\frac{\pi}{2}$ will be halve the size of the other.

$$|X(\omega)| = \delta\left(\omega - \frac{\pi}{4}\right) + \delta\left(\omega - 2\pi + \frac{\pi}{4}\right) + \frac{1}{2}\delta\left(\omega - \frac{\pi}{2}\right) + \frac{1}{2}\delta\left(\omega - 2\pi + \frac{\pi}{2}\right)$$

After downsampling:

$$x_d[n] = x[nM] = 2\cos\left(\frac{3\pi}{4}n\right) + \sin\left(\frac{3\pi}{2}n\right)$$

Its DFT will be:

$$X_d(\omega) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(\frac{\omega - k}{M}\right)$$

Or alternatively write the DFT of $x_d[n]$ normally and add a $\frac{1}{3}$ factor

$$|X_d(\omega)| = \frac{1}{3}\left[\delta\left(\omega - \frac{3\pi}{4}\right) + \delta\left(\omega - 2\pi + \frac{3\pi}{4}\right) + \frac{1}{2}\delta\left(\omega - \frac{3\pi}{2}\right) + \frac{1}{2}\delta\left(\omega - 2\pi + \frac{3\pi}{2}\right)\right]$$

$$|X_d(\omega)| = \frac{1}{3}\left[\delta\left(\omega - \frac{3\pi}{4}\right) + \delta\left(\omega - \frac{5\pi}{4}\right) + \frac{1}{2}\delta\left(\omega - \frac{3\pi}{2}\right) + \frac{1}{2}\delta\left(\omega - \frac{\pi}{2}\right)\right]$$

As we are only interested in the range $[0, \pi]$, only the following pulses remain:

$$|X_d(\omega)| = \frac{1}{3}\left[\delta\left(\omega - \frac{3\pi}{4}\right) + \frac{1}{2}\delta\left(\omega - \frac{\pi}{2}\right)\right]$$

9.2.2 Cutoff for LPF

Suggests a possible low pass filter describing its behavior and the cutoff frequency justifying the choice: what will be the offect on the output signal?

We introduce a cutoff of $\omega = \frac{\pi}{M} = \frac{\pi}{3}$ in order to avoid aliasing.

9.3 20220126-1

We need to apply the first derivative ($h[n] = \{1, -1\}$) in real-time to a digital signal $x[n] = \{1, 2, 0, -1, 0, 1, \cdots\}$ working in the frequency domain on blocks of 4 samples.

9.3.1 DFT

[4pts] Describe how to get the DFT for each block of the signal and for the filter according to the requested task.

Just use the W matrix of dimension 4:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

So the DFT of the filter will be:

$$H[k] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1+j \\ 2 \\ 1-j \end{bmatrix}$$

9.3.2 Overlap and save with DFT

[7pts] Using the overlap and save algorithm, describe how to find the output for the given signal, provide all the intermediate numerical results and the steps to obtain the final result.

Blocks of 4 samples, for overlap and save fot the first block we add $\underbrace{P}_{\text{len(h)}} - 1 = 1$ leading zeros (the overlap):

$x_1[n] = \{0, 1, 2, 0\}$

$x_2[n] = \{0, -1, 0, 1\}$

$x_3[n] = \{1, 0, 0, 0\}$ overlap last block

We have to work in the DFT domain:

$$Y_1[k] = X_1[k] \times H[k] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1+j \\ 2 \\ 1-j \end{bmatrix} = \cdots = \begin{bmatrix} 0 \\ -1-3j \\ 2 \\ -1+3j \end{bmatrix}$$

$$Y_2[k] = X_2[k] \times H[k] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1+j \\ 2 \\ 1-j \end{bmatrix} = \cdots = \begin{bmatrix} 0 \\ -2+2j \\ 0 \\ -2-2j \end{bmatrix}$$

...

Compute the inverse of W (its transpose with complex conjugates), return to time domain, throw the first $P-1=1$ samples, we will get:

$y_1[n] = \{\cdot, 1, 1, -2\}$

$y_2[n] = \{\cdot, -1, 1, 1\}$

$y_3[n] = \{\cdot, -1, 0, 0\}$

$$y[n] = \{1, 1, -2, -1, 1, 1, -1, 0, 0\}$$

9.3.3 Overlap and save in time domain

[2pts] Check that working in the time domain you would get the same result.

Consider the filter $h[n] = \{1, -1\}$ As the filter $h[n]$ has length $P = 2$, we need to add $P-1=1$ "0"s at the beginning to account for the circular convolution effect

$$L = N + P - 1 = 3 + 2 - 1 = 4$$

$x_1[n] = \{0, 1, 2, 0\}$

$x_2[n] = \{0, -1, 0, 1\}$

$x_3[n] = \{1, 0, 0, 0\}$

$$x[n] * h[n] = x[n] * \delta[n] + x[n] * -\delta[n-1]$$

$n=$	0	1	2	3
$x_1[n]$	0	1	2	0
$-x_1[n-1]$	0	0	-1	-2
$x_2[n]$	0	-1	0	1
$-x_2[n-1]$	0	0	1	0
$x_3[n]$	1	0	0	0
$-x_3[n-1]$	0	-1	0	0

We throw the first two as $P-1=1$, keep the next $L-1=3$, so:

$\Rightarrow y_1[n] = \{\cdot \quad 1 \quad 1 \quad -2\}$
$\Rightarrow y_2[n] = \{\cdot \quad -1 \quad 1 \quad 1\}$
$\Rightarrow y_3[n] = \{\cdot \quad -1 \quad 0 \quad 0\}$

And the final result is

$$y[n] = \{y_1\ y_2\ y_3\} = \{1, 1, -2, -1, 1, 1, -1\}$$

9.3.4 Overlap and Add

Overlap and add example

$$x[n] = \{1, 2, 0, -1, 0, 1, \cdots\}$$

We would have:

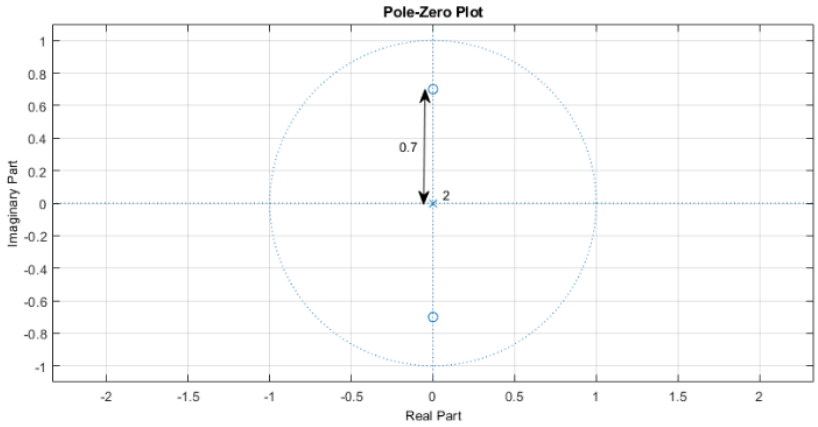
$x_1[n] = \{1, 2, 0, -1\}$

$x_2[n] = \{0, 1, 0, 0\}$

Then we compute linear convution normally, then consider overlapping tails (shift every block by length of block $L = 4$)

9.4 20211108-1

A digital filter $H_m(z)$ is a minimum phase filter with the following pole-zero diagram:



9.4.1 Zero-pole to equation in z

[3pts] Provide the equation of $H_m(z)$ in z

We use:

$$1 - 2\rho\cos(\theta) + \rho^2 \qquad \rho = 0.7,\ \theta = \pm\frac{\pi}{2}$$

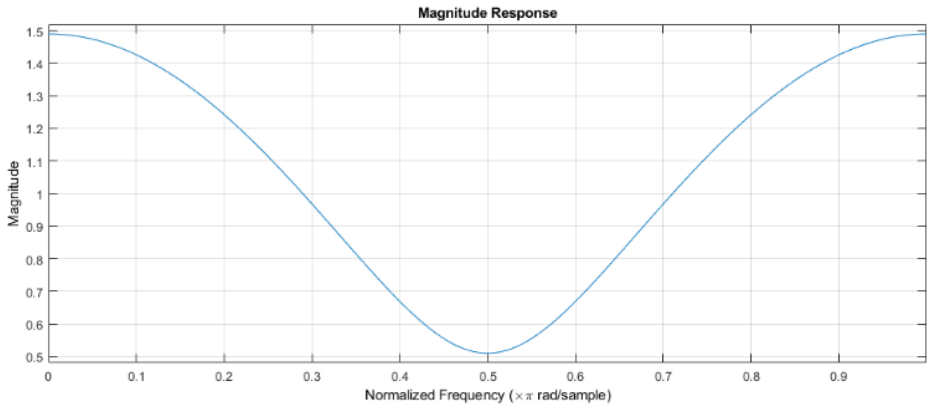
$$H_m(z) = 1 + 0.49z^{-2}$$

9.4.2 Magnitude and phase plot by hand

[3pts] Provide an approximate representation of its magnitude and phase response finding the exact values for $\omega = 0,\ \pi/2,\ \pi$

Magnitude:

- $\omega = 0 \rightarrow z = 1 \rightarrow H_m(1) = 1.49$
- $\omega = \pi/2 \rightarrow z = j \rightarrow H_m(j) = 0.51$
- $\omega = \pi \rightarrow z = -1 \rightarrow H_m(-1) = 1.49$



$$H_m(z) = 1 + 0.49z^{-2}$$

Phase:

- $\omega = 0 \rightarrow z = 1 \rightarrow 0$ due to opposite contributions of zeros
- $\omega = \pi/2 \rightarrow z = j \rightarrow H_m(j) = 0.51$, whose angle is 0 as well
- $\omega = \pi \rightarrow z = -1 \rightarrow 0$ due to opposite contributions of zeros

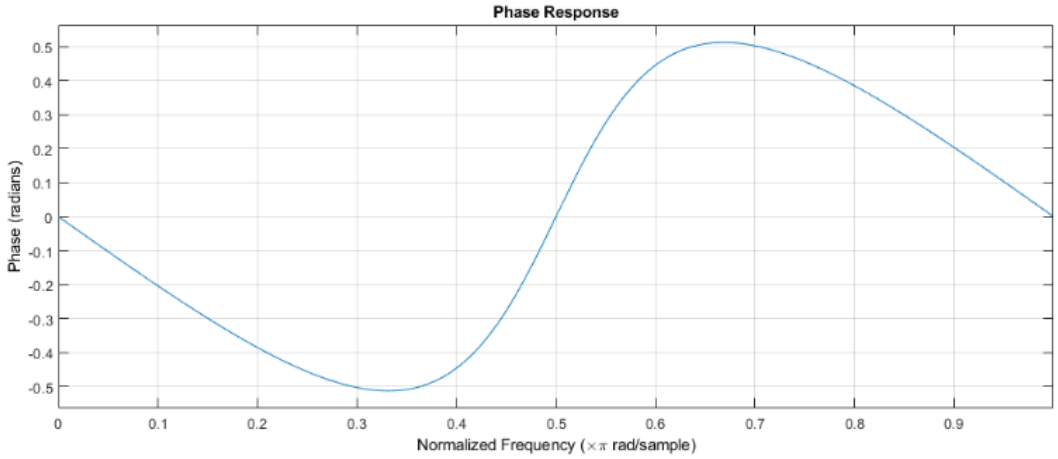
Additional values for plotting:

