

## T-61.3010 Digital Signal Processing and Filtering

## T-61.3010 Digitaalinen signaalinkäsittely ja suodatus

Exercise material for spring 2009 by professor Olli Simula and assistant Jukka Parviainen. Corrections and comments to [t613010@cis.hut.fi](mailto:t613010@cis.hut.fi), thank you!

This material is intended for "paper sessions" on Tuesdays 12-14 L, on Wednesdays 10-12 G, and on Thursdays 14-16 D, six times in spring 2009. Each problem [Pxx] refers to Problem xx in this material. Bring your own copy when coming to the session.

The course follows the book "Digital Signal Processing" by Sanjit K. Mitra. There are three different editions available, 3rd being the newest. Notation (*Mitra 2Ed Sec. 5.2 / 3Ed Sec. 4.2*) refers to the section 5.2 in the 2nd Edition (yellow cover) of Mitra's Book and to the section 4.2 in the 3rd Edition (blue, antenna). There is a brief correspondence table of three editions and errata lists in the course web pages <http://www.cis.hut.fi/Opinnot/T-61.3010/>. Course lecture slides by Olli Simula follow the third edition of Mitra's book.

Other books mentioned in these problems are Heikki Huttunen's book "Signaalinkäsittelyn menetelmät" (in Finnish) which is available from <http://www.cs.tut.fi/~hehu/>, "The Scientist and Engineer's Guide to Digital Signal Processing, Second Edition" by Steven W. Smith, available from <http://www.dspguide.com/>, and "Introduction to Digital Filters" by Julius O. Smith III in <http://ccrma.stanford.edu/~jos/filters/>. Other DSP and related courses in WWW can be found, e.g., Connexions portal <http://cnx.org/> and MIT OpenCourseWare <http://ocw.mit.edu/>.

Index list in the end contains some (problems not fully indexed) terms both in Finnish and English. Terminology lists related to DSP can be found in the web, e.g., "Audiosignaalinkäsittelyn sanasto" by professor Vesa Välimäki <http://www.acoustics.hut.fi/~vpv/ask-sanasto.htm>, an index list by J. Smith in <http://ccrma.stanford.edu/~jos/filters/Index.html>, and a glossary by S. Smith in his book at pages 631-642.

## Contents

Description of Example Problems	2
Notations	5
Example Problems with Solutions	6
Formula tables	166
Index	170

*This copy belongs to:*

## T-61.3010 Digital Signal Processing and Filtering

## Description of Problems

#	Subject
<b>A: Math Background 1-13</b>	
1	complex numbers, Cartesian and polar coordinate systems, Euler's formula
2	Euler's formula, cosine and sine, odd and even functions
3	complex numbers, graphical notation
4	complex-valued function
5	phasors
6	cosine function, amplitude, frequency, phase
7	logarithm, decibels, sinc, modulo, binary number representation
8	roots of a polynomial
9	complex-valued function, roots of polynomial
10	partial fraction expansion / decomposition
11	sum of geometric series
12	integral transforms
13	matrix product
<b>B: Preliminary DSP 14-18</b>	
14	analog, discrete-time and digital signal
15	magnitude/amplitude response of filters LP, HP, BP, BS
16	digital signals and spectra, spectrogram
17	Fourier series, Fourier transforms: CTFT, DTFT, DFT
18	time-frequency-domain analysis and filtering
<b>C: Discrete-time Signals and Systems in Time-domain 19-35</b>	
19	signals and sequences, unit impulse and unit step functions ( $\delta[n]$ , $\mu[n]$ )
20	periodic signals
21	moving average (MA) filter, a simple FIR filter
22	a simple IIR filter
23	flow / block diagram of a discrete-time system
24	recognition of LTI systems, causal LTI systems, filter order, FIR, IIR
25	properties of LTI systems: linear, time-invariant, causal, stable
26	shifted and scaled sequences in LTI system
27	impulse response $h[n]$ , FIR, IIR
28	step response $s[n]$
29	linear convolution $y(t) = x_1(t) \otimes x_2(t)$ of continuous-time signals
30	linear convolution $y[n] = h[n] \otimes x[n]$ of discrete-time signals
31	convolution as products of polynomials
32	deconvolution
33	parallel and cascade (series) LTI systems
34	matched filter
35	auto- and cross-correlation
<b>D: Discrete-time Signals in Frequency-domain 36-42</b>	
continued on next page	

continued from previous page	
#	Subject
36	continuous-time Fourier transform (CTFT)
37	spectrum, CTFT, discrete-time Fourier transform (DTFT), discrete Fourier transform (DFT)
38	DTFT, computation from definition
39	DTFT, using a transform table
40	$2\pi$ -periodic spectrum, DTFT
41	magnitude/amplitude response, periodicity of DTFT
42	analysis of LTI FIR system: frequency, amplitude, phase response, group delay
<b>E: Digital Processing of Continuous-Time Signals 43-49</b>	
43	sampling, Shannon's theorem
44	impulse train and Fourier-series
45	impulse train and sampling in frequency-domain
46	sampling in frequency-domain
47	aliasing
48	sampling, aliasing, anti-aliasing
49	anti-aliasing filter
<b>F: Finite-Length Discrete Transforms 50-52</b>	
50	DFT, matrix product
51	circular shift, DFT
52	circular convolution
<b>G: Discrete-Time Systems in Frequency-domain 53-57</b>	
53	analysis of LTI IIR system, transfer function, convolution theorem, partial fraction expansion
54	amplitude response grafically from pole-zero-plot
55	analysis of LTI IIR system, pole-zero plot
56	transfer function, region of convergence (ROC)
57	scaling factor
<b>H: Filter Types 58-61</b>	
58	linear-phase FIR filters
59	filter types: allpass, zero-phase, linear-phase, minimum-phase, maximum-phase
60	minimum-phase FIR filter, inverse filter
61	parallel system
<b>I: Digital Filter Structures 62-65</b>	
62	LTI subsystems
63	direct form (DF) structures
64	allpass filter
65	polyphase structure
<b>J: IIR Digital Filter Design 66-68</b>	
66	filter specifications
67	analog filter approximations
68	bilinear transform and impulse-invariant method in digital filter design
<b>K: FIR Digital Filter Design 69-71</b>	
69	optimal finite-length approximation of ideal infinite-length impulse response
70	FIR-window method in digital filter design
71	computational comparisons between IIR and FIR filters
continued on next page	

continued from previous page	
#	Subject
<b>L: DSP Algorithm Implementation 72-76</b>	
72	computational set of equations, precedence graph
73	FFT computational complexity
74	radix-2 DIT FFT algorithm
75	binary addition and subtraction, two's complement
76	fixed-point binary number representations
<b>M: Analysis of Finite Wordlength Effects 77-80</b>	
77	quantization, error densities
78	roundoff noise in FIR filters
79	signal-to-noise ration (SNR)
80	error-feedback structure
<b>N: Multirate Digital Signal Processing 81-86</b>	
81	up- and downsampling in time- and frequency domain
82	multirate system analysis
83	linearity of up- and downsampling systems
84	filter bank
85	interpolated FIR filter (IFIR), FIR window method design

T-61.3010 Digital Signal Processing and Filtering

Notations

Notation	Explanation	suomeksi
$x[n]$	input sequence	syötesekvenssi
$y[n]$	output sequence	vasteesekvenssi
$h[n]$	impulse response	impulssivaste
$s[n]$	step response	askelvaste
$\delta[n]$	unit impulse function	yksikköimpulssifunktio
$\mu[n]$	unit step function	yksikköaskelfunktio
$j, i$	imaginary unit $\sqrt{-1}$ , $j$ is preferred in DSP	imaginaariyksikkö
$z, z^*$	complex number $z$ and its complex conjugate	kompleksiluku ja sen liittoluku
$X(e^{j\omega})$	input spectrum	syöteen spektri
$Y(e^{j\omega})$	output spectrum	vasteen spektri
$H(e^{j\omega})$	frequency response of a digital filter	taajuusvaste
$ H(e^{j\omega}) $	magnitude (or amplitude response) of a digital filter	magnitudivaste (tai amplitudivaste)
$A(e^{j\omega})$	amplitude response of a digital filter	amplitudivaste
$\angle H(e^{j\omega})$	phase response of a digital filter	vaiheaste
$\tau(\omega)$	group delay function	ryhmäviive
$H(z)$	transfer function of a filter	siirtofunktio
$H(j\Omega)$	frequency response of an analog filter	taajuusvaste
$f$	frequency, $[f] = \text{Hz} = 1/\text{s}$	taajuus
$\Omega$	angular frequency for continuous-time signals, $\Omega = 2\pi f$ , $[\Omega] = \text{rad/s}$	kulmataajuus
$\omega$	normalized angular frequency for discrete-time sequences, $\omega = 2\pi\Omega/\Omega_T = 2\pi f/f_T$ , $[\omega] = \text{rad/sample}$	normalisoitu kulmataajuus
$f_s, f_T$	sampling frequency	näytteenottotaajuus
$f_{\text{stop}}, f_c$	cut-off frequency for stopband	estokaistan rajataajuus
$f_{\text{pass}}, f_p$	cut-off frequency for passband	päästökaistan rajataajuus
$f_{\text{MATLAB}}$	normalized frequency in Matlab, $f_{\text{MATLAB}} = 2f/f_T$	Matlabissa normalisoitu taajuus
$W_N$	$W_N = e^{j2\pi/N}$ , a frequency component in DFTs	DFT:ssä käytetty komponentti

T-61.3010 Digital Signal Processing and Filtering

Example problems for spring 2009. Corrections and comments to t613010@ics.tkk.fi!

Problems and Solutions

1. **Problem:** Complex numbers and trigonometric functions.

- Express  $z = 2e^{-j\pi}$  in Cartesian coordinates.
- Express  $z = -1 + 2j$  in polar coordinates.
- Which angles satisfy  $\sin(\omega) = 0.5$ ?
- What are  $z + z^*$ ,  $|z + z^*|$ ? and  $\angle(z + z^*)$ ? What are  $zz^*$ ,  $|zz^*|$ ? and  $\angle zz^*$ ?

[L0130] **Solution:** Complex numbers in the *Cartesian coordinate system* (or rectangular coordinate system) are given with  $z = x + jy$ , where  $j$  or  $i$  is the *imaginary unit*  $j \equiv \sqrt{-1}$ . The same in the polar coordinate system is  $z = r \cdot e^{j\theta}$ , where  $r$  is distance to origin (radius) and  $\theta$  angle, see the formula table on page 166. The *complex conjugate*  $z^*$  (or  $\bar{z}$ ) is  $z^* = x - jy = r \cdot e^{-j\theta}$  *Euler's formula* decomposes complex *exponential function* to cosine and sine,  $e^{j\omega} = \cos(\omega) + j \sin(\omega)$ . Remember also that 360 degrees (whole circle) corresponds  $2\pi$  in *radians*.

a) "Brute force" using Euler's formula and  $\cos(-x) = \cos(x)$  and  $\sin(-x) = -\sin(x)$ ,

$$z = 2e^{-j\pi} = 2(\cos(-\pi) + j \sin(-\pi)) = 2(\cos(\pi) - j \sin(\pi)) = -2$$

or using directly the unit circle and seeing that when the angle is  $-\pi$  in radians ( $-180$  degrees) then  $e^{-j\pi} = -1$ .

b) The radius  $r = \sqrt{(-1)^2 + 2^2} = \sqrt{5} \approx 2.2$  and the angle in radians  $\theta = \pi - \arctan(2/1) \approx 2.03 \approx 0.65\pi$ . So,  $z = -1 + 2j = \sqrt{5}e^{j(\pi - \arctan(2))} \approx 2.2e^{j0.65\pi}$ . Note! Always check the right quarter in the figure.

c) In Figure 1(b) a dashed line  $y = 0.5$  gives  $\omega_1 = \arcsin(0.5) = \pi/6$  and  $\omega_2 = \pi - \arcsin(0.5) = 5\pi/6$ , or any  $2\pi$ -multiple.

d) Summing is concatenation of vectors.  $z + z^* = r(e^{j\omega} + e^{-j\omega}) = 2r \cos(\omega) \in \mathbb{R}$ . From previous,  $|z + z^*| = |2r \cos(\omega)|$  and  $\angle(z + z^*) = 0$ . Using Cartesians,  $z + z^* = 2r$ . Product of complex number and its complex conjugate:  $zz^* = (re^{j\omega})(re^{-j\omega}) = r^2e^{j(\omega-\omega)} = r^2$ , and  $|zz^*| = r^2$  and  $\angle zz^* = 0$ .

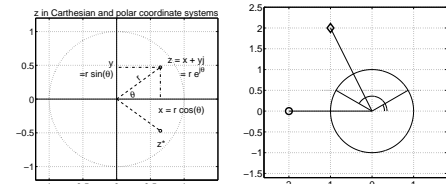


Figure 1: Problem 1, unit circle in complex plane (left), and points for (a), (b), and (c) (right).

2. **Problem:** Examine the connection between exponential functions and cosines and sines.

- Express with cosines and sines:  $e^{j\theta} + e^{j(-\theta)}$ .
- Express with cosines and sines:  $e^{j\theta} - e^{j(-\theta)}$ .
- Express with cosines and sines:  $e^{j\pi/8} \cdot e^{j\theta} - e^{j(-\pi/8)} \cdot e^{j(-\theta)}$ .

[L0140] **Solution:** *Euler's formula*  $e^{j\theta} = \cos(\theta) + j \cdot \sin(\theta)$  can be thought as a *phasor* going around on the unit circle, see Problem 5. It is an unit circle because  $|e^{j\theta}| = \sqrt{\cos^2 + \sin^2} = 1$  always. Cosine is an *even function*  $f(x) = f(-x)$ , whereas sine is *odd function*  $f(x) = -f(-x)$ . Real part of  $e^{j\theta}$  is cosine, and imaginary part is sine.

a) Sum of exponentials at positive frequency  $\theta$  and negative frequency  $-\theta$  gives a real cosine at frequency  $\theta$ :

$$\begin{aligned} e^{j\theta} &= \cos(\theta) + j \cdot \sin(\theta) \\ e^{j(-\theta)} &= \cos(-\theta) + j \cdot \sin(-\theta) = \cos(\theta) - j \cdot \sin(\theta) \\ e^{j\theta} + e^{j(-\theta)} &= 2 \cos(\theta) \in \mathbb{R} \\ \cos(\theta) &= 0.5 \cdot e^{j\theta} + 0.5 \cdot e^{j(-\theta)} \end{aligned}$$

b) In the same way as in (a) but subtracting the last from the first gives

$$\begin{aligned} e^{j\theta} &= \cos(\theta) + j \cdot \sin(\theta) \\ e^{j(-\theta)} &= \cos(-\theta) + j \cdot \sin(-\theta) = \cos(\theta) - j \cdot \sin(\theta) \\ e^{j\theta} - e^{j(-\theta)} &= 2j \sin(\theta) \in \mathbb{C} \\ \sin(\theta) &= \frac{1}{2j} \cdot e^{j\theta} - \frac{1}{2j} \cdot e^{j(-\theta)} = -0.5j \cdot e^{j\theta} + 0.5j \cdot e^{j(-\theta)} \end{aligned}$$

where  $\frac{1}{2j} = -\frac{j}{2}$  as shown in Problem 3(e).

c) This can be thought as phase shift. First, use the rule  $e^x \cdot e^y = e^{x+y}$ ,

$$\begin{aligned} e^{j\theta} \cdot e^{j\pi/8} &= e^{j(\theta+\pi/8)} \\ e^{-j\theta} \cdot e^{-j\pi/8} &= e^{-j(\theta+\pi/8)} \end{aligned}$$

Now, we see using (b)

$$e^{j\pi/8} \cdot e^{j\theta} - e^{j(-\pi/8)} \cdot e^{j(-\theta)} = 2j \sin(\theta + \pi/8)$$

Notice that each sinusoidal (cos, sin) can be replaced by two complex exponentials with a positive and a corresponding negative angle. When considering Fourier analysis, a real cosine signal with frequency  $f_c$  can be represented in the spectrum with a peak at  $f_c$  (in one-sided spectrum) or with peaks at  $f_c$  and  $-f_c$  (in two-sided spectrum). If the two-sided spectrum is not symmetric, then the signal is surely not real but complex. More about this later in Fourier analysis, see Problem 37 and 41.