- $\omega = \pi/4 \rightarrow z = \frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$

$$H_m(z) = 1 - 0.49j \rightarrow \angle(1 + 0.49j) \sim -26^\circ \sim -0.45$$

- $\omega = 3\pi/4 \rightarrow z = -\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$

$$H_m(z) = 1 + 0.49j \rightarrow \angle(1 - 0.49j) \sim 26^\circ \sim 0.45$$



9.4.3 Define maximum phase filter with fixed amplitude

[3pts] Provide the equation of $H_M(z)$ with the proper gain

We write the G version:

$$H_M(z) = z^{-N} H_m^*(z^{-1}) = 0.49 + z^{-2}$$

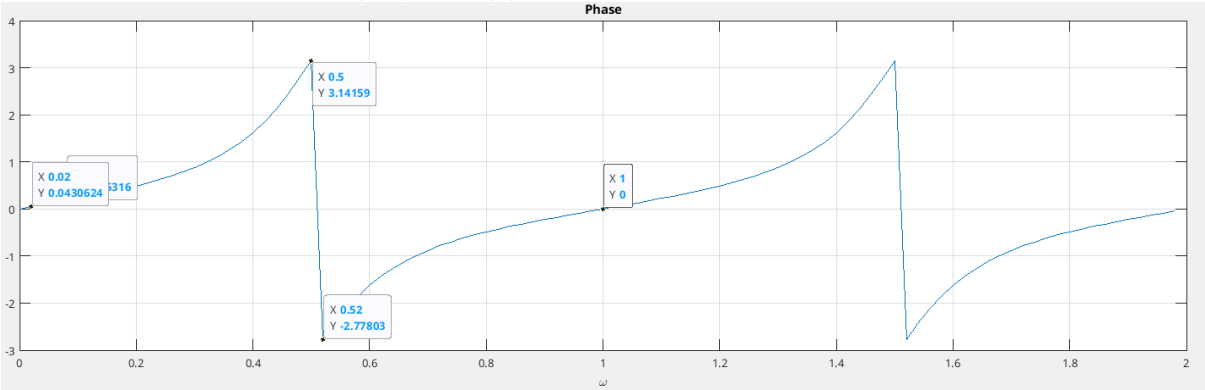
9.4.4 Allpass filter

[3pts] Provide the equation of $H_A(z)$, what will be the effect on the phase at $\omega = 0,\ \pi/2,\ \pi$?

$$H_A(z) = \frac{H_m(z)}{H_M(z)} = \frac{1+0.49z^{-2}}{0.49+z^{-2}}$$

The phase will The phase:

- $\omega = 0 \rightarrow z = 1 \rightarrow H_A(1) = \frac{1.49}{0.49}$ whose angle is 0
- $\omega = \pi/2 \rightarrow z = j \rightarrow H_A(j) = \frac{0.51}{-0.51} = -1$, whose angle is π
- $\omega = \pi \rightarrow z = -1 \rightarrow H_A(-1) = H_A(1)$ whose angle is 0



9.4.5 Difference equation

[3pts] Provide the finite difference equation $y[n] = \cdots$ of the $H_A(z)$

$$H_A(z) = \frac{1+0.49z^{-2}}{0.49+z^{-2}}$$

$$0.49y[n] + y[n-2] = x[n] + 0.49x[n-2]$$

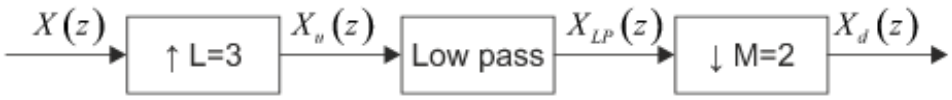
$$y[n] = -\frac{1}{0.49}y[n-2] + \frac{1}{0.49}x[n] + x[n-2]$$

9.5 20211108-2

A continuous signal composed of two sinusoids $y(t) = 10\cos(2\pi \cdot 50t) + 5\sin(2\pi \cdot 75t)$ is sampled at $200Hz$ with an ideal sampler

9.5.1 Upsample by noninteger detailed pipeline

[2pts] We need to change its sample rate to 300 samples per second: describe, to processing chain in order to get the desired result from a quantitative viewpoint (providing values used in each step)



The sampled signal is:

$$x[n] = 10\cos\left(\frac{\pi}{2}n\right) + 5\sin\left(\frac{3\pi}{4}n\right)$$

We know that the DFT of a sinusoid will introduce two symmetric pulses at $\pm\tilde{f}_0$, so in this case we will have 4 pulses. On the next equations we omitted the correct magnitude of each pulse, each term should be multiplied by

$$A = \underbrace{\frac{10}{2}}_{\text{In presence of cos, magnitude of pulse is } A/2} = 5$$

$$X(f) = \delta\left(f - \frac{\pi}{2}\right) + \frac{1}{2}\delta\left(f - \frac{3\pi}{4}\right) + \underbrace{\delta\left(f - \left(2\pi - \frac{\pi}{2}\right)\right) + \frac{1}{2}\delta\left(f - \left(2\pi - \frac{3\pi}{4}\right)\right)}_{\text{symmetric pulses}}$$

The DFT will be $X_u(f) = X(fL)$, or alternatively we know that the peaks are in $\frac{\tilde{\omega}+2\pi k}{L}$

$$X_u(f) = \begin{cases} \delta\left(f - \frac{\pi/2}{3}\right) + \delta\left(f - \frac{\pi/2+2\pi}{3}\right) + \delta\left(f - \frac{\pi/2+4\pi}{3}\right) \\ +\frac{1}{2}\delta\left(f - \frac{3\pi/4}{3}\right) + \frac{1}{2}\delta\left(f - \frac{3\pi/4+2\pi}{3}\right) + \frac{1}{2}\delta\left(f - \frac{3\pi/4+4\pi}{3}\right) \\ + \underbrace{\cdots\cdots\cdots}_{\text{symmetric pulses}} \end{cases}$$

$$X_u(f) = \delta\left(f - \frac{\pi}{6}\right) + \delta\left(f - \frac{5\pi}{6}\right) + \delta\left(f - \frac{9\pi}{6}\right) + \frac{1}{2}\left[\delta\left(f - \frac{\pi}{4}\right) + \delta\left(f - \frac{11\pi}{12}\right) + \delta\left(f - \frac{19\pi}{12}\right)\right] + \underbrace{\cdots\cdots\cdots}_{\text{symmetric pulses}}$$

We can limit our analysis to $[0, \pi]$, so $X_u(f)$ becomes:

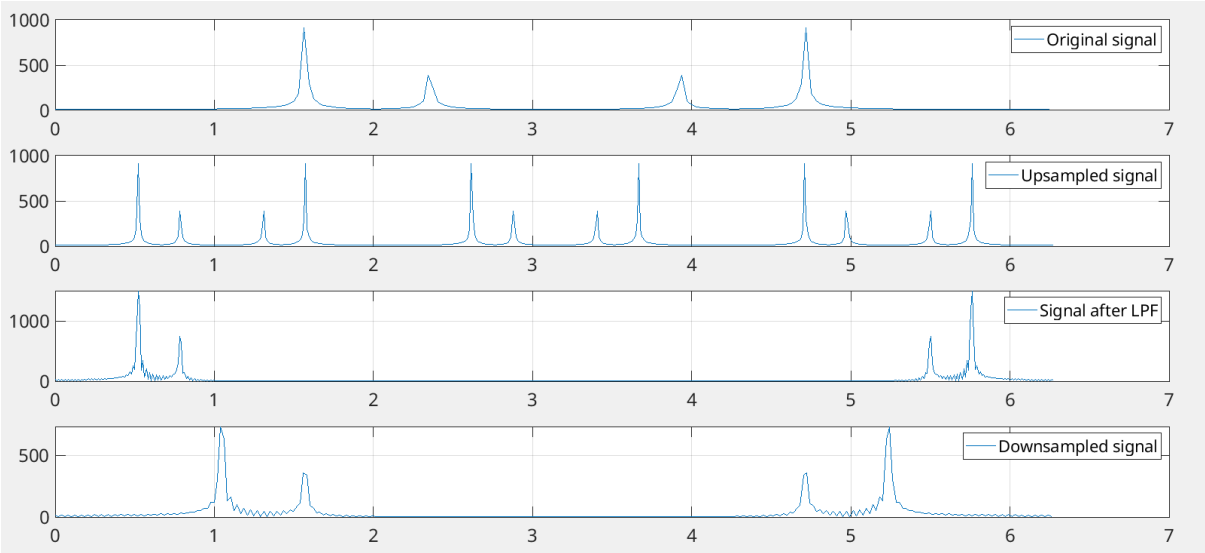
$$X_u(f) = \delta\left(f - \frac{\pi}{6}\right) + \delta\left(f - \frac{3\pi}{6}\right) + \delta\left(f - \frac{5\pi}{6}\right) + \frac{1}{2}\left[\delta\left(f - \frac{\pi}{4}\right) + \delta\left(f - \frac{5\pi}{12}\right) + \delta\left(f - \frac{11\pi}{12}\right)\right]$$

We then apply a LPF with cutoff of $\min\left[\frac{\pi}{3}, \frac{\pi}{2}\right] = \frac{\pi}{3}$, which will keep all pulses before the cutoff and remove all pulses after it:

$$X_{LP}(f) = \delta\left(f - \frac{\pi}{6}\right) + \frac{1}{2}\delta\left(f - \frac{\pi}{4}\right)$$

At the end we apply downsampling with $M = 2$, so the position of each pulse will be multiplied by 2

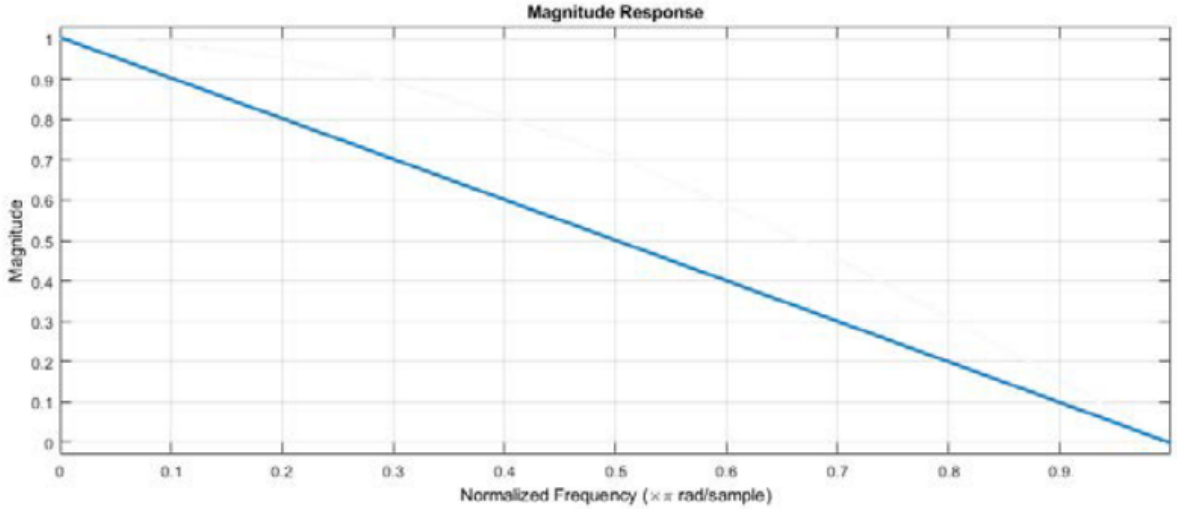
$$X_d(f) = \frac{1}{2}\left[\delta\left(f - \frac{\pi}{3}\right) + \frac{1}{2}\delta\left(f - \frac{\pi}{2}\right)\right]$$