3. **Problem:** Consider the following three complex numbers  $z_1 = 3 + 2j$ ,  $z_2 = -2 + 4j$ , and  $z_3 = -1 - 5j$ .

- Draw the vectors  $z_1$ ,  $z_2$ , and  $z_3$  separately in the complex plane.
- Draw and compute the sum  $z_1 + z_2 + z_3$ .
- Draw and compute the weighted sum  $z_1 - 2z_2 + 3z_3$ .
- Draw and compute the product  $z_1 \cdot z_2 \cdot z_3$ .
- Compute and reduce the division  $z_1/z_2$ .

[L0138] **Solution:**

a) Each *complex number* can be thought as a vector starting from origin and the other end at point  $z$ . See Figure 2(a).

b) Real parts and imaginary parts can be summed separately  $z = (3 - 2 - 1) + (2 + 4 - 5)j = j$ . This can be expressed in the polar coordinates  $z = e^{j(\pi/2)}$ , i.e. on the unit circle (radius 1) and the angle one fourth a circle (90 degrees,  $\pi/2$ ) counterclockwise.

c) If you are computing without computer, be attentive and check twice that all coefficients are correctly reduced.  $z = (3 + 2j) - 2(-2 + 4j) + 3(-1 - 5j) = 4 - 21j$ . Again, in the polar coordinates  $r = \sqrt{(4)^2 + (-21)^2} \approx 21.38$ . The angle  $\theta = \arctan((-21)/(4)) \approx -1.38 \approx -0.44\pi$  ( $-79^\circ$ ).

If  $z = -4 - 21j$ , then the calculator gives  $\theta = \arctan((-21)/(-4)) \approx 1.38 \approx 0.44\pi$  ( $+79^\circ$ ), which is incorrect (**atan** $(-21/-4)$ ). Notice that now  $z$  is in the third quarter and  $\pm\pi$  has to be added. The correct angle is  $\theta = \arctan((-21)/(-4)) - \pi \approx -1.76 \approx -0.56\pi$  ( $-101^\circ$ ), in Matlab either **atan2** $(-21, -4)$  or **angle** $(-4-21j)$ .

d) When using Cartesian coordinates, multiply terms normally and apply  $j^2 = -1$ , e.g.,  $(3 + 2j) \cdot (-2 + 4j) = -6 + 12j - 4j + 8j^2 = -14 + 8j$ . The product in the polar coordinates means multiplying the lengths of vectors and summing the angles.

$$\begin{aligned} z &= (3 + 2j) \cdot (-2 + 4j) \cdot (-1 - 5j) = 54 + 62j \approx 82.2 \cdot e^{j(0.27\pi)} \\ &= \sqrt{9+4} \cdot \sqrt{4+16} \cdot \sqrt{1+25} \cdot e^{j(\arctan(2/3)+\arctan(4/-2)+\arctan(-5)/(-1))} \end{aligned}$$

e) The denominator is now complex. If both sides are multiplied by the *complex conjugate* of the denominator then the denominator becomes real. Just as in Problem 1  $z \cdot z^* = |z|^2 = r^2 \in \mathbb{R}$ . Notice also that  $1/j$  is  $-j$  because  $(1/j) \cdot (j/j) = j/j^2 = -j$ .

$$\begin{aligned} z &= (3 + 2j)/(-2 + 4j) \quad | \quad (-2 - 4j)/(-2 - 4j) \\ &= (2 - 12j)/20 \end{aligned}$$

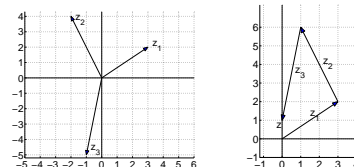


Figure 2: The vectors in Problem 3(a) and (b).

4. **Problem:** Examine a complex-valued function

$$H(\omega) = 2 - e^{-j\omega}$$

where  $\omega \in [0 \dots \pi] \in \mathbb{R}$ .

- Compute values of Table 1 with a calculator. Euler:  $e^{j\omega} = \cos(\omega) + j \sin(\omega)$ .
- Draw the values at  $\omega = \{0, \pi/4, \dots, \pi\}$  into complex plane  $(x, y)$ . Interpolate smoothly between the points.
- Sketch  $|H(\omega)|$  as a function of  $\omega$ . Interpolate smoothly.
- Sketch  $\angle H(\omega)$  as a function of  $\omega$ . Interpolate smoothly.

$\omega$	$x = \text{Real}(H(\omega))$	$y = \text{Imag}(H(\omega))$	$r =  H(\omega) $	$\theta = \angle H(\omega)$
0				
$\pi/4$				
$\pi/2$				
$3\pi/4$				
$\pi$				

Table 1: Problem 4: values of a complex-valued function in rectangular  $(x, y)$  and polar  $(r, \theta)$  coordinates.

[L0131] **Solution:** In this course *complex-valued functions* are widely used, e.g. as frequency responses of the systems or in Fourier transforms. The argument of the function is real-valued  $\omega \in \mathbb{R}$ , but the value of the function is generally complex  $H(\omega) \in \mathbb{C}$  due to complex factor  $e^{j\omega}$ . In case of the transfer function  $H(z)$  both the argument  $z$  and the function  $H(z)$  are complex-valued.

- Sometimes it is possible to simplify  $H(\omega)$ . However, normally it is useful to write down a suitable format for the use of the calculator. In this case, Cartesian coordinate system with  $x$  and  $y$  is used:

$$H(\omega) = 2 - e^{-j\omega} = 2 - (\cos(-\omega) + j \sin(-\omega)) = \underbrace{2 - \cos(\omega)}_x + \underbrace{j \sin(\omega)}_y$$

The variables  $r$  and  $\theta$  of the polar coordinate system are received from the right-angled triangle:  $r = \sqrt{x^2 + y^2}$  and  $\theta = \arctan(y/x)$ .

On the other hand, in this case it is easily seen that there is only a circle ( $e^{-j\omega}$ ) whose origin is shifted to  $z = 2$ .

- Take the columns  $x$  and  $y$  of Table 2 and sketch the curve like in Figure 3(a). There is a line drawn in the plot, from the origin to a point related to  $\omega = 3\pi/4$ , i.e.  $(x, y)$ . The length of the line is  $r$  and the angle between the line and x-axis is  $\theta$ , so it can be written in polar coordinates  $r e^{j\theta}$ .
- Take the column  $r$  of Table 2 and sketch the curve like in Figure 3(b). The plot shows the distance  $r$  from the origin to a point at given value of  $\omega$ .
- Take the column  $\theta$  of Table 2 and sketch the curve like in Figure 3(c). The plot shows the angle  $\theta$  between the origin and a point at given value of  $\omega$ .

$\omega$	$x = \text{Real}(H(\omega))$	$y = \text{Imag}(H(\omega))$	$r =  H(\omega) $	$\theta = \angle H(\omega)$
0	1.0000	0	1.0000	0
$\pi/4$	1.2929	0.7071	1.4736	0.1593 $\pi$
$\pi/2$	2.0000	1.0000	2.2361	0.1476 $\pi$
$3\pi/4$	2.7071	0.7071	2.7979	0.0813 $\pi$
$\pi$	3.0000	0	3.0000	0

Table 2: Problem 4: values of a complex-valued function in rectangular  $(x, y)$  and polar  $(r, \theta)$  coordinates. The row  $3\pi/4$  is highlighted for Figure 3.

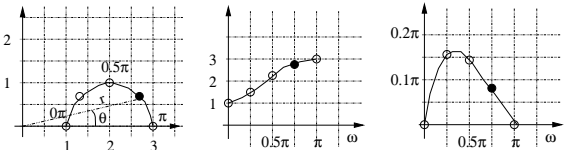


Figure 3: Problem 4: Plots of a complex-valued function, (a)  $H(\omega)$  in complex plane, (b) absolute values  $|H(\omega)|$ , and (c) angle  $\angle H(\omega)$ . The case when  $\omega = 3\pi/4$  is highlighted.

5. **Problem:** Compute values of a complex-valued function  $H(\omega)$

$$H(\omega) = 3e^{-j\omega} + 2e^{-2j\omega} - e^{-3j\omega}$$

at  $\omega = \{0, \pi/2, \pi\}$  considering each complex exponential function as a phasor – not explicitly applying Euler’s formula (decomposition to cosines and sines).

[L0165] **Solution:** The exponential function  $f(\omega) = A e^{j\omega + \theta}$ , where  $A$  is radius and  $\theta$  phase shift, can be considered as a *phasor*, which draws counterclockwise a circle with radius  $A$  when  $\omega = 0 \dots 2\pi$ . The starting and ending point is  $A e^{j\theta}$ .

There is a very nice Java applet demo “Harmonic phasors and Fourier series” in <http://www.jhu.edu/~signals/phasorlecture2/indexphasorlect2.htm>. Choose “Harmonic Phasor Sums” and click positions (3, 0), (5, 0), and (4, 0) in the complex plane for the starting point and press Play button. This phasor rotates counterclockwise ( $e^{j\omega}$  instead of  $e^{-j\omega}$ ). A similar Matlab program `phasor.m` can be found in the course web site. This problem requires a command `phasor([3 2 -1], -1)`;

Consider each exponential as a phasor with radius 3, 2, and 1, starting points (3, 0), (2, 0), and  $(-1, 0)$ , angular frequencies  $-\omega$ ,  $-2\omega$ , and  $-3\omega$ , and figure out / compute the values  $H(\omega) = \{4, -2 - 4j, 0\}$  at  $\omega = \{0, \pi/2, \pi\}$ :

	$3e^{-j\omega}$	$2e^{-2j\omega}$	$-e^{-3j\omega}$	$H(\omega)$
$\omega = 0$	3	2	-1	4
$\omega = \pi/2$	$-3j$	-2	$-j$	$-2 - 4j$
$\omega = \pi$	-3	2	1	0

These values are illustrated in Figure 4, where first three columns are individual phasors and the last column the phasor sum  $H(\omega)$ . The first row is the initial case at  $\omega = 0$ . The second row shows that the first phasor has moved one quarter clockwise ( $-$ ), second

phasor two quarters ( $-2\omega$ ), and third phasor three quarters ( $-3\omega$ ), i.e., moving three times “faster” than the first phasor.

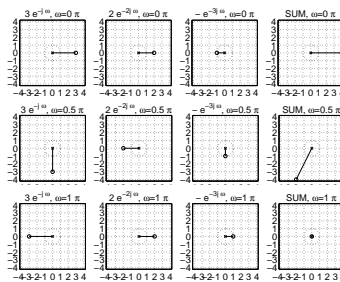


Figure 4: Problem 5: Columns: Phasors and their sum. Rows:  $\omega = 0, \omega = 0.5\pi$ , and  $\omega = \pi$ .

6. **Problem:** A cosine signal can be represented using its angular frequency  $\Omega$  or frequency  $f$ , amplitude  $A$  and phase  $\theta$ :

$$x(t) = A \cos(\Omega t + \theta) = A \cos(2\pi f t + \theta)$$

- Estimate  $A, f, \theta$  for the cosine  $x_1(t)$  in Figure 5(a).
- Sketch a cosine  $x_2(t)$ , with  $A = 2$ , angular frequency 47 rad/s and angle  $-\pi/2$ .
- Express  $x_2(t)$  in (b) using exponential functions.

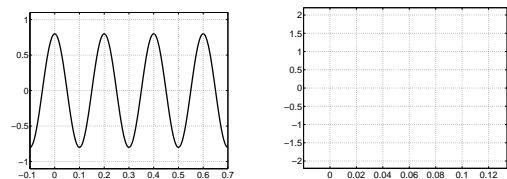


Figure 5: Cosine  $x_1(t)$  (left) and  $x_2(t)$  (right) in Problem 6.

[L0132] **Solution:** There is a lot of variation in symbols in different signal processing books and texts. There is probably also variation in these exercises, however, we try to use the following symbols listed on page 5 and in Table 3 below.

symbol	units	meaning
$f$	Hz	frequency
$\Omega$	rad/s	angular frequency, $\Omega = 2\pi f$
$\omega$	rad	normalized angular frequency, $\omega = 2\pi(\Omega/\Omega_s)$
$f_{MATLAB}$	1	normalized Matlab frequency, $f_{MATLAB} = 2f/f_s$

Table 3: Problem 6, symbols of frequencies.  $f_s$  refers to sampling frequency, and  $\Omega_s = 2\pi f_s$ .

A cosine signal can be represented using its *angular frequency*  $\Omega$  or frequency  $f$ , amplitude  $A$  and phase  $\theta$ :

$$x(t) = A \cos(\Omega t + \theta) = A \cos(2\pi f t + \theta)$$

For a discrete sequence of numbers

$$x[n] = x(t)|_{t=nT_s} = x(t)|_{t=n/f_s} = A \cos(2\pi(f/f_s)n + \theta) = A \cos(\omega n + \theta)$$

where  $T_s$  is sampling interval (period),  $f_s$  sampling frequency, and  $\omega$  *normalized angular frequency*.

- Cosine oscillates between  $-0.8$  and  $0.8$ , so  $A = 0.8$ . There is no phase shift,  $\theta = 0$ . There is one oscillation in 0.2 seconds, so there are 5 periods in one second,  $f = 5$  Hz, or  $\Omega = 2\pi f = 10\pi$  rad/s. Hence,  $x_1(t) = 0.8 \cos(10\pi t)$ .

- b)  $x_2(t)$  can be written directly  $x_2(t) = 2 \cos(47t - \pi/2)$ .  
 Now  $\Omega = 47 \text{ rad/s}$ , hence  $f \approx 7.5 \text{ Hz}$  and the period  $T = 1/f \approx 0.13 \text{ s}$ . At  $t = 0$ ,  $x_2(0) = 2 \cos(-\pi/2) = 0$ , and increasing. Note that  $\cos(\Omega t - \pi/2) \equiv \sin(\Omega t)$ . Amplitude  $A = 2$ , so the curve oscillates between  $-2$  and  $2$ . The curve is plotted in Figure 6(b).  
 c) Using Euler's formula, and properties of cosine (even function  $f(-x) = f(x)$ ) and sine (odd function  $f(-x) = -f(x)$ ),

$$\begin{aligned} e^{j\omega} &= \cos(\omega) + j \sin(\omega) \\ + e^{-j\omega} &= \cos(\omega) - j \sin(\omega) \\ \hline e^{j\omega} + e^{-j\omega} &= 2 \cos(\omega) \end{aligned}$$

$$\begin{aligned} e^{j\omega} &= \cos(\omega) + j \sin(\omega) \\ + -e^{-j\omega} &= -\cos(\omega) + j \sin(\omega) \\ \hline e^{j\omega} - e^{-j\omega} &= 2j \sin(\omega) \end{aligned}$$

Now, it can be seen that

$$\begin{aligned} x_2(t) &= 2 \cos(47t - \pi/2) \\ &= e^{j(47t - \pi/2)} + e^{-j(47t - \pi/2)} \end{aligned}$$

which can be even "simplified" (?) to  $x_2(t) = j[e^{-j47t} - e^{j47t}]$ .

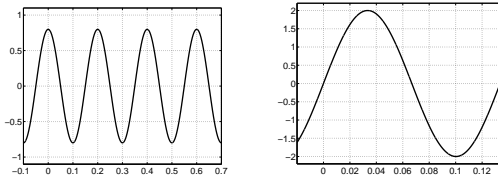


Figure 6: Cosine  $x_1(t)$  (left) and  $x_2(t)$  (right) in Problem 6.

7. **Problem:** Some elementary functions and notations.

- Compute with a calculator:  $\log_8 7$ .
- The power of signal is attenuated from 10 to 0.01. How much is the attenuation in decibels?
- Sketch the curve  $p(x) = \sum_{k=-N}^{+N} kx$  for various  $N$ .
- Sinc-function is useful in the signal processing. It is defined  $\text{sinc}(x) = \sin(\pi x)/(\pi x)$ . Also it is known that  $\sin(x)/x \rightarrow 1$ , when  $x \rightarrow 0$ , and with sinc-function  $\text{sinc}(0) = 1$ . Consider  $h(n) = \sin(0.75\pi n)/(\pi n)$ . What is  $h(0)$ ?
- What is the binary number (1001011)<sub>2</sub> as a decimal number?
- What is the hexadecimal number 0x01EF3A as a decimal number?
- Modulo- $N$  operation for number  $x$  is written here as  $\langle x \rangle_N$ . What is  $\langle -5 \rangle_3$ ?

[L0133] **Solution:**

- logarithm:*  $\log_8 7 = \log_e 7 / \log_e 8 \approx 1.9459 / 2.0794 \approx 0.936$ .  
 Sometimes it is useful to convert, e.g.,  $2^{2008}$  to decimal base:  $2^{2008} = 10^{\tau}$ , taking  $\log_{10}$  on both sides:  $x = 2008 \log_{10} 2 \approx 604.4682$ . Now  $10^{0.4682} \approx 2.9392$ , which finally gives  $2^{2008} \approx 2.9 \cdot 10^{604}$ . Remember also  $2^{10} = 1024 \approx 1000$  (k),  $2^{20} = 1048576 \approx 10^6$  (M), etc. For instance,  $2^{32} = 2^2 \cdot 2^{30} \approx 4\text{G}$ .
- Decibel scales* are widely used to compare two quantities. The decibel difference between two power levels,  $\Delta L$ , is defined in terms of their power ratio  $W_2/W_1$  (p. 99, Rossing et al., The Science of Sound, 3rd Edition, Addison Wesley)

$$\Delta L = L_2 - L_1 = 10 \log_{10} W_2/W_1$$

Now the power (square) of signal is attenuated from 10 to 0.01, so the signal is attenuated by 30 dB:

$$10 \log_{10}(0.01/10) = 10 \log_{10} 10^{-3} = -30$$

In case of computing amplitude response  $|H(e^{j\omega})|$ , e.g. in Matlab directly from the equation or with the command `freqz`, the values are squared for decibels

$$10 \log_{10} (|H/H_0|)^2 = 20 \log_{10} (|H/H_0|)$$

- If  $\Sigma$  confuses, open the expression! There is hardly anything to draw!  

$$p(x) = \sum_{k=-N}^{+N} kx = (-N)x + \dots + (-2)x + (-1)x + 0x + x + 2x + \dots + Nx \equiv 0, \forall N, x$$
- sinc-function* is very useful in the signal processing, and it is defined  $\text{sinc}(x) = \sin(\pi x)/(\pi x)$ . Also it is known that  $\sin(x)/x \rightarrow 1$ , when  $x \rightarrow 0$ , and with sinc-function  $\text{sinc}(0) = 1$ . Fourier-transform of a rectangle (box) signal produces a spectrum with shape of sinc-function, and vice versa, a signal like sinc-function has a spectrum of rectangular shape.  
 Note that the result of the problem is not 1 nor 0,

$$\begin{aligned} h(n) &= \sin(0.75\pi n)/(\pi n) = 0.75 \sin(0.75\pi n)/(0.75\pi n) = 0.75 \text{sinc}(0.75n) \\ h(0) &\rightarrow 0.75 \end{aligned}$$

- The result depends on which number representation is chosen. In case of multi-byte data types numbers can be saved in big-endian or little-endian manner. DSPs are divided to fixed-point and floating-point processors (*IEEE 754*, sign bit, exponent and mantissa fields). Least significant bit (LSB) is normally the last bit, most significant bit (MSB) leftmost. Negative numbers and fractions has to be considered, too. (*Mitra 2Ed Sec. 8.4 / 3Ed Sec. 11.8*) deals with all aspects of the *number representation*.

When both negative and positive  $b$ -bit fraction values are needed, 1001011 is considered to have a sign bit first, and then fraction bits, like  $s_{\Delta}a_{-1}a_{-2} \dots a_{-b}$ . Table 4 contains some possible results with values  $b = 6$  and  $s = 1$ , see also (*Mitra 2Ed Table 8.1, p. 557 / 3Ed Table 11.1, p. 638*).

non-negative fixed-point	1001011	$1 \cdot 64 + 1 \cdot 8 + 1 \cdot 2 + 1 \cdot 1 = 75$
sign-magnitude	1 <sub>Δ</sub> 001011	$(-2s + 1) \sum_{i=1}^b a_{-i} 2^{-i} = -11/64 \approx -0.1719$
ones' complement	1 <sub>Δ</sub> 001011	$-s \cdot (1 - 2^{-b}) + \sum_{i=1}^b a_{-i} 2^{-i} = -52/64 \approx -0.8125$
two's complement	1 <sub>Δ</sub> 001011	$-s + \sum_{i=1}^b a_{-i} 2^{-i} = -53/64 \approx -0.8281$
offset binary	1 <sub>Δ</sub> 001011	$+11/64 \approx +0.1719$

Table 4: Problem 7: Examples on binary number representations with values  $b = 6$  and  $s = 1$ .

- In some programming languages a prefix 0x is reserved for *hexadecimals*, 16-base numbers, where {A, B, C, D, E, F} = {(10)<sub>10</sub>, (11)<sub>10</sub>, ..., (15)<sub>10</sub>}. Now, 0x01EF3A is  $0 \cdot 16^5 + 1 \cdot 16^4 + 14 \cdot 16^3 + 15 \cdot 16^2 + 3 \cdot 16 + 10 = 126778$ . If considering RGB colors, then each component has two hexas, 01.EF.3A, meaning values red = 1 (1/255), green 239, and blue 58. A similar approach for IP addresses (four 2-hexas).
- See also "circular shift of a sequence" (*Mitra 2Ed Sec. 3.4.1, p. 140 / 3Ed Sec. 5.4.1, p. 244*). In the modulo operation,  $(m \bmod N)$ , or,  $\langle m \rangle_N$ , we want to compute the remainder (or residue)  $r$  when  $m$  is divided by  $N$ , that is,  $m = r + kN$ , so that the remainder is always in the range  $0 \dots N - 1$ . Now for  $\langle -5 \rangle_3$  we have the equation with  $m = -5$  and  $N = 3$  as  $-5 = r + k \cdot 3$ . With  $k = -2$   $r$  becomes in the range  $0 \dots 2$  and it is  $r = 1$ . Hence,  $\langle -5 \rangle_3 = 1$ . In case of a periodic sequence ( $N = 3$ ) this yields  $\dots = x[-5] = x[-2] = x[1] = x[4] = x[7] = \dots$   
 A circular buffer is implemented in the instruction sets of many DSPs. Assume that there is a buffer of size 1024 bytes, with addresses 0x0000 to 0x03FF in hexadecimals. New 8-bit (byte) samples are read into a buffer where an address counter (pointer) is increased by one each time. When the counter has the value 0x03FF, the next value is  $\langle 0x0400 \rangle_{0x0400} = 0x0000$ . In other words, the oldest sample is replaced by the newest. See Figure 7 for figures of linear and circular buffers.

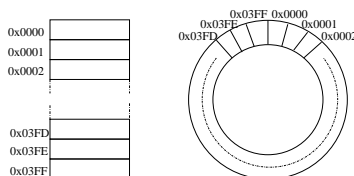


Figure 7: Problem 7: linear and circular buffer with addresses from 0x0000 to 0x03FF.

8. **Problem:** Roots of a polynomial.

- Compute roots of  $H(z) = z^2 + 2z + 2$ .
- Compute roots of  $H(z) = 1 + 16z^{-4}$ .
- Compute long division  $(4z^4 - 8z^3 + 3z^2 - 4z + 6)/(2z - 3)$ .

[L0134] **Solution:** Roots of a polynomial  $p(x)$  can be found from  $p(x) = 0$ .  $N$ th order  $p(x)$  has  $N$  roots, see Problem 8(b).

Roots of a complex  $z = r e^{j(\theta + 2\pi k)}$  are  $\sqrt[N]{z} = |\sqrt[N]{z}| \cdot e^{j(2\pi k/N + \theta/N)}$ , where  $k = 0 \dots N - 1$ . In this course roots of transfer function  $H(z)$  provide information on the behaviour of the filter. The order of the rational polynomial  $H(z) = B(z)/A(z)$  is the maximum of the orders of  $B(z)$  and  $A(z)$ .

- The order of  $H(z)$  is 2. Using the equation for solving the second-order polynomials  $z = (-b \pm \sqrt{b^2 - 4ac})/(2a)$ , the roots are  $z_1 = -1 + j$  and  $z_2 = -1 - j$ . This can be assured by multiplication  $(z - z_1)(z - z_2) = z^2 - (z_1 + z_2)z + z_1 z_2 = z^2 + 2z + 2$ .
- The order of  $H(z)$  is 4, and we have 4 roots. Now, when setting  $H(z) = 1 + 16z^{-4} = 0$ , the equation can be multiplied by  $z^4$  on both sides. Hence,  $z^4 + 16 = 0$  and  $z = \sqrt[4]{-16}$ . Because  $-16 = 2^4 \cdot e^{j(\pi + 2\pi k)}$ , we get four roots using  $\sqrt[4]{z} = |\sqrt[4]{z}| \cdot e^{j(2\pi k/N + \theta/N)}$ . Roots:  $z_k = 2 e^{j(2\pi k/4 + \pi/4)}$ , with  $k = 0 \dots 3$ . Again,  $z_0^4 = (2e^{j\pi/4})^4 = 2^4 e^{j\pi} = 16e^{j\pi} = -16$ , and similarly other  $z_k$  result to  $-16$ . In Figure 8 all four roots are plotted with circles.

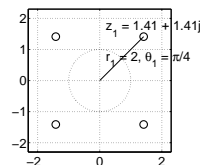


Figure 8: Problem 8(b): four roots of fourth order  $H(z) = 1 + 16z^{-4}$ .

- Division operation can be applied to polynomials just as for normal numbers. Polynomial product and division have a very close connection to the convolution operation. For example, in Matlab there is the same function `conv` for the both operations.  

$$\begin{array}{r} 2z^3 - z^2 - 2 \\ 2z - 3 \overline{) 4z^4 - 8z^3 + 3z^2 - 4z + 6} \\ \underline{-4z^4 + 6z^3} \phantom{- 2} \\ -2z^3 + 3z^2 \phantom{- 4z + 6} \\ \underline{2z^3 - 3z^2} \phantom{- 4z + 6} \\ -4z + 6 \\ \underline{4z - 6} \\ 0 \end{array}$$

9. **Problem:** Examine a complex-valued function ( $z \in \mathbb{C}$ )

$$H(z) = \frac{1 + 0.5z^{-1} + 0.06z^{-2}}{1 - 1.4z^{-1} + 0.48z^{-2}}$$

- Multiply both sides by  $z^2$ .
- Solve  $z^2 + 0.5z + 0.06 = 0$ .
- Solve  $z^2 - 1.4z + 0.48 = 0$ .
- $H(z)$  can be written with five values complex values  $K$ ,  $z_1$ ,  $z_2$ ,  $p_1$ , and  $p_2$

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

What are the five values?

- What are the coefficients of  $H(z)$ . What are the roots of  $H(z)$ ? What is the order of the numerator polynomial of  $H(z)$ ? What is the order of the denominator polynomial of  $H(z)$ ?

[L0139] **Solution:** In this course complex-valued functions are widely used. In case of the transfer function  $H(z)$  both  $z$  and  $H(z)$  are complex-valued. A typical form of a *transfer function* of a FIR filter is

$$H(z) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}$$

and that of an IIR filter is

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Nz^{-N}}$$

- Multiplication  $H(z) \cdot (z^2/z^2)$  does not change the values of  $H(z)$ , but it is more convenient to work with positive exponentials:

$$H(z) = \frac{z^2 + 0.5z + 0.06}{z^2 - 1.4z + 0.48}$$

- Using the formula for second order polynomials  $az^2 + bz + c = 0$

$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

we get easily the roots  $z_1 = -0.3$ ,  $z_2 = -0.2$ . In Matlab you can write `P = [1 0.5 0.06]; roots(P)`.

- Similarly, the roots  $p_1 = 0.8$ ,  $p_2 = 0.6$ .
- Using the notation from (b) and (c),

$$\begin{aligned} H(z) &= K \cdot \frac{(z + 0.3) \cdot (z + 0.2)}{(z - 0.8) \cdot (z - 0.6)} \\ &= K \cdot \frac{z^2 + 0.5z + 0.06}{z^2 - 1.4z + 0.48} \end{aligned}$$

we can scale  $H(z)$  correctly by choosing  $K = 1$ .

- In this case the coefficients were  $\{1, 0.5, 0.06\}$  in numerator polynomial (upper part), and  $\{1, -1.4, 0.48\}$  in denominator polynomial (bottom part).

Roots were computed in (b) and (c). In DSP we call the roots of numerator polynomial as "zeros". The roots of denominator polynomial (bottom part) are "poles". As seen in (d) the same function  $H(z)$  can be expressed either using coefficients or roots (and scaling factor). In the filter analysis the positions of roots give some information on the nature of the filter. More about this in Problem 54.

10. **Problem:** Partial fraction decomposition is used to divide a high-order rational expression into a sum of low-order rational expressions. For example,  $1/(x^2 + 3x + 2) = 1/(x + 1) - 1/(x + 2)$ .

- Decompose  $f(x) = 1/(x^2 + 1)$  into sum of first-order expressions.
- Decompose  $H(z) = (0.4 - 0.2z^{-1})/(1 - 0.1z^{-1} - 0.06z^{-2})$  into sum of first-order expressions.

[L0135] **Solution:** *Partial fraction decomposition* (or *expansion*) is used to divide a high-order rational expression into a sum of low-order rational expressions.

Decomposition is quite trivial if there are not multiple roots neither is the order of numerator polynomial as big or bigger as the order of the denominator polynomial. Decomposition requires taking roots of a polynomial, so it is possible to derive by hands only in some trivial cases, e.g.,  $1/(x^2 + 3x + 2) = 1/(x + 1) - 1/(x + 2)$ . For more complicated cases, see (*Mitra 2Ed Sec. 3.9 / 3Ed Sec. 6.4.3*), or any other math reference. The command in Matlab is `residuez`.

In this course partial fractions are used when finding an explicit form of the impulse response  $h[n]$  from the transfer function  $H(z)$ . In the list of Fourier-transform pairs there are only inverse transforms for the first order expressions. So, if the transfer function is of second-order or higher, it has to be converted to a sum of first-order expressions by partial fraction decomposition (expansion).

Rules of thumb, (1) compute roots of the denominator polynomial, (2) write down the sum of first-order rational polynomials, (3) compute the unknown constants (equation pairs). Note that the decomposition is not unique, but there are several different expressions which lead to the same result.

- Find the roots of the denominator:  $x^2 + 1 = 0 \Rightarrow x_1 = -j$ ,  $x_2 = j$ . Roots can be complex, too! Hence,

$$\begin{aligned} f(x) &= \frac{A}{x - x_1} + \frac{B}{x - x_2} = \frac{A}{x + j} + \frac{B}{x - j} \\ &= \frac{A(x - j) + B(x + j)}{x^2 + jx - jx + 1} = \frac{x(A + B) + j(-A + B)}{x^2 + 1} \end{aligned}$$

$$\Rightarrow \begin{cases} A + B = 0 \\ -A + B = -j \end{cases} \Rightarrow \begin{cases} A = 0.5j \\ B = -0.5j \end{cases}$$

Finally,

$$f(x) = \frac{0.5j}{x + j} - \frac{0.5j}{x - j}$$

- In this course  $z^{-1}$  corresponds a unit delay in time-domain. The numerator polynomial can divided and  $z^{-1}$  terms can be taken to front, and the partial fraction is done only once for  $P(z)$ , whose numerator polynomial is plain 1,

$$\begin{aligned} H(z) &= \frac{0.4 - 0.2z^{-1}}{1 - 0.1z^{-1} - 0.06z^{-2}} \\ &= 0.4 \cdot \frac{1}{\underbrace{1 - 0.1z^{-1} - 0.06z^{-2}}_{P(z)}} - 0.2z^{-1} \cdot \frac{1}{\underbrace{1 - 0.1z^{-1} - 0.06z^{-2}}_{P(z)}} \end{aligned}$$

The denominator of  $P(z)$  is set to zero and multiplied by  $z^2$ :  $z^2 - 0.1z - 0.06 = 0$ . The roots are  $z_1 = 0.3$  and  $z_2 = -0.2$ .

$$\begin{aligned} P(z) &= \frac{A}{1 - 0.3z^{-1}} + \frac{B}{1 + 0.2z^{-1}} \\ &= \frac{A + 0.2Az^{-1} + B - 0.3Bz^{-1}}{1 - 0.1z^{-1} - 0.06z^{-2}} \\ &= \frac{1}{1 - 0.1z^{-1} - 0.06z^{-2}} \quad | \text{ as stated above} \end{aligned}$$

Now we get a pair of equations from comparing numerator polynomials

$$\begin{cases} A + B = 1 \\ 0.2A - 0.3B = 0 \end{cases} \Rightarrow \begin{cases} A = 0.6 \\ B = 0.4 \end{cases}$$

and finally,

$$H(z) = 0.4 \cdot \left( \frac{0.6}{1 - 0.3z^{-1}} + \frac{0.4}{1 + 0.2z^{-1}} \right) - 0.2z^{-1} \cdot \left( \frac{0.6}{1 - 0.3z^{-1}} + \frac{0.4}{1 + 0.2z^{-1}} \right)$$

From this expression of  $H(z)$  we can easily (inverse) transform  $H(z) \rightarrow h[n]$ , see, e.g., Problem 55(f).