Where $N = 200$ was arbitrarly chosen, the number of samples only influences the Y-axis

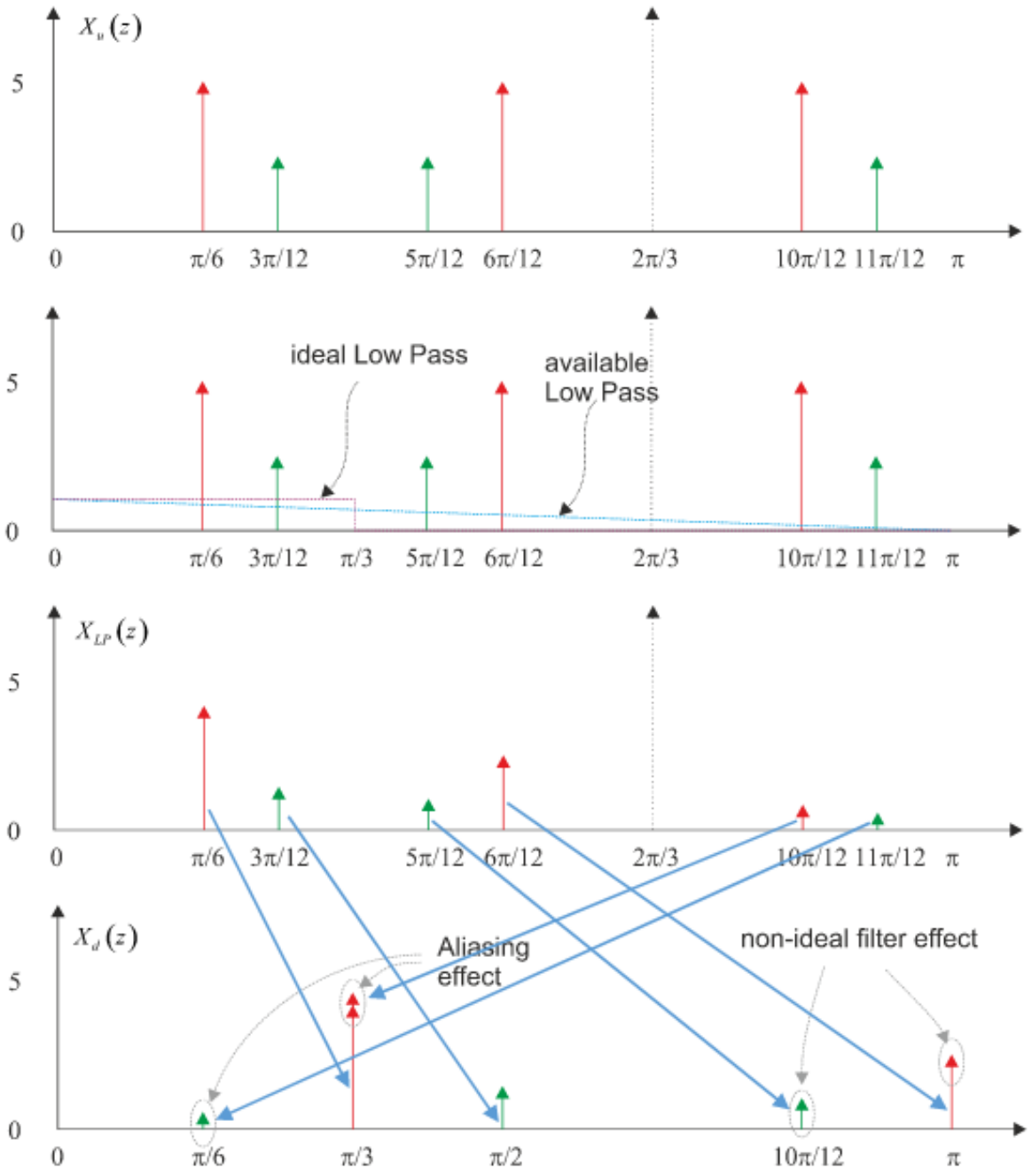
9.5.2 Non-ideal lowpass filter for rational sampling: aliasing

[6pts] In order to get the result we can simply use a lowpass filter whose frequency behavior is depicted in the following graph:



The magnitude decreases linearly from 1 at $\omega = 0$ to 0 at $\omega = \pi$ and the phase is null for the whole spectrum.
Describe the output signal underlining the impact (gain/attenuation) of such a filter on the signal componenets and on the spurious components due to the sample rate change.

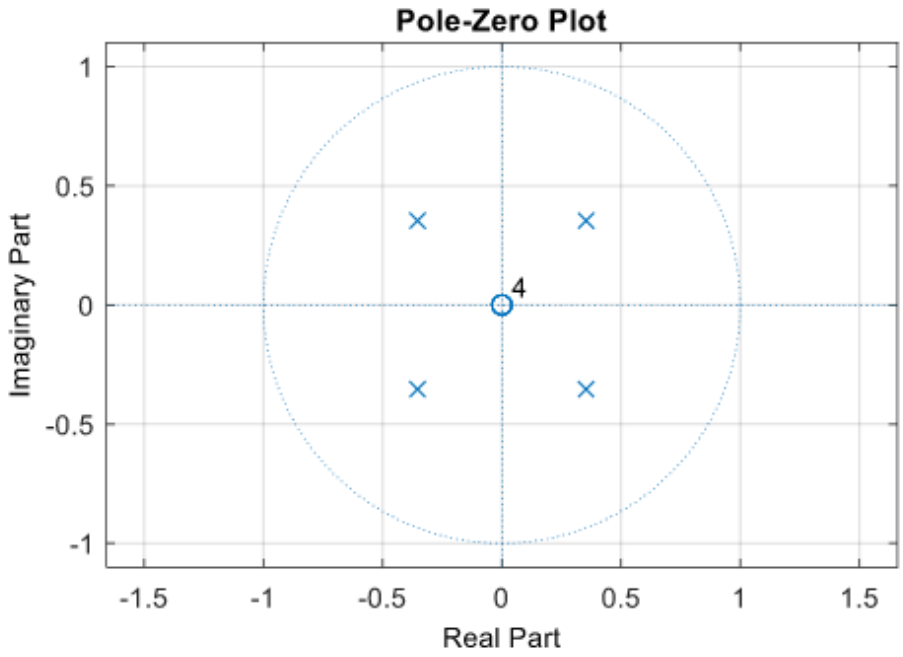
As the lowpass filter is not ideal, some pulses after the cutoff frequency of $\frac{\pi}{3}$ will not be removed completely, but they will be attenuated. The presence of such pulses, even if their amplitude will be low, will introduce aliasing as seen below:



The pulse at $\pi/3$ that introduces aliasing is due to symmetric component that is located AFTER the Nyquist frequency (not seen in the plot): IF ANY PULSE IS FOUND AFTER THE NYQUIST WE WILL HAVE ALIASING

9.6 20200218-1

A filter $H(z)$ has the following zero-pole plot:



Where the distance of every pole from the origin is $\frac{1}{2}$.

9.6.1 Z-transform and pole-zero plot

[2pts] Which kind of filter is it (FIR or IIR)? Is it stable or unstable? Is it causal or not?

[2pts] Provide its z-transform $H(z)$

It is a pure IIR filter, stable and causal (since the number of zeros, to be causal, cannot be greater than the number of poles) Using $1-2\rho\cos\theta+\rho^2$, we find

$$H(z)=\frac{1}{\left(1+\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)\left(1-\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)}$$

$$H(z)=\frac{1}{1+\frac{1}{16}z^{-4}}$$

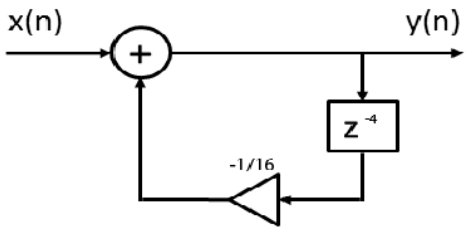
9.6.2 Difference equation

[2pts] Provide its difference equation

$$y(n)=x(n)-\frac{y(n-4)}{16}$$

9.6.3 Filter schema drawing

[3pts] Draw a schematic for the filter implementation



9.6.4 Output samples: IIR

[3pts] Provide the first 9 outputs of the filter for the impulse response $x(n) = \{1, 0, 0, \cdots\}$

[3pts] Provide the first 9 outputs of the filter for the step function response $x(n) = \{1, 1, 1, \cdots\}$

As we have the difference equation of $y(n)$, we compute the output samples by hand.

$$y(n) = x(n) - \frac{y(n-4)}{16}$$

As for $n < 0$ the system does not exist:

- Impulse response

$$\begin{cases} y(0) = x(0) = 1 \\ y(1) = x(1) = 0 \\ y(2) = x(2) = 0 \\ y(3) = x(3) = 0 \\ y(4) = x(4) - \frac{1}{16}y(0) = -\frac{1}{16} \\ y(5) = x(5) - \frac{1}{16}y(1) = 0 \\ y(6) = x(6) - \frac{1}{16}y(2) = 0 \\ y(7) = x(7) - \frac{1}{16}y(3) = 0 \\ y(8) = x(8) - \frac{1}{16}y(4) = \left(-\frac{1}{16}\right)^2 \end{cases}$$

- Step function response

$$\begin{cases} y(0) = x(0) = 1 \\ y(1) = x(1) = 1 \\ y(2) = x(2) = 1 \\ y(3) = x(3) = 1 \\ y(4) = x(4) - \frac{1}{16}y(0) = \frac{15}{16} \\ y(5) = x(5) - \frac{1}{16}y(1) = \frac{15}{16} \\ y(6) = x(6) - \frac{1}{16}y(2) = \frac{15}{16} \\ y(7) = x(7) - \frac{1}{16}y(3) = \frac{15}{16} \\ y(8) = x(8) - \frac{1}{16}y(4) = 1 - \frac{1}{16}\frac{1}{15} \end{cases}$$

9.7 20190910-2

A signal $x[n] = \{-1,-1,2,0,0,1,-2,1,0,1,1,-1\}$ shall be filtered with this FIR fitler: $h[n] = \{1,-1,2\}$. We need to process blocks of 4 samples of the original signal for real time processing and we want to get the result depicting all intermediate steps with these methods:

9.7.1 Overlap and Add

[4pt] Overlap and Add

Subdividing the signal in blocks of L elements, in this case $L = N = 4$ so we get 3 blocks:

$x_1[n] = \{-1,-1,2,0\}$
$x_2[n] = \{0,1,-2,1\}$
$x_3[n] = \{0,1,1,-1\}$

Now we filter using convolution with $h[n] = \{1,-1,2\}$

$x[n] * h[n] = x[n] * \delta[n] + x[n] * -\delta[n-1] + x[n] * 2\delta[n-2] =$
$= \cdots x[n] - x[n-1] + 2x[n-2] =$