11. **Problem:** Sum of geometric series.

- What is sum of series  $S = \sum_{k=0}^{\infty} (0.5)^k$ .
- $S = \sum_{k=10}^{\infty} (-0.6)^{k-2}$ .
- $S = \sum_{k=2}^{\infty} (0.8)^{k-2} \cdot e^{-j\omega k}$ .

[L0136] **Solution:** Sum of *geometric series* is applied in Fourier and  $z$ -transforms. When the ratio  $q$  in geometric series is  $|q| < 1$ , the sum of series converges to  $\sum_{k=0}^{\infty} q^k = 1/(1 - q)$ , and correspondingly  $\sum_{k=0}^N q^k = (1 - q^{N+1})/(1 - q)$ .

Other known series are  $\{1/n\}$  and  $\{1/n^2\}$ . Notice that the sum of the former does not converge, while the latter does.

- Directly from the formula with  $q = 0.5$ ,  $S = 1/(1 - 0.5) = 2$ .
- Open  $\Sigma$  expression if it seems to be difficult.

$$\begin{aligned} S &= \sum_{k=10}^{\infty} (-0.6)^{k-2} = (-0.6)^8 + (-0.6)^9 + (-0.6)^{10} + \dots \\ &= \sum_{k=8}^{\infty} (-0.6)^k \\ &= \sum_{k=0}^{\infty} (-0.6)^k - \sum_{k=0}^7 (-0.6)^k \\ &= 1/(1 + 0.6) - (1 - (-0.6)^8)/(1 + 0.6) = (-0.6)^8/1.6 \approx 0.0105 \end{aligned}$$

- Discrete-time Fourier transform* is defined as

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

$$\begin{aligned} S &= \sum_{k=2}^{\infty} (0.8)^{k-2} \cdot e^{-j\omega k} \quad |k = m + 2 \\ &= \sum_{m=0}^{\infty} (0.8)^m \cdot e^{-j\omega m} \cdot e^{-j2\omega} \\ &= e^{-j2\omega} \cdot \sum_{m=0}^{\infty} (0.8e^{-j\omega})^m \\ &= e^{-j2\omega} \cdot \frac{1}{1 - 0.8e^{-j\omega}} \end{aligned}$$

The term  $e^{-j2\omega}$  can be seen as a time shift (delay in this case) of two units.

12. Problem: Integral transforms, like Fourier transforms, play an important role in signal processing. Compute integrals

- a) X(Omega) = integral from 0 to 4 of e^-j\*Omega\*t dt
b) x[n] = 1/2pi \* integral from 2pi to 2pi X(e^j\*omega) e^j\*omega\*n d\*omega, where X(e^j\*omega) = 2pi\*delta(omega + 0.2pi) - j\*pi\*delta(omega + 0.1pi) + j\*pi\*delta(omega - 0.1pi) + 2pi\*delta(omega - 0.2pi)

[L0137] Solution: A general integral transform is defined by

F(omega) = integral from a to b of f(t)K(omega,t)dt

where K(omega,t) is an integral kernel of the transform, see e.g. (Mitra 2Ed Sec. - / 3Ed Sec. 5.1) or http://mathworld.wolfram.com/IntegralTransform.html. See Problem 17 for discussion on transforms used in DSP.

a) Now x(t) can be considered as a rectangular signal, and its Fourier transform is a sinc-function.

X(Omega) = integral from 0 to 4 of e^-j\*Omega\*t dt = (1/(-j\*Omega))(e^-j\*Omega\*4 - 1) = (1/(-j\*Omega))(e^j\*2\*Omega - e^-j\*2\*Omega) = (1/(-j\*Omega))(-e^-j\*2\*Omega)(2j sin(2\*Omega)) = 4e^-j\*2\*Omega(sin(2\*Omega)/(2\*Omega)) = 4e^-j\*2\*Omega sinc(2\*Omega/pi)

Now we have computed continuous-time Fourier transform (CTFT). It is often written as X(j\*Omega) instead of X(Omega) in DSP.

b) Now we are computing inverse discrete-time Fourier transform (IDTFT), (Mitra 2Ed Eq. 3.7 / 3Ed Eq. 3.16). The spectrum X(e^j\*omega) consists of four peaks (only!) at Omega = {-0.2pi, -0.1pi, 0.1pi, 0.2pi}, which are written with Dirac's delta function delta(omega - omega\_0) and computed as

integral from 2pi to 2pi k \* delta(omega - omega\_0) \* f(omega) d\*omega = k \* f(omega\_0)

when omega = omega\_0, and zero elsewhere. Integral is taken from any period of 2pi, and here we choose -pi ... pi. Euler's formula (e^j\*theta + e^-j\*theta = 2cos(theta) and e^j\*theta - e^-j\*theta = 2j sin(theta)), see Problem 2, is applied and the result is a discrete sequence x[n] of numbers. Note that each peak pair at -omega\_0 and omega\_0 make up a sinusoidal, if the phases are correctly chosen.

x[n] = 1/2pi \* integral from 2pi to 2pi X(e^j\*omega) e^j\*omega\*n d\*omega, where X(e^j\*omega) = 2pi\*delta(omega + 0.2pi) - j\*pi\*delta(omega + 0.1pi) + j\*pi\*delta(omega - 0.1pi) + 2pi\*delta(omega - 0.2pi) ... = 2cos(0.2pi\*n) - sin(0.1pi\*n) in R

13. Problem: Using notation W\_N = e^-j\*2pi/N and matrix

D\_4 = [W\_4^0 W\_4^1 W\_4^2 W\_4^3; W\_4^1 W\_4^2 W\_4^3 W\_4^0; W\_4^2 W\_4^3 W\_4^0 W\_4^1; W\_4^3 W\_4^0 W\_4^1 W\_4^2]

compute X = D\_4\*x, when x = [2 3 5 -1]^T

[L0163] Solution: A common notation W\_N is defined as W\_N = e^-j\*2pi/N. We see that

|W\_N| = |e^-j\*2pi/N| = |cos(2pi/N) - j sin(2pi/N)| = sqrt(cos(2pi/N)^2 + sin(2pi/N)^2) = 1

That is, the points W\_N^k are lying uniformly spaced clockwise on the unit circle. W\_N is periodic with every N, e.g., W\_N^(2N+3) = e^-j\*2pi\*(2N+3)/N = e^-j\*4pi - e^-j\*6pi/N = e^-j\*6pi/N = W\_N^3. Note that W\_N^0 = 1. An example with N = 16 is given in Figure 9.

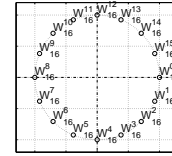


Figure 9: Problem 13: Uniformly spaced {W\_N^k} = {W\_N^0, W\_N^1, ..., W\_N^15} with N = 16.

When N = 4, the angle between each point is 2pi/4 = pi/2 (equivalent to 90 degrees). We can compute values W\_4^0 = 1, W\_4^1 = e^-j\*pi/4 = e^-j\*pi/2 = -j, W\_4^2 = -1, and W\_4^3 = j. Using the periodic properties of W\_N^k the square matrix D\_4 is

D\_4 = [1 1 1 1; 1 -j -1 j; 1 -1 1 -1; 1 j -1 -j]

Size of matrix D\_4 is 4 rows and 4 columns (4 x 4), and that of column vector x is (4 x 1). In the matrix product X = D\_4\*x dimensions must agree: (4 x 4)(4 x 1), and the final size of X is (4 x 1).

X = D\_4\*x = [1 1 1 1; 1 -j -1 j; 1 -1 1 -1; 1 j -1 -j] \* [2; 3; 5; -1] = [1\*2+1\*3+1\*5+1\*(-1); 1\*2-j\*3-1\*5+j\*(-1); 1\*2-1\*3+1\*5-1\*(-1); 1\*2+j\*3-1\*5-j\*(-1)] = [9; -3-4j; 5; -3+4j]

We have computed here discrete Fourier transform (DFT) for a real sequence {2, 3, 5, -1}. The result, here {9, -3-4j, 5, -3+4j}, is generally complex-valued. There are several symmetric properties of DFT that are discussed later.

The matrix D\_4^H (D\_4^H, Hermitian) is transpose of D\_4 with complex-conjugate values:

D\_4^H = [1 1 1 1; 1 j -1 -j; 1 -1 1 -1; 1 -j -1 j]

14. Problem: Consider an analog signal x(t) = pi \* cos(4\*pi\*t). Plot the analog signal, the discrete-time signal sampled with 10 Hz, and the digital signal with accuracy to integer numbers.

[L0280] Solution: Analog signal: both t and x(t) in R. For example, you can measure the outside temperature at any time with continuous scale.

Discrete-time signal (sequence): signal x[n] may get any values at certain time moments, x[n] in R, n in Z. Often explained as a sampled version of analog signal.

Digital signal (sequence): signal x[n] is discrete also with amplitude values, n, x[n] in Z. A pure sinusoidal is x(t) = A \* cos(Omega\*t + theta) and now we have x(t) = pi \* cos(4\*pi\*t). The angular frequency is Omega = 4\*pi rad/s and the frequency f = 2 Hz while Omega = 2\*pi\*f. The period of the sinusoidal signal is T = 1/f = 0.5 second.

The sampling frequency is f\_s = 10 Hz, i.e., samples are taken every T\_s = 0.1 seconds: t -> nT\_s = n/f\_s. This gives a discrete-time sequence

x[n] = pi \* cos(0.4\*pi\*n)

where the normalized angular frequency is omega = 0.4\*pi rad/sample.

Some numeric values are given below in a table, and the plots in Figure 10, where in (a) t runs from 0 to 1.25 seconds, and in (b) and (c) n correspondingly from 0 to 12.

Table with 4 columns: t, n, (a) x(t), (b) x[n], (c) x[n]. Rows show values for t=0, 0 < t < 0.1, 0.1, 0.5 and corresponding n and signal values.

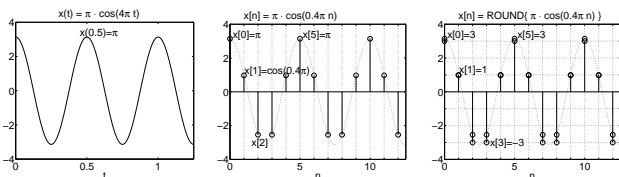


Figure 10: Problem 14: (a) analog signal, (b) discrete-time signal, (c) digital signal.

Discrete-time or digital signal is often called a sequence instead of signal. For instance, in (b) we have a sequence {pi, pi\*cos(0.4pi), pi\*cos(0.8pi), pi\*cos(1.2pi), pi\*cos(1.6pi), pi, ...} and in (c) a sequence {3, 1, -3, -1, 3, ...}. Note that pi is an irrational number which cannot be expressed accurately with finite number of bits.

In practice, A/D (analog to digital) converter discretizes the analog signal into digital signal with a certain accuracy. For instance, in audio recordings (CD quality) the sampling frequency is 44100 Hz (44100 samples each second) and the each data sample is expressed with 16 bits, i.e., having 2^16 = 65536 discrete levels between (appr.) -1 ... +1.

Definitions of signal types (analog, digital, etc.) may vary from book to book.

15. Problem: There is a sketch of the magnitude response of a typical lowpass filter in Figure 11. Sketch some other examples of the magnitude (amplitude) responses of the following filters in the frequency domain: (a) highpass filter, (b) bandpass filter, (c) bandstop filter, (d) notch filter, (e) multiband filter, and (f) comb filter.

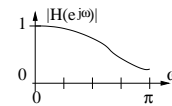


Figure 11: Problem 15: Magnitude (amplitude) response of a typical lowpass filter.

[L0288] Solution: In this course we are dealing with digital LTI filters which are frequency-sensitive filters, in other words, you can amplify or attenuate sinusoidal components of the input signal. Furthermore, LTI constructions are key elements in, e.g., adaptive filters.

Typical LTI filters are lowpass, highpass, bandpass, and bandstop. A word lowpass means that low frequency components of the input signal are left or even amplified (boosted) by the filter whereas high frequencies are attenuated (suppressed). Sketches for these basic filter types are given in Figure 12.

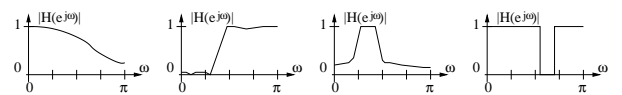


Figure 12: Problem 15: Plots of some magnitude (amplitude) responses of filters. (a) lowpass (LP), (b) highpass (HP), (c) bandpass (BP), and (d) bandstop (BS) filter.

There are a few things to notice here. The magnitude response |H(e^j\*omega)| in R, or often also amplitude response, is the absolute value of the frequency response H(e^j\*omega) in C (generally), which is a function of (normalized angular) frequency omega in R.

Both the x- and y-axis can be either in linear or logarithmic (decibel) scale depending on the application. The x-axis of a magnitude response of a digital LTI filter is normally scaled to the range 0 ... pi which is equivalent to 0 ... f\_r/2 where f\_r is the sampling frequency. The maximum of the magnitude response is often scaled to unity, that is, the maximum in y-axis is either 1 (linear) or 0 decibels (logarithmic). See more spectra examples from Matlab in Problem 16.

An example of an ideal filter (bandstop in this case) is given in Figure 12(d)

H(e^j\*omega) = { 1, 0 <= omega < omega\_1, omega\_2 <= omega < pi; 0, omega\_1 <= omega < omega\_2 }

where omega\_1 and omega\_2 are cut-off frequencies. Ideal filters are important in theoretical derivations but they may give poor results in practice (ringing effect with audio or images). In practice we allow some ripple (deviation) both in the stopband and passband, see Figure 12(a,b,c). Smooth transitions from stopband to passband make the filter order lower which is often desirable.

There are some other LTI filters, too. The spectrum of a *notch filter* is one (or almost) for all frequencies except  $\omega_0$ , which is filtered out, see Figure 13(a). It can be achieved with a simple IIR filter by setting a pole-zero pair at a cut-off frequency, see Problem 54. A *comb filter* is periodic and therefore it is often used to remove harmonic components, see Figure 13(b). For example, distortion of multiples of 50 Hz (AC supply) is common. A *multiband filter*, see Figure 13(c), can be utilized in an equalizer to boost treble (discant) or bass tones. An example of a 10-channel software equalizer can be seen in Figure 14. The user can adjust ten different sliders from which the amplitude response curve is formed. In the right-most figure all high frequencies are attenuated.

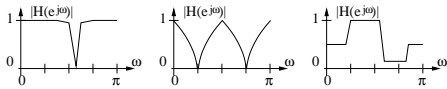


Figure 13: Problem 15: Plots of some magnitude (amplitude) response of filters. (a) notch filter, (b) comb filter, and (c) multiband filter.

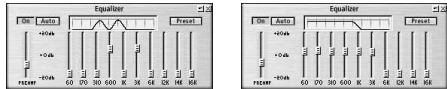


Figure 14: Problem 15: Example of an equalizer in a software (WinAmp).

**Remark.** All magnitude responses in Figure 12 and 13 are sketched by hand. They y-axis is linear and scaled in range  $0 \dots 1$ . In Figure 15 there are figures of a lowpass filter plotted in Matlab. Figure 15(a) is from command `freqz(B, A)`. The top subfigure is the magnitude response, where x-axis linear  $[0 \dots \pi] \times \pi$ , and y-axis logarithmic. The bottom subfigure contains the phase response. Figure 15(b) is from commands `[H, w] = freqz(B, A); plot(w, abs(H))`. Both x-axis and y-axis are linear, x-axis is scaled to  $0 \dots \pi$ . Figure 15(c) is from commands `[H, f] = freqz(B, A, [], f_s); plot(f, 20*log10(abs(H)))`. Now x-axis is linear but y-axis is logarithmic, and x-axis is scaled to  $0 \dots 22050$ , that is, half of the sampling frequency when  $f_s = 44100$  Hz (CD quality). Note that magnitude response curves look like the same in (a) top subfigure and (c), while in (b) the same curve is in linear scale. The ripple of 1 decibel (dB) in passband corresponds oscillation between  $0.891 \dots 1$  in linear y-axis. Similarly, the stopband attenuation of 40 dB corresponds 0.01.

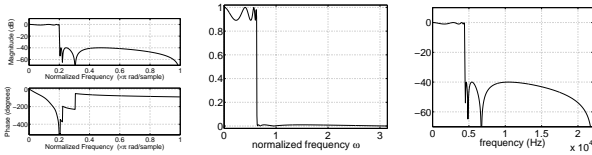


Figure 15: Problem 15: Magnitude responses of a LP filter.

The *spectrogram* is a *short-time Fourier transform (STFT)* method, (Mitra 2Ed Sec. 11.3.1 / 3Ed Sec. 15.3.1). It is nothing but computing spectra in small time windows. In a spectrogram, see Figure 17(c), time is in x-axis and frequency in y-axis, and the gray level in z-axis shows how strong a certain frequency (sinusoidal) is at a certain time moment. It can be seen that the component at 1 kHz is present only in the end.

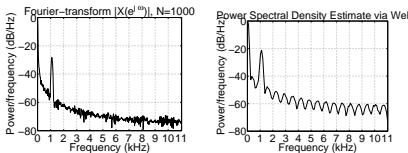


Figure 17: Problem 16: (a)  $|X|$ , Fourier transform with full length, (b)  $|X|$ , spectrum estimation (Welch) using DFT-256, both plotting a peak at about 1100 Hz, (c) spectrogram of the signal, time in x-axis and frequency in y-axis. See the high-frequency (1100 Hz) component with white color starting at 0.03 seconds.

**Remark.** In Fourier analysis we assume that the signal examined is *stationary*. In Figure 16(c) we see a signal which is not stationary in this time window. The standard way is to examine the signal in small consecutive (or/and overlapping) time windows or frames, just like in the spectrogram above. The window size and other parameters are application-dependent. In speech analysis it is around 10-40 ms.

When taking Fourier transform of the signal in Figure 16(c) we get the spectrum in Figure 17(a). However, if we see first the spectrum, we cannot exactly say, how the signal looks like. For example, even if the spectrum estimation in Figure 18(a) is very similar to that in Figure 17(a), the original signal is very different, see Figure 18(b).

Note that the conversion  $x[n] \leftrightarrow X(e^{j\omega})$  via Fourier transform is unique (Fourier is bijection function), but we normally examine only absolute value  $|X(e^{j\omega})|$ , which can be achieved by several (infinite number)  $x[n]$ .

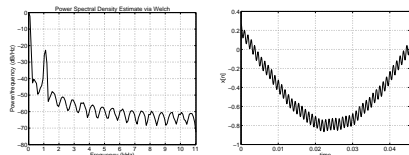


Figure 18: Problem 16: (a)  $|X|$ , spectrum estimation (Welch) using DFT-256, very similar to a spectrum above, (b) the original signal for (a), the occurrence of the high-frequency signal component is different from above.

16. **Problem:** Explain the spectra in Figure 16(a) and (b). What does a spectrum of a signal in Figure 16(c) look like? How about its spectrogram?

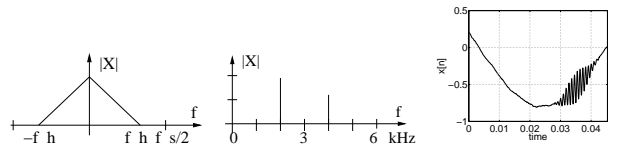


Figure 16: Problem 16: (a) two-sided triangular spectrum, frequency in x-axis, (b) one-sided line spectrum, frequency in x-axis, (c) signal waveform, time in x-axis.

[L0290] **Solution:** In Problem 15 we saw several types of filters depicted in frequency-domain. Now we are considering signals and their spectra.

The triangular spectrum in Figure 16(a) typically depicts a spectrum  $X$  of any kind of signal, or, we are not interested how the signal looks like in time-domain. However, one should find out from the context, if we are talking about spectrum  $X(j\Omega)$  of an analog signal after continuous-time Fourier transform (CTFT),  $X(e^{j\omega})$  of a discrete sequence after discrete-time Fourier transform (DTFT), or  $X[k]$  of a discrete sequence after discrete Fourier transform (DFT).  $X[k]$  can be thought as a sampled version of  $X(e^{j\omega})$ .

In Figure 16(a) we have a *two-sided spectrum* which is band limited between  $-f_h$  and  $f_h$ . It is symmetric around y-axis, that is, the corresponding signal in time-domain is real-valued (like all signals found in nature), see Problem 41. The highest frequency component is  $f_h$  which is less than half of the sampling frequency  $f_s/2$ .

The *line spectrum* in Figure 16(b) contains two peaks at 2 kHz and 4 kHz. If the signal is real-valued, as it is often by default, the *one-sided spectrum* is enough because its left side is a mirror image of the right side. One could draw the left side of the spectrum by adding 2 units high peak at -2 kHz and 1 unit high peak at -4 kHz. The signal  $x[n]$  or  $x(t)$  can be quite easily plotted from the spectrum, e.g.,  $x(t) = 2 \cos(2\pi \cdot 2000 \cdot t) + \cos(2\pi \cdot 4000 \cdot t)$ .

In Figure 16(c) there is a signal which contains one slowly varying signal, a quickly changing part at 0.03...0.04 seconds, and small amount of other noise. We can imagine that we will have a line spectrum with two peaks: one strong peak at very low frequency and the other, smaller peak at a frequency that can be estimated from the figure. We can read that there are 11 oscillations in 0.03...0.04 seconds, that is, 1100 oscillations a second (1100 Hz).

Figure 16(c) is plotted from Matlab. In this case we have a discrete sequence  $x[n]$  of 1000 numbers in a computer, from which we can compute another discrete sequence  $X[k]$  by DFT (`fft`). Using all 1000 samples we get the absolute value of discrete Fourier transform in Figure 17(a). Two peaks close to 0 and 1100 Hz can be found.

There is a large number of *spectrum estimation* methods. Figure 17(b) shows the result of Welch approximation using only 256 samples (`pwelch`). It also detects a peak at 1 kHz. See discussion on spectrum spreading in DFT in Remark of Problem 37.

In both figures we clearly see that there is energy at zero frequency (or low frequency) and at about 1 kHz which reflects the high-frequency component starting at 0.03 seconds. However, the time information, when the high-frequency component occurs, cannot be seen from the spectrum. Therefore we can analyze the signal in time-frequency-domain.

17. **Problem:** Categorize Fourier-transforms for aperiodic/periodic analog/digital signals.

[L0291] **Solution:** Integral transforms are important and essential tools in signal processing. With Fourier-transforms you can change between time-domain and frequency-domain, see Problem 18. In a high level several Fourier transforms can be categorized into four classes according to signal type (analog/digital) and periodicity (periodic/aperiodic), see Table 5 from (Huttunen: *Signaalinkäsittelyn menetelmät*, p. 33) and (Smith: *DSPguide*, Ch. 8, p. 145). In addition, more general z-transform for digital, and Laplace transform (with variable s) for analog signals are applied. There exist lots of other transforms, too, e.g., cosine and Hadamard transforms used in image processing.

All transforms have exact mathematical definitions from which transforms can be computed. However, there are several parametrized formula tables which offer ready-calculated transform pairs and computation rules, see Page 166. Examples from Problem 36 on.

Next we introduce the Fourier transforms in Table 5. Small symbol, e.g.  $x$ , refers to a time-domain signal which is then analyzed (decomposed) to a frequency-domain signal (spectrum)  $X$ , written with a corresponding capital letter. The inverse operation, synthesis, forms an unique signal  $x$  from spectral components  $X$ .

	continuous-time signal	discrete-time signal
aperiodic signal	Fourier transform (cont. $\rightarrow$ cont.)	Discrete-time Fourier transform (discrete $\rightarrow$ cont.)
periodic signal	Fourier series (cont. $\rightarrow$ discrete)	Discrete Fourier transform (discrete $\rightarrow$ discrete)

Table 5: Problem 17: Classification of Fourier transforms.

**Fourier transform.** Continuous-time and aperiodic signal  $x(t)$  is transformed. The result  $X(j\Omega)$  is continuous-frequency and aperiodic. Mitra's book uses the term *continuous-time Fourier transform (CTFT)*, (Mitra 2Ed p. - / 3Ed p. 118). The analysis and synthesis equations are  $X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt$ , and  $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega$ . Note that  $t$  and  $\Omega \in \mathbb{R}$ .

**Discrete-time Fourier transform.** Discrete-time and aperiodic signal  $x[n]$  is transformed. The result  $X(e^{j\omega})$  is continuous-frequency and periodic with  $2\pi$ . See more about *discrete-time Fourier transform (DTFT)* in (Mitra 2Ed p. 117 / 3Ed p. 122). The analysis and synthesis equations are  $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$ , and  $x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n} d\omega$ . Note that  $n \in \mathbb{Z}$  and  $\Omega \in \mathbb{R}$ .

**Fourier series.** Continuous-time and periodic signal  $x(t)$  is transformed. The result are the Fourier series coefficients  $a_k$  with a fundamental frequency  $\Omega_0$ . Coefficients are therefore discrete-frequency and aperiodic. The analysis and synthesis equations are  $a_k = \frac{1}{T} \int_p x(t)e^{-jk\Omega_0 t} dt$ , and  $x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\Omega_0 t}$ . Note that  $t \in \mathbb{R}$  and  $k \in \mathbb{Z}$ .

**Discrete Fourier transform.** Discrete-time and periodic signal  $x[n]$  is transformed. The result  $X[k]$  is discrete-frequency and periodic. This is what billions of chips do right now all over the world! When Matlab plots you a nice, smooth frequency response (DTFT) curve of a digital filter, it actually computes DFT so densely in frequency-domain that the result seems to be continuous. (Just like when Matlab plots you an audio signal waveform, it is actually only a series of points, sequence, connected with lines!) See more about *discrete Fourier transform (DFT)* in (Mitra 2Ed p. 131 / 3Ed p. 234). In DFT equations we use a shorthand  $W_N = e^{-j2\pi/N}$ , and the analysis and synthesis equations are  $X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}$ , where  $0 \leq k \leq N-1$ , and  $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-kn}$ , where  $0 \leq n \leq N-1$ . Note that  $n$  and  $k \in \mathbb{Z}$ .

18. **Problem:** Consider a signal  $x[n]$  shown in Figure 19. Assume that there exists a lowpass filter and a highpass filter with a cut-off at 1 kHz. Sketch the outputs of the both filters.

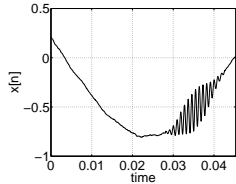


Figure 19: Problem 18: Input signal  $x[n]$ .

[L0289] **Solution:** First, some sketches in frequency domain are given as results to this problem. Second, figures from Matlab with some real (but here ungiven) signals and filter constructions are plotted. Third, the overall picture of the time-frequency-domain analysis is given in the end of this problem (Remark).

The signal in Figure 19 was already analyzed in Problem 16. It contains slowly changing part (low frequency) and quickly changing part (high frequency, approx. 1100 Hz, only present at 0.03...0.04 s). Therefore we can sketch a spectrum  $X(e^{j\omega})$  of the signal in left in Figure 20.

Filtering in frequency-domain is purely a product of the input spectrum  $X(e^{j\omega})$  and the filter  $H(e^{j\omega})$ . Now the cut-off is at 1 kHz and the magnitude response is sketched in middle in Figure 20.

The output spectrum  $Y(e^{j\omega})$  of a lowpass filter (LP) is sketched in right in Figure 20. There is a peak at very low frequency meaning that only the slowly changing part of the signal is left. Filtering with a highpass filter (HP) gives a peak at 1.1 kHz, which in time-domain corresponds quickly changing part of signal at 0.03 seconds.

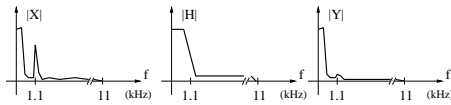


Figure 20: Problem 18: Sketch of lowpass filtering. From left to right: input spectrum  $X(e^{j\omega})$ , transfer function  $H(e^{j\omega})$  of LP filter, output spectrum  $Y(e^{j\omega})$ .

In reality, we should, of course, know an exact form of  $H(e^{j\omega})$  to filter the signal. Next, the problem is illustrated with some real sequence values and Matlab figures.

The magnitude response of the LP filter using the cut-off frequency 1 kHz is given in frequency-domain in middle-bottom in Figure 21, and that of HP filter in middle-bottom in Figure 22.

Filtering in frequency-domain is a product of the input spectrum  $X(e^{j\omega})$  and the filter  $H(e^{j\omega})$ . Hence, the output spectrum of LP filter is given in right-bottom in Figure 21

from which the sequence  $y[n]$  is plotted in right-top. The output spectrum of HP filter is in Figure 22, respectively.

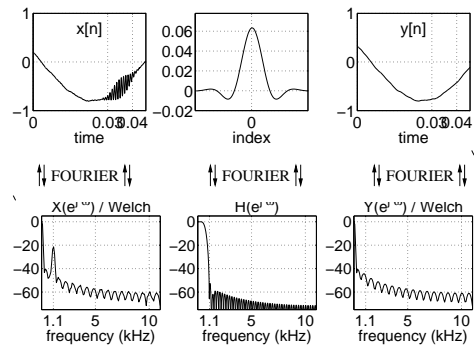


Figure 21: Problem 18: Lowpass filtering. Top row: input signal  $x[n]$ , impulse response  $h[n]$  of LP filter, output signal  $y[n]$ . Bottom row: input spectrum  $X(e^{j\omega})$ , transfer function  $H(e^{j\omega})$  of LP filter, output spectrum  $Y(e^{j\omega})$ .

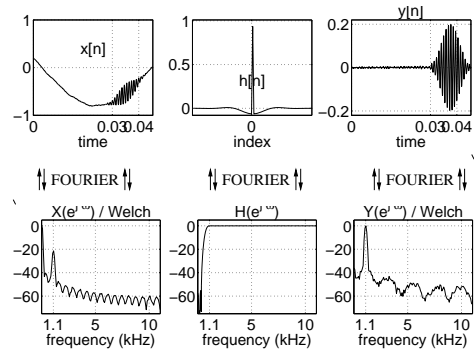


Figure 22: Problem 18: Highpass filtering. Top row: input signal  $x[n]$ , impulse response  $h[n]$  of HP filter, output signal  $y[n]$ . Bottom row: input spectrum  $X(e^{j\omega})$ , transfer function  $H(e^{j\omega})$  of HP filter, output spectrum  $Y(e^{j\omega})$ .

**Remark.** This course deals with SISO systems, that is, a system with a single input and a single output. The systems are mainly linear and time-invariant (LTI) digital ones. The analysis of signals and systems requires Fourier transform (or equivalent) to change from time-domain to frequency-domain and back.

The big picture of the analysis can be seen in Figure 23. There is an input  $x[n]$ , a system  $h[n]$  and an output  $y[n]$  in time-domain, or  $X(z)$ ,  $H(z)$ , and  $Y(z)$  in frequency-domain, respectively. The system is called a filter.

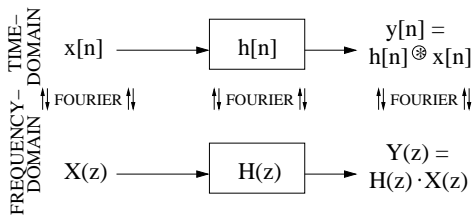


Figure 23: Problem 18: Filtering in time-domain and frequency-domain. Time-domain: Input signal  $x[n]$ , impulse response  $h[n]$ , output  $y[n] = h[n] \otimes x[n]$  by convolution. Frequency-domain: Input spectrum  $X(z)$ , transfer function  $H(z)$ , output  $Y(z) = H(z) \cdot X(z)$  by multiplication. Fourier or  $z$ -transform (etc.) are needed in the analysis.