$n=$	0	1	2	3	4	5
$x_1[n]$	-1	-1	2	0		
$-x_1[n-1]$	0	1	1	-2	0	
$2x_1[n-2]$	0	0	-2	-2	4	0
$x_2[n]$	0	1	-2	1		
$-x_2[n-1]$	0	0	-1	2	-1	
$2x_2[n-2]$	0	0	0	2	-4	2
$x_3[n]$	0	1	1	-1		
$-x_3[n-1]$	0	0	-1	-1	1	
$2x_3[n-2]$	0	0	0	2	2	-2

$\Rightarrow y_1[n] = \{-1 \quad 0 \quad 1 \quad -4 \quad 4 \quad 0\}$
$\Rightarrow y_2[n] = \{0 \quad 1 \quad -3 \quad 5 \quad -5 \quad 2\}$
$\Rightarrow y_3[n] = \{0 \quad 1 \quad 0 \quad 0 \quad 3 \quad -2\}$

Considering the overlapping tails, as:

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} y_r[n - r\underbrace{L}_{4}]$$

$$\begin{bmatrix} -1 & 0 & 1 & -4 & 4 & 0 & & & & & & \\ & & & & 0 & 1 & -3 & 5 & -5 & 2 & & \\ & & & & & & & & 0 & 1 & 0 & 0 & 3 & -2 \end{bmatrix}$$

And the final result is

$$y[n] = \{-1 \quad 0 \quad 1 \quad -4 \quad 4 \quad 1 \quad -3 \quad 5 \quad -5 \quad 3 \quad 0 \quad 0 \quad 3 \quad -2\}$$

9.7.2 Overlap and Save

[4pt] Overlap and Save

Subdividing the signal in blocks of L elements, in this case $L = N = 4$ so we get 3 blocks:

$x_1[n] = \{-1,-1,2,0\}$
$x_2[n] = \{0,1,-2,1\}$
$x_3[n] = \{0,1,1,-1\}$

As the filter $h[n]$ has length $P = 3$, we need to add $P-1 = 2$ "0"s at the beginning to account for the circular convolution effect, and for each $x_{i>1}$ we add the 2 samples before, hence in overlap and save

$$L = N + P - 1 = 4 + 2 = 6$$

$x_1[n] = \{\mathbf{0},\mathbf{0},-1,-1,2,0\}$
$x_2[n] = \{\mathbf{2},\mathbf{0},0,1,-2,1\}$
$x_3[n] = \{-\mathbf{2},\mathbf{1},0,1,1,-1\}$

Now we filter using circular convolution with $h[n] = \{1,-1,2\}$. But we don't compute the circular convolution traditionally, we use this strategy:

$$x_L \circledast h_L = \left[\underbrace{\cdots\cdots}_{P-1 \text{ to discard}} \quad \cdots\cdots \underbrace{\cdots}_{L-1}\right]$$

a circular convolution of block L will discard the first $P-1$ samples, then the rest till $L-1$ will be the same obtained from a normal convolution

$$x[n] * h[n] = x[n] * \delta[n] + x[n] * -\delta[n-1] + x[n] * 2\delta[n-2] = x[n] - x[n-1] + 2x[n-2]$$

$n =$	0	1	2	3	4	5
$x_1[n]$	0	0	-1	-1	2	0
$-x_1[n-1]$	0	0	0	1	1	-2
$2x_1[n-2]$	4	0	0	0	-2	-2
$x_2[n]$	2	0	0	1	-2	1
$-x_2[n-1]$	-1	-2	0	0	-1	2
$2x_2[n-2]$	-4	2	4	0	0	2
$x_3[n]$	-2	1	0	1	1	-1
$-x_3[n-1]$	1	2	-1	0	-1	-1
$2x_3[n-2]$	2	-2	-4	2	0	2

We throw the first two as $P-1 = 2$, so:

$\Rightarrow y_1[n] = \{\cdot \quad \cdot \quad -1 \quad 0 \quad 1 \quad 4\}$
$\Rightarrow y_2[n] = \{\cdot \quad \cdot \quad 4 \quad 1 \quad -3 \quad 5\}$
$\Rightarrow y_3[n] = \{\cdot \quad \cdot \quad -5 \quad 3 \quad 0 \quad 0\}$

And the final result is

$$y[n] = \{y_1\ y_2\ y_3\} = \{-1 \quad 0 \quad 1 \quad 4 \quad 4 \quad 1 \quad -3 \quad 5 \quad -5 \quad 3 \quad 0 \quad 0\}$$

9.8 20190722-1

A signal $x(t)$ is sampled at $10kHz$, and we want to find the relative magnitude of its components at $1kHz$ w.r.t. the components at $4kHz$ but we cannot use the FFT.

Therefore we decide to filter the signal at $1kHz$ and $4kHz$ multiplying it with a real sinusoid at $1kHz$ and at $4kHz$ respectively.

9.8.1 Not using FFT/DFT, use a cosine and pole-zero plot

[3 pts] Depict the pole zero plots for the two sinusoids.

The signal is sampled at $10kHz$ so

$$F_s = 10kHz$$

If we could use th DFT or FFT, we could just use

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n} \Big|_{f=1kHz}$$

Instead we exploit a real sinusoid in the time domain:

$$s(n) = \cos(2\pi \overline{f} n)$$

Whose Fourier transform will result in two pulses (two δ's) in the frequency domain as it is a cosine: in $\overline{f}$ and in $-\overline{f}$. If we compute

$$X(f) \cdot DTFT(\cos(2\pi f n))\big|_{f=1kHz} = X(f = 1kHz)\delta(f - 1kHz) + X(f = 1kHz)\delta(f + 1kHz)$$

As we consider normalized frequencies and compute the z-transform:

$$f_1 = 1kHz \rightarrow \tilde{f}_1 = \frac{f_1}{Fs}$$

$$\cos\left(2\pi \frac{1kHz}{10kHz} n\right) \cdot \underbrace{u(n)}$$

Discrete unitary step, the signal is defined only in the positive values

Using

$$Z\{r^n \cos(\omega_0 n) u(n)\} = \frac{1 - r\cos(\omega_0) z^{-1}}{1 - 2r\cos(\omega_0) z^{-1} 1 + r^2 z^{-2}}$$

$$Z - TX = \frac{1 - \cos(\pi/5) z^{-1}}{1 - 2\cos(\pi/5) z^{-1} + z^{-2}}$$

Alternatively we convert it to the exponential form and solve the z-transform explicitely

$$\begin{cases} \frac{e^{j\frac{\pi}{5}n}+e^{-j\frac{\pi}{5}n}}{2} \cdot u(n) \\ X(z) = \sum_{n=-\infty}^{+\infty} x(n) \cdot z^{-n} \end{cases}$$

So due to geometric sum:

$$Z-TX(e^{j\frac{\pi}{5}n} \cdot u(n)) = \sum_{n=0}^{\infty} e^{j\frac{\pi}{5}n} z^{-n} = \sum_{n=0}^{\infty} \left(e^{j\frac{\pi}{5}} z^{-1}\right)^n = \frac{1}{1-e^{j\frac{\pi}{5}z^{-1}}}$$

$$Z-TX = \frac{1}{2}\left(\frac{1}{1-e^{j\frac{\pi}{5}z^{-1}}} + \frac{1}{1-e^{-j\frac{\pi}{5}z^{-1}}}\right) = \cdots$$

From

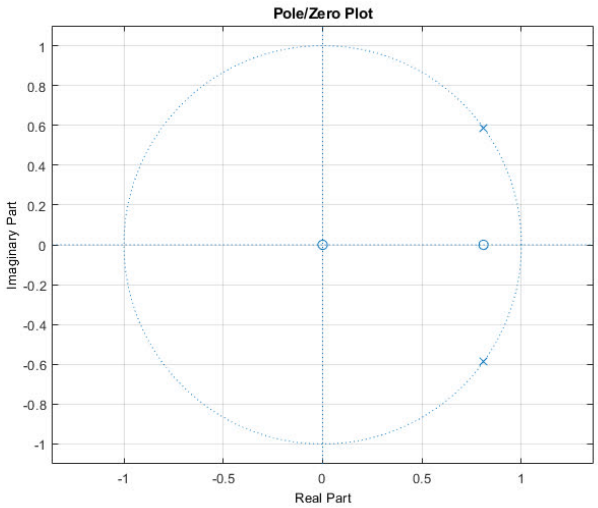
$$H_1(z) = \frac{1-\cos(\pi/5)z^{-1}}{1-2\cos(\pi/5)z^{-1}+z^{-2}}$$

We can find the zeros and the poles

- For the denominator we exploit:

  $$1-2\rho\cos(\theta)z^{-1}+\rho^2 z^{-2} \leftrightarrow \rho(\cos(\theta) \pm j\sin(\theta))$$

  Poles in $e^{\pm j\frac{\pi}{5}}$ **plus double zero in the origin**
- For the numerator the zeros are just the coefficients of z^{-k}, so we have a zero in $\cos(\frac{\pi}{5})$ **plus a pole in the origin which will cancel out with the double zero in the origin leaving a single zero**



Same reasoning for $4Hz$, we will get

$$H_2(z) = \frac{1-\cos(4\pi/5)z^{-1}}{1-2\cos(4\pi/5)z^{-1}+z^{-2}}$$

9.8.2 Difference equations (anti z-transform)

[6 pts] Define the difference equations both for the first and the second sinusoid.

Everything at numerator will be x's on the right, everything at the denominator will be y's on the left:

$H_1(z) = \frac{1-\cos(\pi/5)z^{-1}}{1-2\cos(\pi/5)z^{-1}+z^{-2}}$

$y_1(n) - 2\cos(\pi/5)y(n-1) + y(n-2) = x(n) - \cos(\pi/5)x(n-1)$

$H_2(z) = \frac{1-\cos(4\pi/5)z^{-1}}{1-2\cos(4\pi/5)z^{-1}+z^{-2}}$

$y_2(n) - 2\cos(4\pi/5)y(n-1) + y(n-2) = x(n) - \cos(4\pi/5)x(n-1)$

9.8.3 How to apply the filters

[3 pts] Describe how we shall apply the previously defined filters in order to measure the relative intensity of the input signal at 1kHz with respect to 4kHz.

In order to apply the filters we have to take a portion of the input signal and process it with the two filters (the difference equations above).
To avoid polarizations in the measure we should take a set of samples multiple of the wavelengths (in term of samples) of both the sinusoids.
The ratio of the two magnitudes of the outputs will give the result.

9.8.4 Cosine method weakness w.r.t. Fourier Transform

[1 pts] What is a weakness of the proposed system with respect to the Fourier Transform?

The weakness with respect to the Fourier Transform is related to the fact that we are not using the phase as in the Discrete Fourier Transform, so, signal components with the same amplitude and frequency but different phases will give different results.
By multiplying our signal with a signal in the Fourier domain, it means that we are convolving the signal with a cosine:

$$y(n) = \sum_m x(n-m)\cos(2\pi \underbrace{\tilde{f}}_{1\text{kHz}} n)$$

While the DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{k}{N}n}\Big|_{f=1kHz} = \sum_{n=0}^{N-1} x(n)\left(\cos(2\pi fn) - j\sin(2\pi fn)\right)\Big|_{f=1kHz}$$

So we are not considering a sin component.