Signal is a function of independent variables such as time (speech) or position (images) (Mitra 2Ed Sec. 1.0, p. 1 / 3Ed Sec. 1.0, p. 1). It nature signals carry information, e.g., a speech signal is a longitudinal waveform transporting energy from speaker's speech organ to listener's hearing. Signals have certain properties: they can be analog ( $x(t)$ ) or digital ( $x[n]$ ), periodic or aperiodic. In this course digital signals (sequences) are plain sequences of numbers which can be processed and saved in a computer. There exist analog/digital (AD) and digital/analog (DA) hardware to make conversion  $x(t) \leftrightarrow x[n]$ .

System takes an input, processes it and returns an output. It has also some properties. In case of digital systems, realized either software or hardware, system can be, e.g., linear, time-invariant (or shift-invariant), causal or stable. See more in Problem 25.

In *time-domain* we speak about input and output signals  $x[n]$  and  $y[n]$ . All symbol letters are typically written with lower case. A LTI system is identified uniquely with its *impulse response*  $h[n]$ , and the output  $y[n]$  is computed as a convolution of sequences  $h[n]$  and  $x[n]$ .

In *frequency-domain* (or *transform-domain*) signals are transformed and the corresponding symbol letters are capitalized.  $X(e^{j\omega})$  and  $Y(e^{j\omega})$  are input and output spectra, whereas  $H(e^{j\omega})$  is the *frequency response* of the system ( $H(z)$  is called *transfer function*). Here the output  $Y(e^{j\omega})$  is computed as a product of spectrum  $X(e^{j\omega})$  and frequency response  $H(e^{j\omega})$ .

Altering between time-domain and frequency-domain is achieved by applying Fourier transform and inverse Fourier transform (or other integral transforms, like  $z$ -transform and Laplace transform). For example, if we want to compute the convolution in frequency-domain, we first apply discrete Fourier transform (DFT) to  $x[n]$  and  $h[n]$  in order to get  $X[k]$  and  $H[k]$ , then we compute  $Y[k] = H[k] \cdot X[k]$ , and finally apply inverse discrete Fourier transform (IDFT) to get  $y[n]$ . More about transforms in Problem 17.

19. **Problem:** Sketch the following sequences around the origin

- a)  $x_1[n] = \sin(0.1\pi n)$
- b)  $x_2[n] = \sin(2\pi n)$
- c)  $x_3[n] = \delta[n-1] + \delta[n] + 2\delta[n+1]$
- d)  $x_4[n] = \delta[-1] + \delta[0] + 2\delta[1]$
- e)  $x_5[n] = \mu[n] - \mu[n-4]$
- f)  $x_6[n] = x_3[-n+1]$

[L0297] **Solution:** The *unit impulse function*  $\delta[n]$  and the *unit step function*  $\mu[n]$  (or  $u[n]$ ) are defined

$$\delta[n] = \begin{cases} 1, & \text{when } n = 0 \\ 0, & \text{when } n \neq 0 \end{cases} \quad \mu[n] = \begin{cases} 1, & \text{when } n \geq 0 \\ 0, & \text{when } n < 0 \end{cases}$$

A *sequence* or a *discrete-time signal*  $x[n]$  can be shown with individual values  $x[k]$  and impulse function as

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k]$$

For example, if  $x_e[0] = 2$ ,  $x_e[3] = -4$ , and  $x_e[n] = 0$  elsewhere, we get

$$x_e[n] = x_e[0] \cdot \delta[n-0] + x_e[3] \cdot \delta[n-3] = 2\delta[n] - 4\delta[n-3]$$

The discrete-time signal is purely a sequence of numbers. There are equivalent ways to write a sequence, e.g. in (c),  $x_3[n] = 2\delta[n+1] + \delta[n] + \delta[n-1]$ , or  $x_3[n] = \{2, \underline{1}, 1\}$ , where non-zero values are listed and the underlined position is at  $n = 0$ . Non-zero values can be also written explicitly, e.g.,  $x_3[-1] = 2$ ,  $x_3[0] = 1$ ,  $x_3[1] = 1$ , and  $x_3[k] = 0$  elsewhere.

There are different ways to draw discrete-time signals. Here we use "pins" or "stems", which emphasizes that the sequence is discrete-time. See examples in Figure 24.

While  $\sin(\cdot)$ ,  $\delta[n]$ , and  $\mu[n]$  are functions of  $n$ , we can compute these functions at certain points of  $n$  by using tables, like in (a)

$n$	$\sin(0.1\pi n)$	$x_1[n]$
-2	$\sin(0.1\pi(-2))$	$\approx -0.588$
-1	$\sin(0.1\pi(-1))$	$\approx -0.309$
0	$\sin(0.1\pi(-0))$	0
1	$\sin(0.1\pi(1))$	$\approx 0.309$
2	$\sin(0.1\pi(2))$	$\approx 0.588$

Note that in (b) the argument for the sine function is always  $2\pi$ -multiple and therefore  $\dots = \sin(0) = \sin(2\pi) = \dots = 0$ . Shifted sequences can be written in a table, like in (c),

$n$	$\delta[n-1]$	$\delta[n]$	$2\delta[n+1]$	$x_3[n]$
-2	$\delta[-2-1] = 0$	$\delta[-2] = 0$	$2\delta[-2+1] = 0$	$0+0+0 = 0$
-1	$\delta[-1-1] = 0$	$\delta[-1] = 0$	$2\delta[-1+1] = 2$	$0+0+2 = 2$
0	$\delta[0-1] = 0$	$\delta[0] = 1$	$2\delta[0+1] = 0$	$0+1+0 = 1$
1	$\delta[1-1] = 1$	$\delta[1] = 0$	$2\delta[1+1] = 0$	$1+0+0 = 1$
2	$\delta[2-1] = 0$	$\delta[2] = 0$	$2\delta[2+1] = 0$	$0+0+0 = 0$

In (d) there are only constants  $\delta[-1] = \delta[1] = 0$  and  $\delta[0] = 1$  from the definition, hence  $x_4[n] = 1$  for all  $n$ . Sequences can be summed together, like in (e),  $x_5[n] =$



$\{\underline{1}, 1, 1, 1, 1, \dots\} - \{0, 0, 0, 0, 1, 1, 1, \dots\} = \{\underline{1}, 1, 1, 1\}$ . If you have troubles to read time-reversed sequences, e.g., in (f),  $x_6[n] = x_3[-n + 1]$  in (f), you can simply put values into a table:

$n$	$x_3[n]$	$x_6[n] = x_3[-n + 1]$
-1	$x_3[-1] = 2$	$x_6[-1] = x_3[-(-1) + 1] = x_3[2] = 0$
0	$x_3[0] = 1$	$x_6[0] = x_3[-(0) + 1] = x_3[1] = 1$
1	$x_3[1] = 1$	$x_6[1] = x_3[-(1) + 1] = x_3[0] = 1$
2	$x_3[2] = 0$	$x_6[2] = x_3[-(2) + 1] = x_3[-1] = 2$

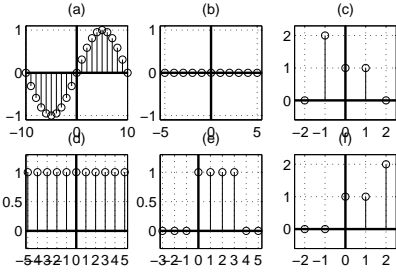


Figure 24: Sequences of Problem 19. Top row: (a)-(c), bottom: (d)-(f).

**Remark.** For real-world physical signals, e.g., speech signal, there exists the *sampling frequency*  $f_T$  (Hz). The inverse  $1/f_T$  gives us the time in seconds between each sample.

Assume that there is an audio file `kiisseli.wav`, which contains a Finnish word “kiisseli” pronounced by female, see Figure 25. The file can be loaded into Matlab. It can be seen that the length of the digital sequence  $x[n]$  is 29001, each number is coded with 8 bits ( $2^8 = 256$  quantization levels between -1 and +1), and the sampling frequency is  $f_T = 22050$  Hz (sampling period  $T = 1/f_T \approx 45.4\mu s$ ). The sequence  $x[n]$  is in the memory as “double” type meaning double precision floating point, 8 bytes (64 bits) per sample. The whole sequence requires therefore 232008 bytes linear memory. For instance, the sample  $x(1)$  is -0.094 and  $x(29001)$  0.023.

```
Matlab commands: [x, fs, nbits] = wavread('kiisseli.wav');
length(x), class(x), x(1), x(end).
```

We can now write our signal using  $\delta$  notations

$$x[n] = -0.094\delta[n] + \dots + 0.023\delta[n - 29000]$$

The length of  $x[n]$  in seconds is  $L_s = 29001/f_T = 1.315$  s, which can be derived from the fact that in one second there are  $f_T$  samples.

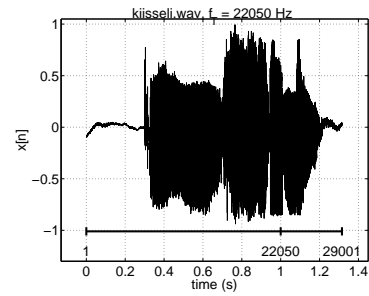


Figure 25: Remark of Problem 19. The waveform of a speech signal in the file `kiisseli.wav`. While sampling frequency is 22050 Hz, there are 22050 samples in each second. Therefore 29001 samples correspond  $\approx 1.3$  seconds.

20. **Problem:** Which of the following signals are periodic? Determine the length of the fundamental period for periodic signals.

- a)  $x(t) = 3 \cos(\frac{8\pi}{31}t)$
- b)  $x[n] = 3 \cos(\frac{8\pi}{31}n)$
- c)  $x(t) = \cos(\frac{\pi}{8}t^2)$
- d)  $x[n] = 2 \cos(\frac{\pi}{6}n - \pi/8) + \sin(\frac{\pi}{16}n)$
- e)  $x[n] = \{\dots, 2, 0, 1, 2, 0, 1, 2, 0, 1, \dots\}$
- f)  $x[n] = \sum_{k=-\infty}^{+\infty} \delta[n - 4k] + \delta[n - 4k - 1]$

[L0162] **Solution:** A *continuous-time signal*  $x(t)$  is *periodic* if there exists period  $T \in \mathbb{R}$ , for which  $x(t) = x(t + T)$ ,  $\forall t$ . The *fundamental period* is the smallest  $T_0 > 0$ .

A *discrete-time signal* (sequence)  $x[n]$  is *periodic*, if there exists period  $N \in \mathbb{Z}$ , for which  $x[n] = x[n + N]$ ,  $\forall n \in \mathbb{Z}$ . The *fundamental period* is the smallest  $N_0 > 0$ .

Examination of periodicity can be restricted to a certain time range.

One way to find the period is to replace  $t$  by  $t + T$  ( $n$  by  $n + N$ ) and try if the equation  $x(t) = x(t + T)$  holds.

Another way to find the period of a sinusoidal component (sine, cosine) is to express the function in form of  $x(t) = \sin(2\pi \cdot f \cdot t)$  where  $f$  is frequency ( $\Omega = 2\pi f$  is angular frequency). Then  $T = 1/f$ .

Typically signals can be synthesized as a sum of sinusoids. Sines, cosines, and exponential functions are all  $2\pi$ -periodic, e.g.,  $A \sin(\Omega t + \theta) \equiv A \sin(\Omega t + \theta + 2\pi k)$ . Note that the amplitude  $A$  or phase shift  $\theta$  does not have effect on periodicity.

If there is a sum of cosines, like in (d), one has to find period  $T_0$  ( $N_0$ ), to which all periods of individual cosines are multiples, see Figure 27. Correspondingly, in the frequency domain one has to find a fundamental frequency  $f_0$ , with which all individual frequencies can be represented as multiples.

Signals (a)..(f) are plotted for a small range in Figure 28. The non-periodicity of the signal is often easy to see from the signal plot, for instance, the signal (c) is clearly not periodic.

a) Periodic. Substitute  $t$  by  $(t + T)$  and test, if the additional part is multiple of  $2\pi$ .

$$\begin{aligned} x(t) &= 3 \cos\left(\frac{8\pi}{31}t\right) = 3 \cos\left(\frac{8\pi}{31}(t + T)\right) = 3 \cos\left(\frac{8\pi}{31}t + \frac{8\pi}{31}T\right) \\ &= 3 \cos\left(\frac{8\pi}{31}t + 2\pi \underbrace{\left(\frac{4}{31}T\right)}_{=2\pi k}\right) \\ 2\pi k &= 2\pi \frac{4}{31}T \\ T &= (31/4)k \end{aligned}$$

When  $T = (31/4)k$ , then  $2\pi$ -multiple is added to the argument of the original cosine, and  $x(t) = x(t + T)$  holds. The fundamental period is the shortest period  $T_0 = 31/4$ , that is,  $k = 1$ .

b) Periodic. Substitute  $n$  by  $(n + N)$  is a similar way as in (a)

$$\begin{aligned} x[n] &= 3 \cos\left(\frac{8\pi}{31}n\right) = 3 \cos\left(\frac{8\pi}{31}(n + N)\right) = 3 \cos\left(\frac{8\pi}{31}n + \frac{8\pi}{31}N\right) \\ &= 3 \cos\left(\frac{8\pi}{31}n + 2\pi \underbrace{\left(\frac{4}{31}N\right)}_{=2\pi k}\right) \\ 2\pi k &= 2\pi \frac{4}{31}N \\ N &= (31/4)k \end{aligned}$$

Again  $N = (31/4) \cdot k$ , where the period  $N$  has to be integer. The smallest possible  $k = 4$  gives the length of the fundamental period  $N_0 = 31$ .

Notice also the difference of the results in (a) and (b), where  $x(t) = x(t + (31/4))$ , but  $x[n] = x[n + 31]$ . The signals are plotted in the same axis in Figure 26.

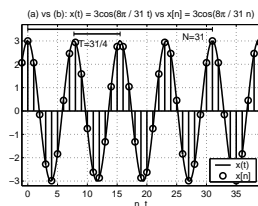


Figure 26: A visualization of the difference of fundamental periods of similar looking analog and discrete-time signals in Problem 20(a) and (b).  $T_0 = 31/4$  but  $N_0 = 31$ .

c) Not periodic (aperiodic), the latter term depends on  $t$ . The result can be also seen

in Figure 28(c).

$$\begin{aligned} x(t) &= \cos\left(\frac{\pi}{8}t^2\right) = \cos\left(\frac{\pi}{8}(t + T)^2\right) = \cos\left(\frac{\pi}{8}t^2 + \frac{\pi}{8}(2tT + T^2)\right) \\ &= \cos\left(\frac{\pi}{8}t^2 + 2\pi \underbrace{\left(\frac{tT}{8} + \frac{T^2}{16}\right)}_{=2\pi k}\right) \\ 2\pi k &= 2\pi \left(\frac{tT}{8} + \frac{T^2}{16}\right) \\ T &= -t \pm \sqrt{t^2 - 16k} \quad \text{no constant } T \text{ exists} \end{aligned}$$

d) Periodic,  $N_0 = 48$ . Compute first individual periods of each sinusoidal. The fundamental period is the *least common multiple* (LCM) of periods  $N_1 = 12$ ,  $N_2 = 16 \Rightarrow N_0 = 4N_1 = 3N_2 = 48$ . See Figure 27.

$$x[n] = 2 \cos\left(\frac{\pi}{6}n - \pi/8\right) + \sin\left(\frac{\pi}{16}n\right) = 2 \cos\left(2\pi \frac{1}{12}n - \pi/8\right) + \sin\left(2\pi \frac{1}{16}n\right)$$

Correspondingly, the fundamental angular frequency is the *greatest common divisor* (GCD) of individual frequencies ( $\omega = 2\pi/N$ ):  $\omega_1 = \pi/6 = 4\pi/24$ ,  $\omega_2 = \pi/8 = 3\pi/24 \Rightarrow \omega_0 = \pi/24 \Rightarrow \omega_1 = 4\omega_0$ ,  $\omega_2 = 3\omega_0$ . More about computing LCM and GCD can be found, e.g. “Beta, Mathematics Handbook for Science and Engineering”. There are Matlab commands `lcm` and `gcd`, too.

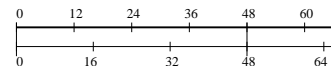


Figure 27: Problem 20(d): least common multiple (LCM) for  $N_1 = 12$  and  $N_2 = 16$  is  $N_0 = 48$ .

- e) (Assume that) the period is  $N_0 = 3$ , i.e.  $x[0] = x[\pm 3k] = 2$ ,  $x[1] = x[\pm 3k + 1] = 0$ ,  $x[2] = x[\pm 3k + 2] = 1$ , where the integer  $k > 0$ .
- f)  $N_0 = 4$ . “Open” the sequence if you do not see it directly:

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{+\infty} \delta[n - 4k] + \delta[n - 4k - 1] \\ &= \dots + \underbrace{\delta[n + 4] + \delta[n + 4 - 1]}_{k=-1} + \underbrace{\delta[n] + \delta[n - 1]}_{k=0} + \underbrace{\delta[n - 4] + \delta[n - 4 - 1]}_{k=1} + \dots \\ &= \{\dots, 1, 1, 0, 0, \underline{1}, 1, 0, 0, 1, 1, 0, 0, \dots\} \end{aligned}$$

**Remark.** *Real-life signals* are seldom periodic in strict sense; almost periodic signals are sometimes called *quasi-periodic*. See Figure 29, where the signal part of a vowel /i/ in 0.02 second range is not exactly periodic  $x(t) = x(t + 0.02s)$  but almost.

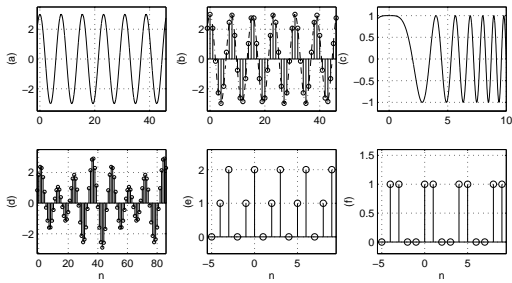


Figure 28: Signals and sequences in Problem 20, (a)..(c) in top row, (d)..(f) in bottom row. It can be seen that (at least) (c) is not periodic in the scene shown.

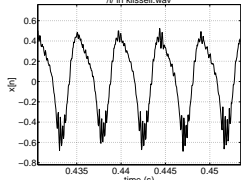


Figure 29: Problem 20: example of a quasi-periodic signal.

21. **Problem:** Compute “a two-point moving average” temperature from daily temperatures in DSPVillage in early July:

July	1st	2nd	3rd	4th	5th	6th	7th
°C	12	16	15	22	20	24	23

[L0279] **Solution:** A two-point moving average takes two adjacent samples, sum them together, divide by two, and goes on to the following point. The purpose of averaging is to suppress quick changes, possibly noise, in the signal.

In order to compute the average temperature on 3rd July, we take temperatures of 3rd and 2nd, and compute mean of them as shown below. First and last values of the result are not comparable because the other value is assumed to be zero. This problem is known as a *border effect* and it has to be dealt with in appropriate way.

	1st	2nd	3rd	4th	5th	6th	7th	
$x[n]$	12	16	15	22	20	24	23	
$x[n-1]$		12	16	15	22	20	24	23
MA-2	6	14	15.5	18.5	21	22	23.5	11.5

22. **Problem:** Compute an average temperature by adding half of temperature today and half of average temperature yesterday. Use the same daily temperature data as in Problem 21:

July	1st	2nd	3rd	4th	5th	6th	7th
°C	12	16	15	22	20	24	23

[L0287] **Solution:** Take a new sample and the last computed output value, sum them together, divide by two, and go on to the following point. The purpose of averaging is to suppress quick changes, possibly noise, in the signal.

In order to compute the average temperature on 3rd July, we take temperature of 3rd and the average of 2nd, and compute mean of them as shown below. Borders have to be dealt with in appropriate way.

	1st	2nd	3rd	4th	5th	6th	7th	
$x[n]$	12	16	15	22	20	24	23	
$y[n-1]$		6	11	13	17.5	18.8	21.4	22.2
$y[n]$	6	11	13	17.5	18.8	21.4	22.2	11.1

Now the averaging process can be thought as a filtering operation:

$$y[n] = \frac{x[n] + y[n-1]}{2} = \{6, 11, 13, 17.5, 18.8, 21.4, 22.2, 11.1, 5.6, \dots\}$$

This corresponds a LTI system with the infinite-length impulse response

$$h[n] = 0.5 \cdot 0.5^n \mu[n] = \{0.5, 0.25, 0.125, \dots\}$$

and the frequency response (see, e.g., Problem 42 and 55)

$$H(e^{j\omega}) = \frac{0.5}{1 - 0.5e^{-j\omega}}$$

The corresponding flow (block) diagram is plotted in Figure 31. In this case there exists a feedback loop – the filter type is now IIR (infinite impulse response, IIR), whereas Problem 21 shows an example of the other type FIR.

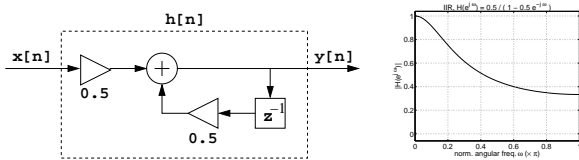


Figure 31: Problem 21: (a) Flow (block) diagram of the filter. Input  $x[n]$  is filtered by the system  $h[n]$ , which is dashed and includes everything between input and output, and the output is  $y[n]$ . The system includes a feedback loop which is equivalent that the system is IIR. (b) Magnitude response  $|H(e^{j\omega})|$  of an IIR filter. It can be seen that it is a lowpass filter.

Formally, the input sequence is  $x[n] = \{12, 16, 15, 22, 20, 24, 23\}$  or

$$x[n] = 12\delta[n] + 16\delta[n-1] + 15\delta[n-2] + 22\delta[n-3] + 20\delta[n-4] + 24\delta[n-5] + 23\delta[n-6]$$

The averaging process can be thought as a filtering operation:

$$y[n] = \frac{x[n] + x[n-1]}{2} = \{6, 14, 15.5, 18.5, 21, 22, 23.5, 11.5\}$$

This corresponds a LTI system with the finite-length (now  $N = 2$ ) impulse response

$$h[n] = \frac{\delta[n] + \delta[n-1]}{2} = \{0.5, 0.5\}$$

and the frequency response (see, e.g., Problem 42 and 55)

$$H(\omega) = \frac{1 + e^{-j\omega}}{2}$$

which in DSP literature is normally written as  $H(e^{j\omega})$ . The corresponding flow (block) diagram is plotted in Figure 30. In this case computation “flows” into right direction without any feedback loops – the filter type is now FIR (finite impulse response, FIR), whereas Problem 22 shows an example of the other type IIR.

Slow changes in the signal (temperature) mean low frequencies and quick changes mean high frequencies. Averaging smoothens the signal. In DSP, we say that it is a *lowpass filter* while it preserves low frequencies in the signal but attenuates high frequencies. High-frequency variation is often considered as *noise*.

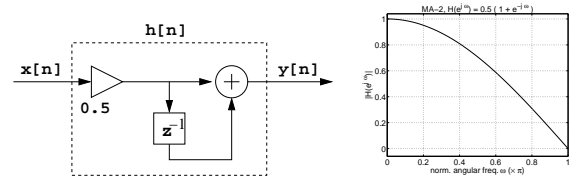


Figure 30: Problem 21: (a) Flow (block) diagram of the filter. Input  $x[n]$  is filtered by the system  $h[n]$ , which is dashed and includes everything between input and output, and the output is  $y[n]$ . A delay unit (register), here “ $z^{-1}$ ”, is sometimes named as “D”. There are no loops and therefore the filter type is FIR. (b) Magnitude response  $|H(e^{j\omega})|$  of MA-2 filter. It can be seen that it is a lowpass filter.

**Remark.** The sliding window can be increased in order to have smoother results. A weekly average would be MA-7, that is,  $y[n] = (1/7) \cdot \sum_{k=0}^6 x[n-k]$ . MA-filter can be also noncausal (not real-time), e.g.,  $y[n] = (1/2) \cdot (x[n+1] + x[n])$ , if needed. MA-filter can even be implemented in a recursive manner, which can make computation much lighter, see (Smith: DSPguide, Ch. 15, p. 283). Border effects occurring in start and end have to be dealt with in an appropriate way.

22. **Problem:** Compute an average temperature by adding half of temperature today and half of average temperature yesterday. Use the same daily temperature data as in Problem 21:

July	1st	2nd	3rd	4th	5th	6th	7th
°C	12	16	15	22	20	24	23

[L0287] **Solution:** Take a new sample and the last computed output value, sum them together, divide by two, and go on to the following point. The purpose of averaging is to suppress quick changes, possibly noise, in the signal.

In order to compute the average temperature on 3rd July, we take temperature of 3rd and the average of 2nd, and compute mean of them as shown below. Borders have to be dealt with in appropriate way.

	1st	2nd	3rd	4th	5th	6th	7th	
$x[n]$	12	16	15	22	20	24	23	
$y[n-1]$		6	11	13	17.5	18.8	21.4	22.2
$y[n]$	6	11	13	17.5	18.8	21.4	22.2	11.1

Now the averaging process can be thought as a filtering operation:

$$y[n] = \frac{x[n] + y[n-1]}{2} = \{6, 11, 13, 17.5, 18.8, 21.4, 22.2, 11.1, 5.6, \dots\}$$

This corresponds a LTI system with the infinite-length impulse response

$$h[n] = 0.5 \cdot 0.5^n \mu[n] = \{0.5, 0.25, 0.125, \dots\}$$

and the frequency response (see, e.g., Problem 42 and 55)

$$H(e^{j\omega}) = \frac{0.5}{1 - 0.5e^{-j\omega}}$$

The corresponding flow (block) diagram is plotted in Figure 31. In this case there exists a feedback loop – the filter type is now IIR (infinite impulse response, IIR), whereas Problem 21 shows an example of the other type FIR.

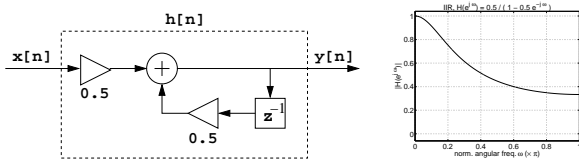


Figure 31: Problem 21: (a) Flow (block) diagram of the filter. Input  $x[n]$  is filtered by the system  $h[n]$ , which is dashed and includes everything between input and output, and the output is  $y[n]$ . The system includes a feedback loop which is equivalent that the system is IIR. (b) Magnitude response  $|H(e^{j\omega})|$  of an IIR filter. It can be seen that it is a lowpass filter.

23. **Problem:** Express the input-output relations of the discrete-time systems in Figure 32.

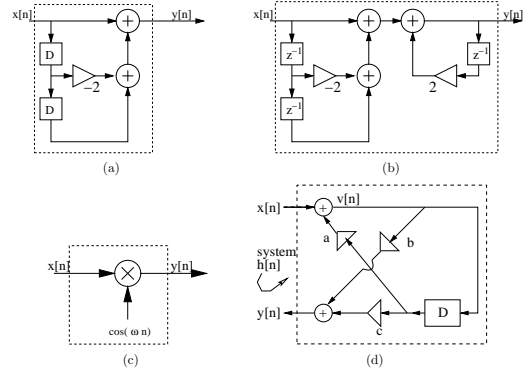


Figure 32: Discrete-time systems for Problems 23, 27, and 28.

[L0244] **Solution:** In this problem there are several types of discrete-time systems. Notice that the scope of this course is LTI systems (linear and time-invariant). LTI systems are very easy to detect, they are relatively simple but very useful. In this course the system input  $x[n]$  and output  $y[n]$  are 1-dimensional except some examples with pictures (2D). For LTI-systems the input-output relation can be written with a difference equation or a set of difference equations.

There are some basic operations on sequences (signals) in discrete-time systems ( $x$  refers to input to the system / operation,  $y$  output) shown also in Figure 33:

- sum of signals (sequences)  $y_1[n] = x_1[n] + x_2[n]$
- signal multiplication (by constant)  $y_1[n] = a x_1[n]$
- delay or advance of signal  $y_1[n] = x_1[n \pm k]$
- product of signals, modulator (non-LTI systems)  $y_1[n] = x_1[n] \cdot x_2[n]$
- branch / pick-off node  $y_1[n] = x_1[n], y_2[n] = x_1[n]$

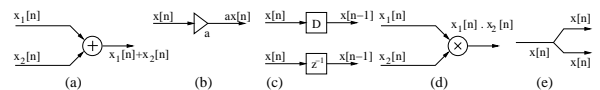


Figure 33: Problem 23: Basic operations in discrete-time systems.

Now we analyze the systems depicted in Figure 32.

- a) Difference equation:  $y[n] = x[n] - 2x[n-1] + x[n-2]$ . LTI-filter, type FIR (see Problem 24).
- b) The memory registers / unit delays are often drawn either “D” (delay) or “ $z^{-1}$ ” (refers to a delay in z-transform). Note that the output is fed back in the loop. The left part of the filter is the same as (a). The sequence right after the second summing on top line is  $y[n]$  which goes both to the output and down to feedback loop. Therefore the terms coming into the last summing unit are  $x[n] - 2x[n-1] + x[n-2]$  from left and  $2y[n-1]$  from the loop. The difference equation is

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

The system is LTI and type IIR (see Problem 24).

- c) Input signal  $x[n]$  is multiplied by a sequence  $\cos(\omega n)$  (not a constant). This operation is called modulation and is not LTI. The relation can be written as

$$y[n] = x[n] \cdot \cos(\omega n)$$

- d) This is so called lattice structure (Mitra 2Ed Sec. 6 / 3Ed Sec. 8). In order to get relationship between  $x[n]$  and  $y[n]$  temporary variables are used after each summing unit. In this case, there is one temporary variable  $v[n]$ , and the set of difference equations is

$$\begin{aligned} v[n] &= x[n] + a v[n-1] \\ y[n] &= b v[n] + c v[n-1] \end{aligned}$$

The temporary variable  $v[n]$  can be simplified away, but it is easier to determine the transfer function  $H(z)$  in frequency domain and then apply inverse z-transform, which is discussed later. The system is LTI and IIR (see Problem 24).

**Remark.** The simplified difference equation for the system in (d) can be received by eliminating all temporary  $v[n]$  sequences:

$$\begin{aligned} x[n] &= v[n] - a v[n-1] & | & \text{ x on left side} \\ y[n] &= b v[n] + c v[n-1] & | & \text{ y on left side} \\ -bx[n] &= -b v[n] + ab v[n-1] & | & \text{ y[n] - bx[n] cancels v[n]} \\ -ay[n-1] &= -ab v[n-1] - ac v[n-2] \\ -cx[n-1] &= -c v[n-1] + ac v[n-2] & | & \text{ all v[n-1], v[n-2] cancelled} \end{aligned}$$

which finally gives  $y[n] = ay[n-1] + bx[n] + cx[n-1]$ . The discrete-time system does some computation for sequences of numbers. Therefore it is straightforward to write down a computer program, e.g. in (a),

```
x1 := 0; x2 := 0;
while TRUE {
  x2 := x1;
  x1 := x0;
  x0 := read_next_input(input_stream);
  y := x0 - 2*x1 + x2;
  write_output(output_stream, y);
}
```

or if all samples are known in advance and written in vector  $x$ ,

```
for (k = 2; k <= length(x); k++) {
  y[k] := x[k] - 2*x[k-1] + x[k-2];
}
```

24. **Problem:** Look at the flow (block) diagrams in Figure 34.

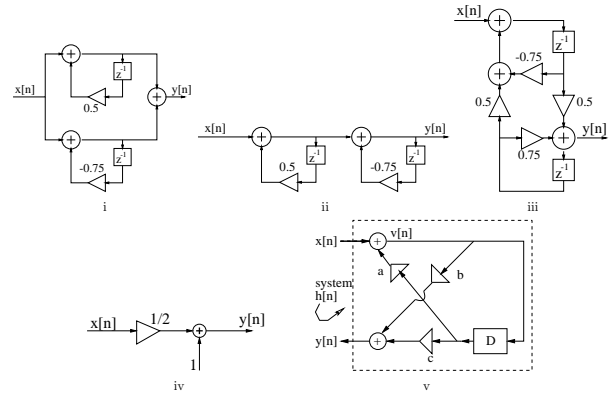


Figure 34: Flow diagrams of Problem 24.