9.9 20190212-2

A signal is sampled at $10kHz$ filling all the available band. We need to upsample it to $15kHz$.

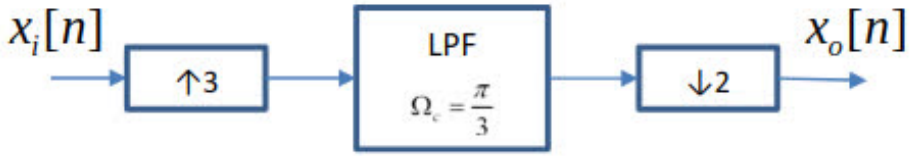
9.9.1 Upsample by noninteger pipeline

[3pts] Provide the full pipeline to properly upsample the signal minimizing the presence of artifacts or aliasing and detailing the parameters of every bock

As we upsample to a noninteger, we can:

$$10 \to 30 \to 15$$

So we first upsample by $L = 3$ and then downsample by $M = 2$, which means we introduce a LPF of $\Omega_c = \min\left(\frac{\pi}{3}, \frac{\pi}{2}\right) = \frac{\pi}{3}$ (in other words a filter with that cutoff $H(z)\big|_{\omega_c=\frac{\pi}{3}, f_c=\frac{1}{6}}$)



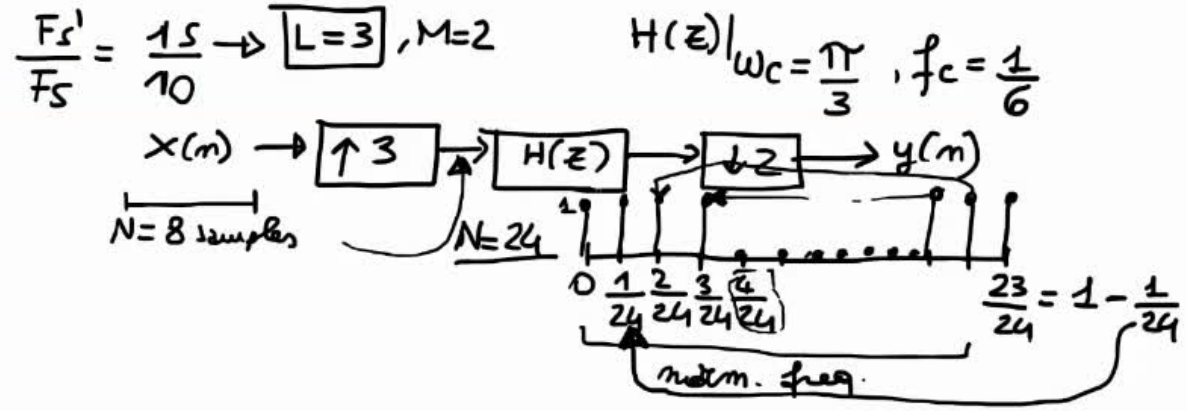
9.9.2 Cutoff DFT

[7pts] Assuming that we are working on blocks of 8 samples and we are filtering the signal in frequency domain, describe a possible implementation of the previous system working in the frequency domain (using the DFT).

8 samples, so $N = 8$, after upsample we will have $8 \cdot 3 = 24$ samples. If we have to implement that filter with that frequency and 24 samples, so

$$f = \left[0, \frac{1}{24}, \frac{2}{24}, \cdots, \frac{23}{24}\right]$$

24 samples that represent normalized frequencies, a cutoff of $f_c = \frac{1}{6}$ means that we keep the pulses till f_c and introduce at the end 3 pulses (symmetry)



9.10 20190116-1

A digital signal is made of 3 samples: $x[n] = \{2, 3, -1\}$. Working exclusively in the Discrete frequencies domain, we want to find its continuous component (0 frequency) removing the high frequencies.

9.10.1 DFT to remove the high frequencies

[3pts] Define the W matrix to perform the DFT of the 3 samples signal.

[2pts] Calculate $X[k]$, the DFT of the input signal

[3pts] Define the filter $H[k]$ in the frequency domain in order to remove the high frequencies and extract the filtered signal $Y[k]$

$$W_i = e^{-j\frac{2\pi}{i}} \qquad i = 3$$

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_i^1 & W_i^2 \\ 1 & W_i^2 & W_i^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix}$$

$$X[k] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 - 2\sqrt{3}j \\ 1 + 2\sqrt{3}j \end{bmatrix}$$

As we want to remove the high frequencies, we choose

$$H[k] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow Y[k] = X[k] \cdot H[k] = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}$$

9.10.2 DFT matrix inverse, out put in time domain

[4pts] Transform $H[k]$ and $Y[k]$ into their time domain representation

The inverse opf W is its conjugate transpose divided by $N = 3$

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix} \rightarrow W^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix}$$

$$y[n] = W^{-1}Y[k] = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4/3 \\ 4/3 \\ 4/3 \end{bmatrix}$$

$$h[n] = W^{-1}H[k] = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

9.11 20190116-2

A maximum phase filter has the following transfer function:

$$H_M(z) = \frac{\left(1-2\sqrt{2}z^{-1}+4z^{-2}\right)\left(1+2\sqrt{2}z^{-1}+4z^{-2}\right)}{4+z^{-2}}$$

9.11.1 Zero-pole plot

[2 Pts.] Provide its zeros-poles plot

Poles:

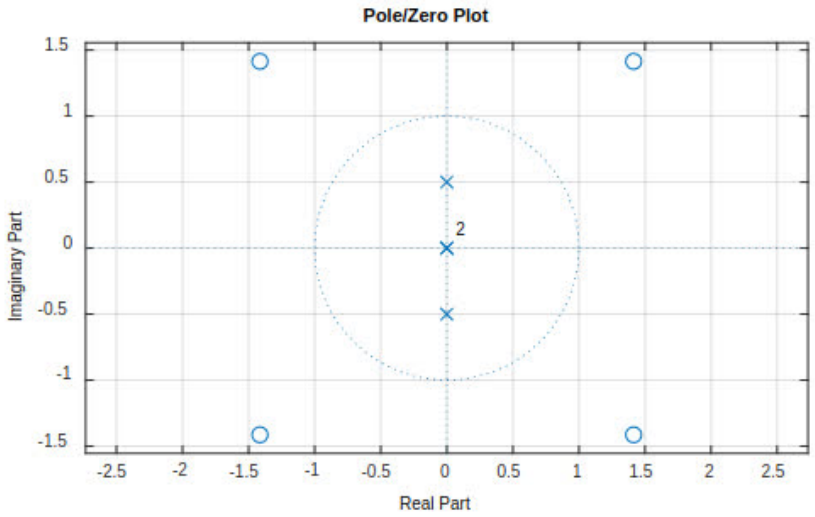
$$z^{-2} = -4 \rightarrow \pm\frac{i}{2}$$

Zeros, make sure the $\Delta < 0$, otherwise we cannot use the formula for complex conjugates. In this case the condition holds, so

$$1-2\rho\cos\theta z^{-1}+\rho^2 z^{-2} \Rightarrow \begin{cases} 2\sqrt{2} = 2\rho\cos\theta \\ 4 = \rho^2 \end{cases} \Rightarrow \begin{cases} \rho = 2 \\ \theta = \pm\frac{\pi}{4} \end{cases}$$

$$1-2\rho\cos\theta z^{-1}+\rho^2 z^{-2} \Rightarrow \begin{cases} -2\sqrt{2} = 2\rho\cos\theta \\ 4 = \rho^2 \end{cases} \Rightarrow \begin{cases} \rho = 2 \\ \theta = \pm\frac{3}{4}\pi \end{cases}$$

We also introduced in the origin 2 zeros and 4 poles, so we will have 2 poles:



9.11.2 Define minimum phase filter with fixed amplitude/magnitude

[4 Pts.] Provide its minimum phase version, $H_m(z)$, with exactly the same magnitude response.

As $\rho = 2$, now $\rho = \frac{1}{2}$:

$$H_m(z) = A\frac{\left(1-\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)\left(1+\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)}{4+z^{-2}}$$

Then we find A by imposing:

$$H_M(z=0) = H_m(z=0) \Rightarrow \frac{1}{4} = A\frac{1}{4} \Rightarrow A = 16$$

Alternatively write the G version of those that introduce maximum phase

$$H_m(z) = \frac{\left(4-2\sqrt{2}z^{-1}+z^{-2}\right)\left(4+2\sqrt{2}z^{-1}+z^{-2}\right)}{4+z^{-2}}$$

9.11.3 Cosine signal and generic filter

[5 Pts.] A signal $x(t) = \cos(2\pi 1000t)$ is sampled at $4kHz$. Define the outputs $y_M[n]$ and $y_m[n]$ with the proper amplitude and phase from the two filters $H_M(z)$ and $H_m(z)$.

A cosine by a generic filter H will give another cosine with different amplitude and phase.
$f_0 = 1000 = 1kHz$, $F_s = 4kHz$, so the normalized frequency $\tilde{f}_0 = \frac{f_0}{F_s} = \frac{1}{4}$ and $\tilde{\omega}_0 = \frac{\pi}{2}$
In the z-transform we will have (we always consider the normalized!):

$$z_0 = 1 * e^{j\omega} = e^{j2\pi f} = e^{j\pi/2} = j$$

- Maximum phase

$$H_M(z) \rightarrow \begin{cases} |H_M(z=j)| = \cdots = \frac{17}{3} \\ \angle H_M(z=j) = \angle H_M(z) = \frac{\left(1-2\sqrt{2}j^{-1}-4\right)\left(1+2\sqrt{2}j^{-1}-4\right)}{4-1} \end{cases}$$

  The phase:

$$\angle H_M(z) = \frac{\left(2\sqrt{2}j-3\right)\left(-2\sqrt{2}j-3\right)}{3} = \angle(-3+2\sqrt{2}j) + \angle(-3-2\sqrt{2}j) + \angle(3)$$

  The phase of a real number is zero in the complex plane, while the phase of the first term will be in the third quadrant and the phase of the second term will be the first mirrored, so summing them together will get $\angle H_M(j) = 2\pi$

$$y_M(n) = A \cdot \cos(\tilde{\omega}_0 n + \phi) = \frac{17}{3}\cos(\frac{\pi}{2}n + 2\pi)$$

- Minimum phase, as we **imposed that the magnitude is the same**

$$H_m(z) \rightarrow \begin{cases} |H_m(z=j)| = |H_M(z=j)| = \frac{17}{3} \\ \angle H_M(z=j) = \angle\frac{\left(3+2\sqrt{2}j\right)\left(3-2\sqrt{2}j\right)}{3} = 0 \end{cases}$$

  The phase is 0 as the phase of the first is in the first quadrant $\angle H_m(j) = 0$

$$y_m(n) = A \cdot \cos(\tilde{\omega}_0 n + \phi) = \frac{17}{3}\cos(\frac{\pi}{2}n + 0)$$

9.12 20181107-1

An analog signal $x(t) = 2\cos(2\pi 20t) + 3\sin(2\pi 60t) + 4\cos(2\pi 80t)$ is sampled at 160 samples/s and filtered with an IIR filter with the following finite differences equation:

$$y[n] = x[n] + \sqrt{2}x[n-1] + x[n-2] - 0.9\sqrt{2}[n-1] - 0.81y[n-2]$$

9.12.1 Z-transform and pole-zero plot

[2pts] Provide the z-transform of the filter.

[3pts] Provide the zeros-poles plot of the filter.

$$F_s = 160$$

So the sampled signal is

$$x(t) = 2\sin(\frac{\pi}{4}n) + 3\cos(\frac{3\pi}{4}n) + 4\sin(\pi n)$$

$$H(z) = \frac{1 + \sqrt{2}z^{-1} + z^{-2}}{1 + 0.9\sqrt{2}z^{-1} + 0.81z^{-2}}$$

- Numerator:

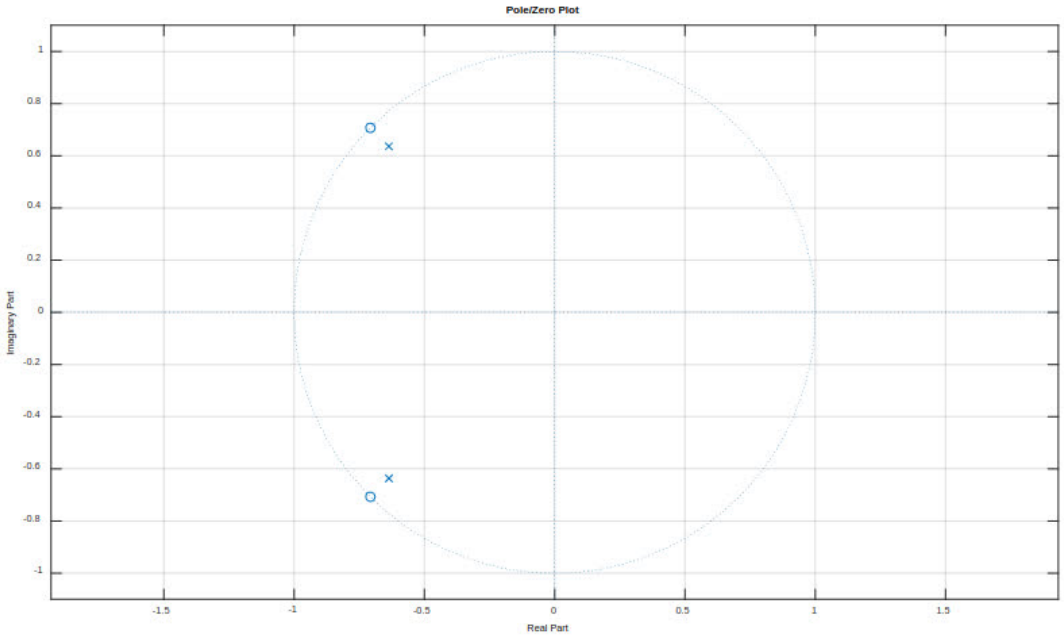
$$\begin{cases} \rho^2 = 1 \\ 2\rho\cos\theta = -\sqrt{2} \end{cases} \Rightarrow \begin{cases} \rho = 1 \\ \theta = \pm\frac{3\pi}{4} \end{cases}$$

  Two zeros in $e^{\pm j\frac{3\pi}{4}}$, two poles in 0
- Denominator:

$$\begin{cases} \rho^2 = 0.81 \\ 2\rho\cos\theta = -0.9\sqrt{2} \end{cases} \Rightarrow \begin{cases} \rho = 0.9 \\ \theta = \pm\frac{3\pi}{4} \end{cases}$$

  Two poles in $0.9e^{\pm j\frac{3\pi}{4}}$, two zeros in 0

So no zeros or poles at the origin, system stable as all poles and zero in the circle:



9.12.2 Magnitude and phase plot by hand

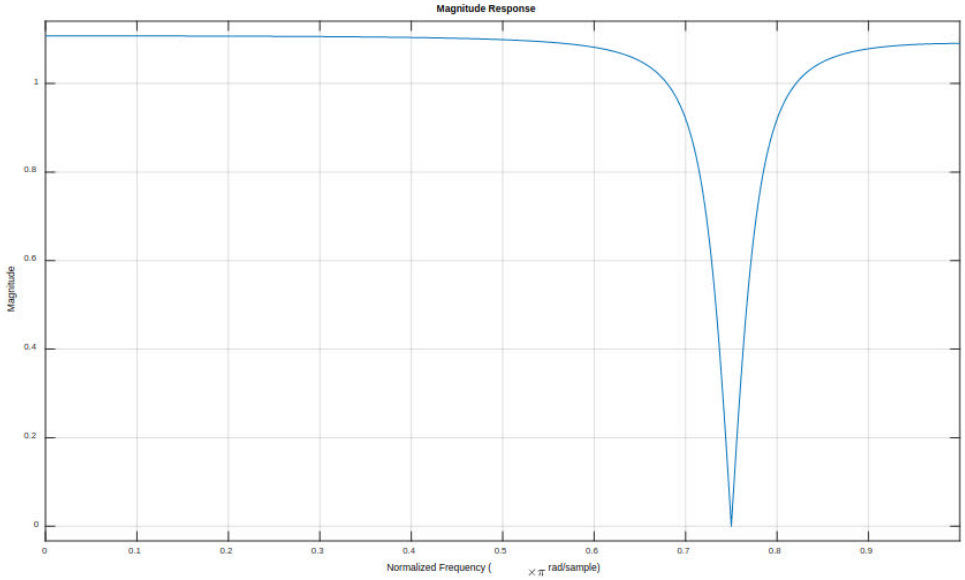
[3pts] Represent an approximate behavior of the magnitude and phase in the range $(0\ -\pi)$.

It is a notch

$$H(z) = \frac{1+\sqrt{2}z^{-1}+z^{-2}}{1+0.9\sqrt{2}z^{-1}+0.81z^{-2}}$$

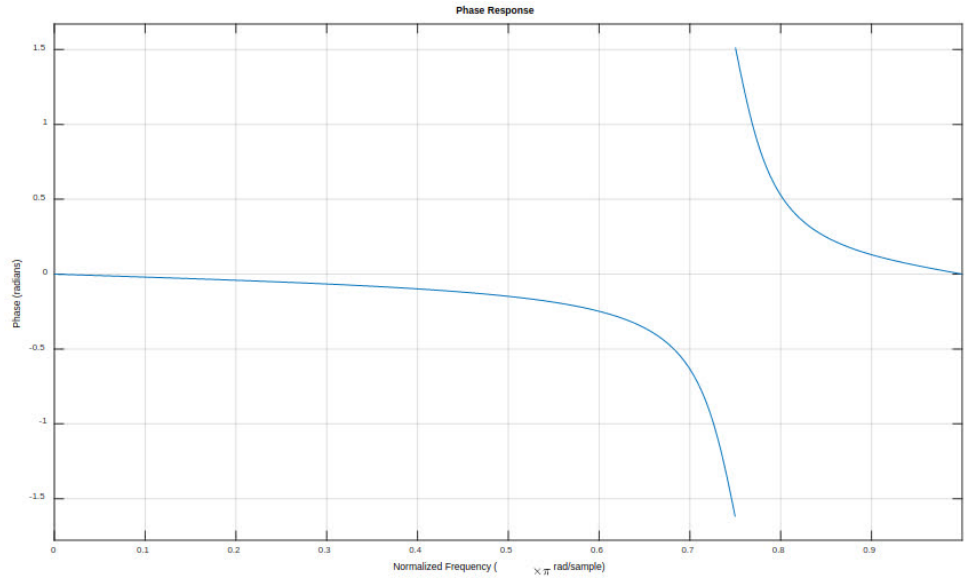
The range is normalized in $(0\ 1)$. We start from the magnitude, we compute some notable points:

- $|H(z=1,\omega=0)| = \frac{2+\sqrt{2}}{1.81+0.9\sqrt{2}} \sim 1.1075$
- $|H(z=j,\omega=\pi/2)| = \left|\frac{-\sqrt{2}j}{0.19-0.9\sqrt{2}j}\right| = \frac{\sqrt{2}}{\sqrt{0.19^2+0.81*2}} \sim 1.0989$
- $|H(z=e^{-j\frac{3\pi}{4}},\omega=3\pi/4)| = 0$, as in that angle we meet the zero so we are multiplying by 0
- $|H(z=-1,\omega=\pi)| = \frac{2-\sqrt{2}}{1.81-0.9\sqrt{2}} \sim 1.0904$



About the phase, a notch will have jumps, we have to look at the zero plot:

- $\angle H(z=1) = 0$ as we have opposite contributes for both poles and zeros
- $\angle H(z=e^{-j\frac{3\pi}{4}})$ we don't know the exact value, but at that pole we will have a jump of π
- $\angle H(z=-1) = 0$ as we have opposite contributes for both poles and zeros



9.12.3 Sampled signal, cosine signal and generic filter

[5pts] What will be the discrete output signal when the input is the sampled version of $x(t)$

$$F_s = 160$$

The signal is

$$x(t) = 2\cos(2\pi 20t) + 3\sin(2\pi 60t) + 4\cos(2\pi 80t)$$

The sampled signal is:

$$x(t) = 2\cos(\frac{\pi}{4}n) + 3\sin(\frac{3\pi}{4}n) + 4\cos(\pi n)$$

It has three components, so:

- $\tilde{f}_1 = \frac{f_1}{F_s} = \frac{20}{160} = \frac{1}{8} \rightarrow \tilde{\omega}_1 = \frac{\pi}{4} \rightarrow z_1 = 1 * e^{j\pi/4}$

$$\begin{cases} |H(z_1)| = \cdots = 1.1068 \\ \angle H(z_1) = \cdots = -0.053 \end{cases}$$

- $\tilde{f}_2 = \frac{f_2}{F_s} = \frac{60}{160} = \frac{3}{8} \rightarrow \tilde{\omega}_2 = \frac{3\pi}{4} \rightarrow z_2 = 1 * e^{j3\pi/4}$

$$\begin{cases} |H(z_2)| = 0 \\ \angle H(z_2) = \text{ does not matter, the magnitude is 0 so the term disappears} \end{cases}$$

  As a matter of fact the filter is a notch in $\tilde{\omega}_2$

- $\tilde{f}_3 = \frac{f_3}{F_s} = \frac{80}{160} = \frac{1}{2} \rightarrow \tilde{\omega}_3 = \pi \rightarrow z_3 = 1 * e^{j\pi} = -1$

$$\begin{cases} |H(z_3)| = 1.0904 \\ \angle H(z_3) = 0 \end{cases}$$

The output is:

$$y[n] = 2 \cdot 1.1065\cos\left(\frac{\pi}{4}n - 0.053\right) + 4 \cdot 1.09\cos(\pi n)$$

9.13 20161121-1

A signal is sampled at $80kHz$. We want to transform the sampled digital signal into a signal sampled at $120kHz$.

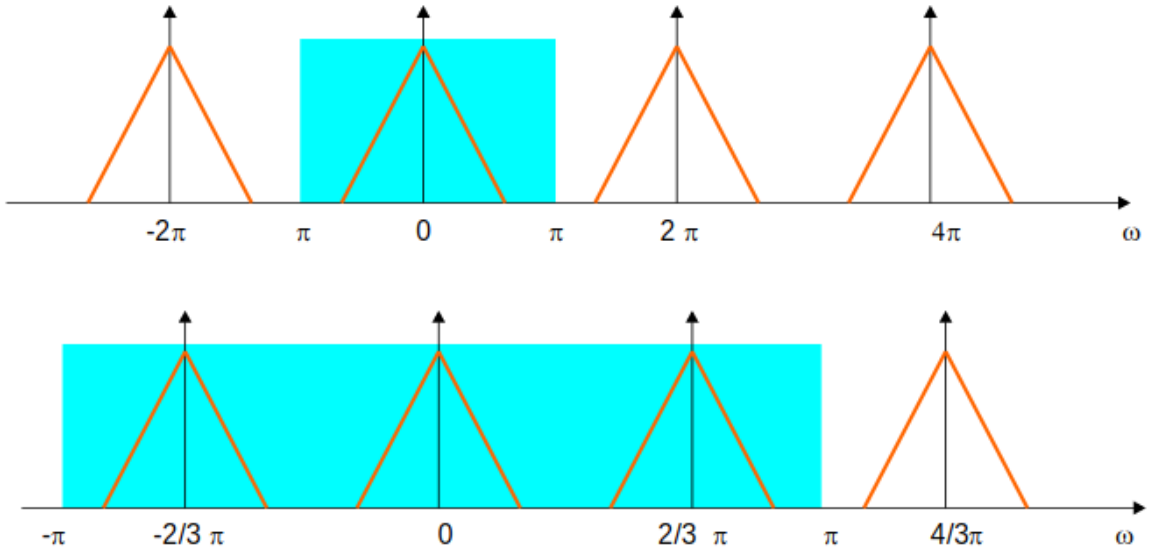
9.13.1 Upsample by noninteger pipeline

Define the whole pipeline to get this result providing accurate description of the filters adopted both for sampling and re-sampling.

$$80 \rightarrow 240 \rightarrow 120$$

$$L = 3 \qquad M = 2$$

In the upsample the spectrum will be compressed of an order of 3, replicas will appear below the Nyquist frequency.
We add a lowpass filter with a cutoff of $\min\left[\frac{\pi}{3}, \frac{\pi}{2}\right] = \frac{\pi}{3}$ to remove the replicase below the Nyquist frequency. At the same time the LPF will prevent aliasing that the downsampling would have introduced.



in cyan it is represented the region below the Nyquist frequencies before and after the upsampling.

9.13.2 DFT matrix in upsampling

We want to re-sample the signal in real-time working on blocks of 1024 samples and in frequency domain.

1. Describe how the DFT can be obtained, provide how the transform matrix can be realized for this specific case and the operations to get the DFT of the input signal
2. Define a proper filter in the frequency domain for the specific task of the first question and how it has to be applied to the signal
3. Describe how the filtered signal can be transformed back into the time domain. Are there issues related to circular convolution for periodicity assumption for this specific case?

1. In order to apply the upsampling and then the filter in the frequency domain I need to take 1024 samples from the signal, add two samples with zero value every sample and then define the DFT Matrix.

2. The DFT kernel is $w = e^{j\frac{2\pi}{3\cdot 1024}} = e^{j\frac{\pi}{1536}}$, so the W matrix for the DFT will be 3072×3072 matrix. Once I multiply this matrix with the set of upsampled samples I get the DFT of the upsampled signal, then we apply the low pass filter in the frequency domain by removing all the frequencies between $\frac{\pi}{3}$ and $2\pi - \frac{\pi}{3} = \frac{5}{3}\pi$ (since now I have to consider that my spectrum is represented in the frequency range between 0 and 2π).
   This can be done simply setting to zero all frequencies between the 513^{th} sample (3072/6+1) and the 2560^{th} sample (3072*5/6) of the DFT of my signal which is exactly a low-pass filter rectangular in the frequency domain.

3. In order to go back into the time domain we have to pre-multiply the 3072 samples in the frequency domain to the inverse of our previous matrix.
   Analyzing what we are performing in the time domain, since we are multiplying the spectrum of upsampled signal with a low pass rectangle filter $rect\left(\frac{f}{2\cdot 1/6}\right)$ we are implicitly performing a circular convolution with the IDFT of the rectangle filter: $h(n) = \text{sinc}\left(\frac{n}{3}\right)$ sinc 3 whose range of the first lobe is 6 samples.
   We can then conclude that for a generic signal the circular convolution error is present and, due to the narrowness of the filter its effect operates significantly just of a few samples close to the signal tails.
   Whenever we are multiplying by matrix in DFT, we are introducing some artifacts in the cyclic convolution in the time domain

9.14 20161121-2

The following signal:

$$x(t) = 5 + 3\cos(2\pi f_1 t) + \cos(2\pi f_2 t)$$

Where $f_1 = 10kHz$ and $f_2 = 40kHz$ is sampled at $80kHz$ and is filtered with the following filter:

$$H(z) = \frac{1 - \sqrt{2}z^{-1} + z^{-2}}{1 - 0.7\sqrt{2}z^{-1} + 0.49z^{-2}}$$

9.14.1 Cosine signal and generic filter (plus a continuous term)

Plot the fitler in the z-plane and provide the output signal in the time domain.

The filter has:

- Numerator:

$$\begin{cases} 2\rho\cos(\theta) = \sqrt{2} \\ \rho^2 = 1 \end{cases} \Rightarrow \begin{cases} \theta = \pm\frac{\pi}{4} \\ \rho = 1 \end{cases}$$

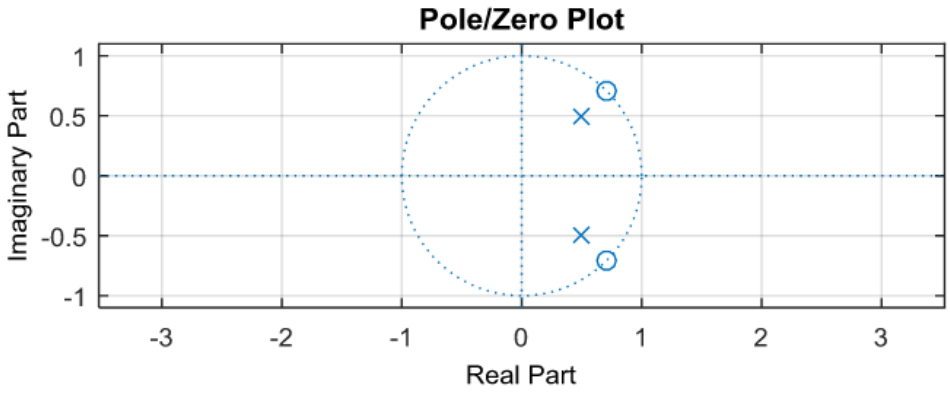
  Two zeros in $e^{\pm j\frac{\pi}{4}}$, two poles in 0

- Denominator:

$$\begin{cases} 2\rho\cos(\theta) = 0.7\sqrt{2} \\ \rho^2 = 0.49 \end{cases} \Rightarrow \begin{cases} \theta = \pm\frac{\pi}{4} \\ \rho = 0.7 \end{cases}$$

  Two poles in $0.7e^{\pm j\frac{\pi}{4}}$, two zeros in 0

So no zeros or poles at the origin:



$$H(z) = \frac{1 - \sqrt{2}z^{-1} + z^{-2}}{1 - 0.7\sqrt{2}z^{-1} + 0.49z^{-2}}$$

The sampled signal will be:

$$x[n] = 5 + 3\cos\left(\frac{\pi}{4}n\right) + \cos(\pi n)$$

It has three components:

- 5 continuous component, $\tilde{\omega}_0 = 0 \to z_0 = 1$

$$\begin{cases} |H(z_0)| = \frac{|1-\sqrt{2}+1|}{|1-0.7\sqrt{2}+0.49|} \sim 1.17 \\ \angle H(z_1) = 0 \end{cases}$$

- $\tilde{\omega}_1 = \frac{\pi}{4} \to z_1 = 1 \cdot e^{j\pi/4}$

$$\begin{cases} |H(z_1)| = 0 \qquad \text{It is a zero} \\ \angle H(z_1) = \cdots \end{cases}$$

- $\tilde{\omega}_2 = \pi \to z_2 = 1 \cdot e^{j\pi} = -1$

$$\begin{cases} |H(z_2)| = \frac{|1+\sqrt{2}+1|}{|1+0.7\sqrt{2}+0.49|} \sim 1.38 \\ \angle H(z_2) = 0 \end{cases}$$

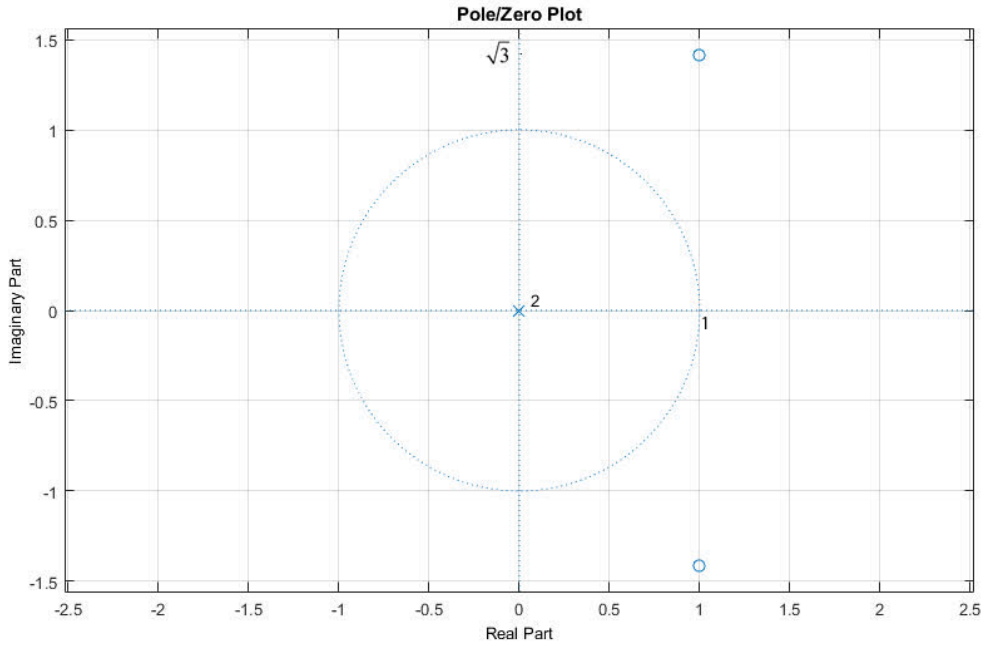
The output will be

$$y[n] = 5 \cdot 1.17 + 0 + 1.38 \cdot \cos(\pi n + 0)$$

$$y[n] = 5.85 + 1.38 \cdot (-1)^n$$

9.15 20160223-1

A filter $H_1(z)$ presents the following zero-pole plot:



Where the two zeros are in $1 \pm \sqrt{3}j$

9.15.1 Define minimum phase filter with fixed amplitude

Define a minimum phase filter $H_2(z)$ with **exactly** the same amplitude response

As two zeros are outside, we are dealing with maximum phase filter: we want to bring the zeros inside keeping the same amplitude, so we must **move zeros to their complex conjugate position, inside the circle**.
For example the zero on top-right has distance ~ 2, we move it inside the same distance segment closer to origin till distance=2^{-1}, then we flip it w.r.t. x-axis.
A couple of conjugate zeros with **negative** ρ, so we use:

$$1 - 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2} \leftrightarrow \rho(\cos(\theta) \pm j\sin(\theta))$$

$$H_1(z) = 1 - 2\rho \underbrace{\frac{1}{2}}_{\text{Cosine of } \theta \text{ for zero at } +\sqrt{3}} z^{-1} + \rho^2 z^{-2} = -2 \cdot 2\frac{1}{2} z^{-1} + 2^2 z^{-2}$$

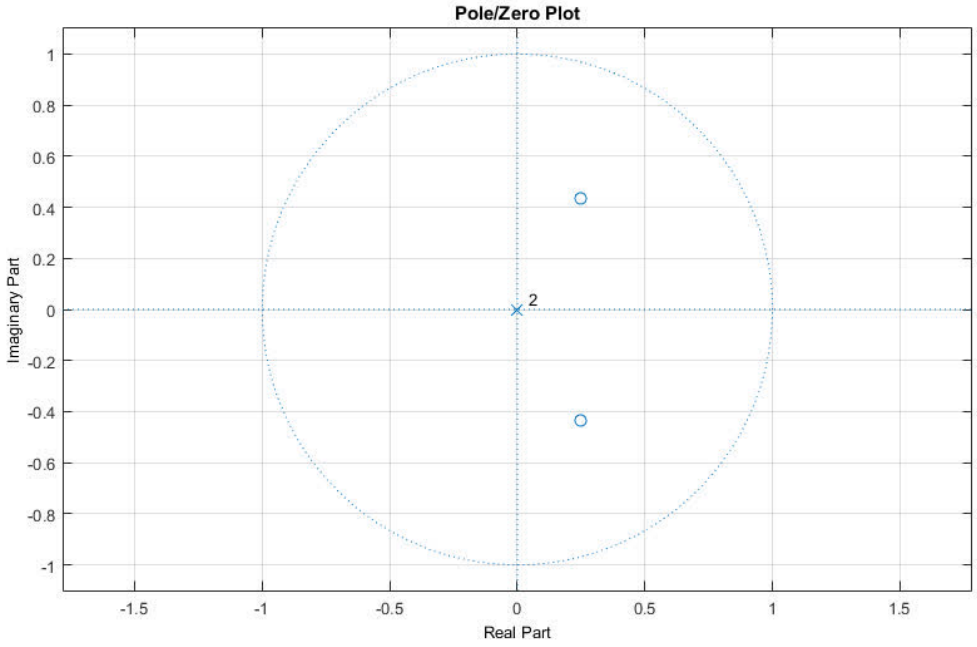
$$H_1(z) = 1 - 2z^{-1} + 4z^{-2}$$

Consider the conjugate (we put $\rho = 2^{-1} = \frac{1}{2}$)

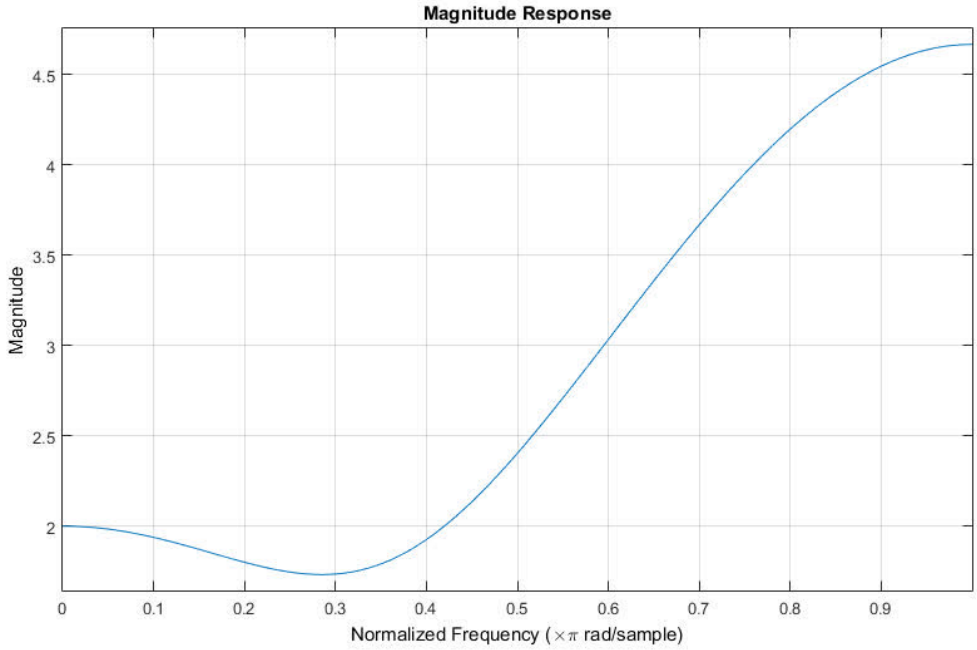
$$H_2(z) = A\left(1 - \frac{1}{2}z^{-1} + \frac{1}{4}z^{-4}\right)$$

This is the minimum phase filter. Imposing:

$$H_1(z=1,\omega=0) = H_2(z=1) \Rightarrow 3 = A \cdot \frac{3}{4} \Rightarrow A = 4$$



Magnitude is the same but with a different gain



9.15.2 IIR filter to remove effect

Define a pure IIR filter $H_3(z)$ that, placed after the filter $H_2(z)$ is able to completely remove its effect

Remove the zeros

$$H_3(z) = 4\frac{1}{1-\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}} = 4H_2(z)^{-1}$$

9.15.3 Output samples

Find the first five output samples of the filters $H_1(z), H_2(z), H_3(z)$

$$x[n] = \delta[n]$$

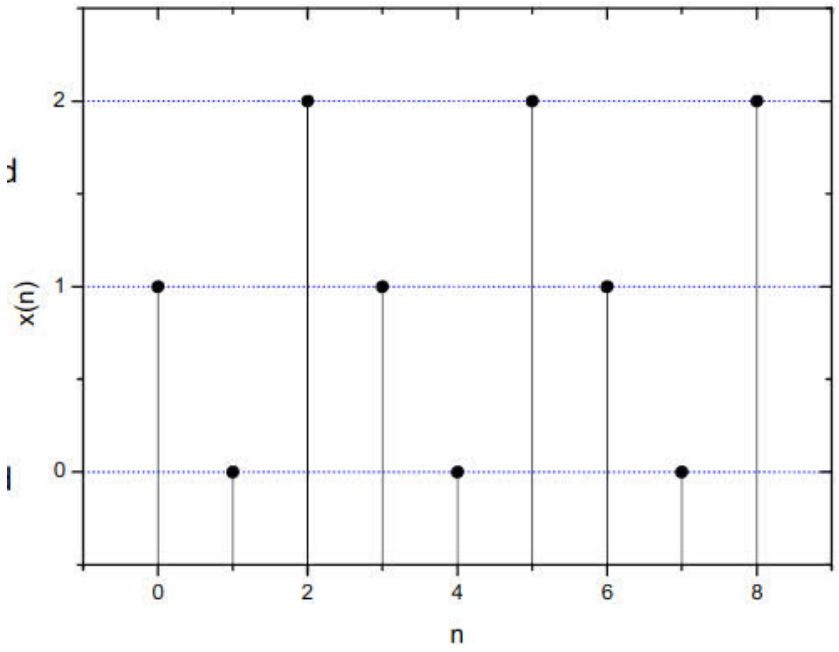
$$y_1[n] = x[n] - 2x[n-1] + 4x[n-2]$$

So: $\{1, -2, 4\}$. For y_2 solve similarly. For y_3:

$$y_3[n] = 4\left(\frac{1}{2}y[n-1] - \frac{1}{4}y[n-2]\right) + x[n]$$

9.16 20160223-2

We want to remove the continuous component (zero frequency) from the periodic signal $x(n)$ represented below just working with the DFT.



9.16.1 DFT to remove continuous component

Propose a filter $H(k)$ in the frequency domain and apply it to the signal (provide also the W matrix).

From the picture we can deduce:

$$x(n) = \delta(n) + 2\delta(n-2)$$

With period 3. Reminder, the W matrix is

$$W = \begin{bmatrix} W_i^0 & W_i^0 & W_i^0 & W_i^0 & \cdots & W_i^n \\ W_i^0 & W_i^1 & W_i^2 & W_i^3 & \cdots & W_i^{n-1} \\ W_i^0 & W_i^2 & W_i^4 & W_i^6 & \cdots & W_i^{2(n-1)} \\ W_i^0 & W_i^3 & W_i^6 & W_i^9 & \cdots & W_i^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ W_i^0 & W_i^{n-1} & W_i^{2(n-1)} & W_i^{3(n-1)} & \cdots & W_i^{(n-1)^2} \end{bmatrix}$$

Where $W_i = e^{-j\frac{2\pi}{i}}$. In our case $i = 3$

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix}$$

So the DFT of x(n) will be:

$$X(k) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ \sqrt{3}j \\ -\sqrt{3}j \end{bmatrix}$$

As we want to remove the continuous component, we want to put 3 to 0 (the conjugates are 1 frequency), so we choose a

$$H(k) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Applying it to the signal:

$$Y(k) = X(k) \cdot H(k) = \begin{bmatrix} 0 \\ \sqrt{3}j \\ -\sqrt{3}j \end{bmatrix}$$

9.16.2 DFT matrix inverse, output in time domain

Provide also the output in time domain $y(n)$ and a period of the filter in the time domain $h(n)$.

The inverse of W is its conjugate transpose divided by $N = 3$:

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \end{bmatrix} \rightarrow W^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix}$$

$$y(n) = W^{-1} \cdot Y(k) = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 0 \\ \sqrt{3}j \\ -\sqrt{3}j \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

$$h(n) = W^{-1} \cdot H(k) = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} + j\frac{\sqrt{3}}{2} & -\frac{1}{2} - j\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} - j\frac{\sqrt{3}}{2} & -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1/3 \\ -1/3 \end{bmatrix}$$

9.17 20110202-1

Consider a signal made of infinite repertition of the following 4 samples: $\{0;1;2;3\}$

9.17.1 DFT

[4pts] Define the DFT matrix and provide the 4 Fourier coefficients for this signal

$$W_4 = e^{-j\frac{2\pi}{4}} = e^{-j\frac{\pi}{2}}$$

$$W = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$X(k) = Wx = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

9.17.2 Output samples: convolution

[4pts] An high pass filter, whose z-transform is $1-z^{-1}$ is applied to the signal, what will be the samples of the fitlered signal?

$$h[n] = \delta[n] - \delta[n-1]$$

Careful, the signal is PERIODIC, so

$$x[n-1] = \{3;0;1;2;3\}$$

And

$$x[n] * h[n] = \{-3;1;1;1;-3\}$$

9.17.3 DFT

[4pts] What will be the DFT of the final signal?

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -3 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -4 \\ -4 \\ -4 \end{bmatrix}$$

Or compute the DFT of the filter h and then multiply element-wise y and h