- a) What does LTI mean? In what ways can the system be proved (Problem 25) or shown to be LTI?
- b) Which systems are linear and time-invariant (LTI) without any computation?
- c) Which systems have feedback?
- d) Which LTI systems are FIR and which are IIR?

[L0245] **Solution:** In this problem we try to recognise LTI systems by their layout.

- a) LTI = linear AND time-invariant (=shift-invariant) system. These two properties belong to a system not to a signal. Other system properties can be, e.g. stability, causality, or if it needs memory or if it can be inverted.

See Problem 25 for mathematical proofs.

Recognition of LTI systems from the flow (block) diagrams: there are only (1) sums of signals, (2) multiplication by a constant, (3) delays or advances. The components were introduced in Problem 23, see Figure 33 at page 40.

LTI systems can be represented with a linear constant coefficient difference (or differential in case of analog system) equation

$$\sum_k d_k y[n-k] = \sum_k p_k x[n-k]$$

where  $\{d_k\}$  and  $\{p_k\}$  are constants. Often in practice, we use causal finite-dimensional LTI systems  $\sum_{k=0}^N d_k y[n-k] = \sum_{k=0}^M p_k x[n-k]$ , where the order of the system is given by  $\max\{N, M\}$  (Mitra 2Ed Sec. 2.6.0 / 3Ed Sec. 2.7.0). If the system cannot be written in the format above, it is not a (causal) LTI system.

- b) LTI? Only summing, delays, amplifications by constants. (i) Yes, (ii) Yes, (iii) Yes, (iv) No, adding a constant, (v) Yes.
- c) Feedback means that some of the output (or internal) values are fed back in the system. Computation can be said to be recursive or iterative. There are loops in (i), (ii), (iii), and (v).
- d) FIR = Finite (length) Impulse Response. IIR = Infinite (length) Impulse Response. If the system has a feedback loop somewhere in the structure, it is also IIR at the same time. The output value is computed using older output values, i.e. there is recursion. This can be seen that there are also terms  $y[n-k]$ ,  $k \neq 0$ , in the difference equation. If there is no loop and computation flows forward all the time, then the system is FIR. This can be seen that there is only the term  $y[n]$  in the left side of the difference equation above. FIR: (iv) has an impulse response of finite length but it is not LTI. IIR: (i), (ii), (iii), and (v) have infinite (length) impulse response because of feedback loops.

25. **Problem:** For each of the following discrete-time systems, determine whether or not the system is (1) linear, (2) causal, (3) stable, and (4) shift-invariant. The sequences  $x[n]$  and  $y[n]$  are the input and output sequences of the system.

- a)  $y[n] = x^3[n]$ .
- b)  $y[n] = \gamma + \sum_{l=-2}^2 x[n-l]$ ,  $\gamma$  is a nonzero constant.
- c)  $y[n] = \alpha x[-n]$ ,  $\alpha$  is a nonzero constant.

[L0238] **Solution:** Properties of the discrete-time system, see (Mitra 2Ed Sec. 2.4.1, 2.5.3, 2.5.4 / 3Ed Sec. 2.4.2, 2.5.3, 2.5.4).

**Linearity:**

If  $y_1[n]$  and  $y_2[n]$  are the responses to the input sequences  $x_1[n]$  and  $x_2[n]$ , respectively, then for an input

$$x[n] = \alpha x_1[n] + \beta x_2[n],$$

the response is given by

$$y[n] = \alpha y_1[n] + \beta y_2[n].$$

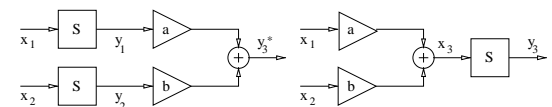


Figure 35: Linearity. If the linear combination  $y_3^*$  of outputs of  $x_1$  and  $x_2$  is the same as the output  $y_3$  of the linear combination of inputs, then the system  $S$  is linear.

Linear systems offer possibility to use *superposition*, that is, a signal can be divided to small portions, processed individually, and summed back.

**Remark.** When considering a constant-coefficient difference equation like  $y[n] + 0.5y[n-1] = x[n] + 0.5x[n-1]$  the system is **not** linear, if the initial conditions are not zero, i.e.,  $y[-1] \neq 0$ . When initial values are zero, the system is said to be in rest. See (Mitra 2Ed Ex. 2.30, p. 92 / 3Ed Ex. 2.37, p. 92).

**Causality:**

The  $n_0$ -th output sample  $y[n_0]$  depends only on previous output values and input samples  $x[n]$  for  $n \leq n_0$ , and does not depend on input samples for  $n > n_0$ . In case of a LTI system, the system is causal if and only if impulse response  $h[n] = 0$  for all  $n < 0$ . In other words, if the system does not predict anything, it is causal.

**Stability:**

Bounded input, bounded output (BIBO) stability: If a bounded input ( $B_x$  is a finite constant)

$$|x[n]| < B_x < \infty, \quad \forall n$$

produces a bounded output ( $B_y$  is a finite constant)

$$|y[n]| < B_y < \infty, \quad \forall n$$

as a response then the system is BIBO stable (see (a) and (b) at Page 45 for details). In case of LTI-system, the system is stable if and only if  $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$ .

**Time/Shift-invariance:**

If  $y_1[n]$  is the response to an input  $x_1[n]$ , then the response to an input

$$x[n] = x_1[n - n_0]$$

is simply

$$y[n] = y_1[n - n_0],$$

where  $n_0$  is any positive or negative integer.

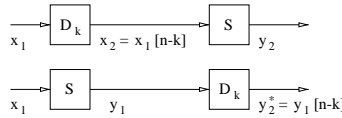


Figure 36: Time invariance. If the output  $y_2$  of delayed input is the same as delayed output  $y_2^*$ , then the system  $S$  is time-invariant.

a)  $y[n] = x^3[n]$ .

Take inputs  $x_1[n]$  and  $x_2[n]$ , the outputs are then  $y_1[n] = x_1^3[n]$  and  $y_2[n] = x_2^3[n]$ . Now the linear combination of the input signals is  $x_3[n] = \alpha x_1[n] + \beta x_2[n]$  and the output is

$$y_3[n] = (\alpha x_1[n] + \beta x_2[n])^3 \neq \alpha x_1^3[n] + \beta x_2^3[n] = \alpha y_1[n] + \beta y_2[n].$$

Hence the system is **not linear**.

Since there is no output before the input hence the system is **causal**.

The system is **stable**: Assume  $|x[n]| < B_x$ , then

$$|y[n]| = |x^3[n]| \leq |x[n]|^3 < B_x^3 = B_y < \infty.$$

The system is **time-invariant**: Assume input  $x_1[n]$  and output  $y_1[n]$ , then response of input  $x[n] = x_1[n - n_0]$  is

$$y[n] = (x[n])^3 = (x_1[n - n_0])^3 = y_1[n - n_0]$$

b)  $y[n] = \gamma + \sum_{l=-2}^2 x[n - l]$ ,  $\gamma$  is a nonzero constant.

Use linear combination  $\alpha x_1[n] + \beta x_2[n]$  as the input

$$\begin{aligned} y_3[n] &= \gamma + \sum_{l=-2}^2 (\alpha x_1[n - l] + \beta x_2[n - l]) \\ &= 0.5\gamma + \alpha \sum_{l=-2}^2 x_1[n - l] + 0.5\gamma + \beta \sum_{l=-2}^2 x_2[n - l] \\ &\neq \alpha \gamma + \alpha \sum_{l=-2}^2 x_1[n - l] + \beta \gamma + \beta \sum_{l=-2}^2 x_2[n - l] \\ &= \alpha y_1[n] + \beta y_2[n], \end{aligned}$$

where  $\alpha$  and  $\beta$  are not fixed. The system is hence **nonlinear**.

The system is **not causal**, because there can be output before input, when  $l \in [-2, -1]$ .

System is **stable**: Assume bounded input  $|x[n]| < B_x$ , then

$$|y[n]| = |\gamma + \sum_{l=-2}^2 x[n - l]| \leq |\gamma| + \sum_{l=-2}^2 |x[n - l]| < |\gamma| + 5B_x = B_y < \infty$$

The system is also **time-invariant**: Assume input  $x_1[n]$  and output  $y_1[n]$ , then response of input  $x[n] = x_1[n - n_0]$  is

$$y[n] = \gamma + \sum_{l=-2}^2 x_1[n - n_0 - l] = y_1[n - n_0].$$

c)  $y[n] = \alpha x[-n]$ ,  $\alpha$  is a nonzero constant.

The system is **linear**, **stable** and **noncausal**.

Assume inputs  $x_1[n], x_2[n]$  and outputs  $y_1[n], y_2[n]$ , respectively, then

$$\begin{aligned} y_1[n] &= \alpha x_1[-n], \\ y_2[n] &= \alpha x_2[-n]. \end{aligned}$$

Let  $x[n] = x_1[n - n_0]$ , then

$$\begin{aligned} y[n] &= \alpha x[-n] = \alpha x_1[-n - n_0] \\ &\neq \alpha x_1[n_0 - n] = \alpha x_1[-(n - n_0)] = y_1[n - n_0] \end{aligned}$$

and the system is **not time-invariant**.

26. **Problem:** A LTI system with an input  $x_1[n] = \{1, 1, 1\}$  gives an output  $y_1[n] = \{0, 2, 5, 5, 3\}$ . If a new input is  $x_2[n] = \{1, 3, 3, 2\}$ , what is the output  $y_2[n]$ ?

[L0281] **Solution:** While the LTI system is both linear (L) and time-invariant (TI) we can apply *superposition*. Now  $x_2[n]$  can be synthesized as a sum of original  $x_1[n]$  and a shifted and scaled  $2x_1[n - 1]$ :

$$x_2[n] = x_1[n] + 2x_1[n - 1] = \{1, 1, 1\} + \{0, 2, 2, 2\} = \{1, 3, 3, 2\}$$

Because of LTI, the output  $y_2[n]$  is also a sum of original  $y_1[n]$  and a shifted and scaled  $2y_1[n - 1]$ , as shown in Figure 37

$$y_2[n] = y_1[n] + 2y_1[n - 1] = \{0, 2, 5, 5, 3\} + \{0, 0, 4, 10, 10, 6\} = \{0, 2, 9, 15, 13, 6\}$$

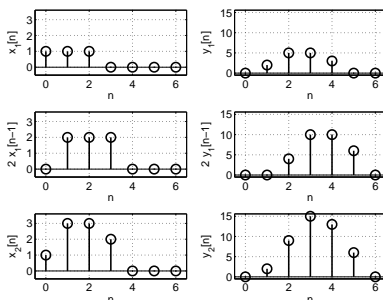


Figure 37: Problem 26: Left column: sequences  $x_1[n]$ ,  $2x_1[n - 1]$ , and  $x_2[n] = x_1[n] + 2x_1[n - 1]$ . Right column: sequences  $y_1[n]$ ,  $2y_1[n - 1]$ , and  $y_2[n] = y_1[n] + 2y_1[n - 1]$ . This holds for linear and time-invariant (LTI) systems.

**Remark.** A standard way to solve this problem is to compute deconvolution of  $x_1[n]$  and  $y_1[n]$ , and then apply the result to the new input  $x_2[n]$ . See Problem 32.

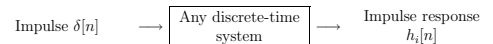
If we know how the LTI system modifies an impulse  $\delta[n]$  into an output  $y[n] = h[n]$ , we can build an arbitrary synthesis  $x[n] = \sum_k x[k]\delta[n - k]$  and receive the output  $y[n]$  by convolving  $x[n]$  and  $h[n]$ . See Problem 27 or (*Smith: DSPguide, Ch. 5, p. 98-99*).

Notice that a synthesis signal  $x[n]$  can be decomposed in several ways. For example, the synthesis of  $15 + 25$  is always 40, but 40 can be decomposed to  $1 + 39$ ,  $-1 + 41$ ,  $20 + 20$ , etc. In case of signals, Fourier analysis (decomposition) or Fourier synthesis (inverse Fourier) is applied.

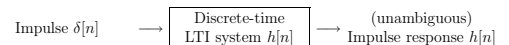
27. **Problem:** Impulse response  $h[n]$  is the response of the system to the input  $\delta[n]$ .

- a) What is the impulse response of the system in Figure 32(a)? What is the connection to the difference equation? Is this LTI system stable/causal?
- b) What are the first five values of impulse response of the system in Figure 32(b)? Hint: Fetch the input  $\delta[n]$  and read what comes out. Is it possible to say something about stability or causality of the system?
- c) What are the first five values of impulse response of the system in Figure 32(d)?

[L0237] **Solution:** Impulse response  $h[n]$  is the response of the system to the input  $\delta[n]$ .



Here  $h_i[n]$  may get different values at different moments. However, a discrete-time LTI (linear and time-invariant) has only one  $h[n]$  which specifies the system completely (*Mitra 2Ed Sec. 2.5.1 / 3Ed Sec. 2.5.1*).



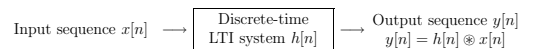
For a LTI system (see Problems 24 and 25) the *stability condition* is

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

and the *causality condition*

$$h[n] = 0, \quad \forall n < 0$$

If the impulse response  $h[n]$  is known for a LTI system, then the output  $y[n]$  can be computed for any input  $x[n]$  by convolution.



a) Difference equation of the system is  $y[n] = x[n] - 2x[n - 1] + x[n - 2]$ . Let the input be  $\delta[n]$  and read what comes out.

$n$	$x[n]$	$-2x[n - 1]$	$x[n - 2]$	$y[n] = x[n] - 2x[n - 1] + x[n - 2]$
...	0	0	0	0
-1	0	0	0	0
0	1	0	0	1
1	0	-2	0	-2
2	0	0	1	1
3	0	0	0	0
...	0	0	0	0

The impulse response is

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

The length  $L\{\cdot\}$  of the impulse response is finite,  $L\{h[n]\} = 3 < \infty$ . So, the filter is FIR (finite (length) impulse response).

Notice that in case of FIR filter (no feedbacks, flow always going forward), the impulse response can be easily gotten from the corresponding difference equation just by replacing  $y$  by  $h$  and each  $x$  by  $\delta$  (Mitra 2Ed Sec. 2.5.1 / 3Ed Sec. 2.5.1). In case of IIR, this leads to a recursive computation of  $h[n]$ .

All FIR systems are always stable because the length of impulse response is finite, and therefore also the sum of absolute values is finite:  $\sum_n |h[n]| < \infty$ , in this case  $\sum_n |h[n]| = 1 + 2 + 1 = 4 < \infty$ .

This FIR system is causal while  $h[n] = 0$  for all  $n < 0$ . In the difference equation there is no future terms ( $x[n + k]$ ,  $k > 0$ ) in right side of equation.

- b) There is a feedback in the filter whose difference equation is

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

The impulse response is the response for impulse, so just feed a delta function in and read what comes out. The initial value  $y[-1]$  is by default zero. If  $y[-1]$  were non-zero, the system would not be linear, i.e., not LTI (Mitra 2Ed Sec. 2.4.1 / 3Ed Sec. 2.4.2).

$n$	$x[n]$	$-2x[n - 1]$	$x[n - 2]$	$2y[n - 1]$	$y[n] = 2y[n - 1] + x[n] - 2x[n - 1] + x[n - 2]$
...	0	0	0	0	0
-1	0	0	0	0	0
0	1	0	0	0	1
1	0	-2	0	2	0
2	0	0	1	0	1
3	0	0	0	2	2
4	0	0	0	4	4
...	0	0	0	...	...

Computing values in the right-most column is equivalent to a recursive ( $h[n - 1]$  needed for  $h[n]$ ) expression

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

which is found by replacing each  $y$  by  $h$  and each  $x$  by  $\delta$ . Because of the loop,  $L\{h[n]\} = \infty$ , which corresponds IIR filter (infinite (length) impulse response).

The system is clearly causal because  $h[n] = 0$  for all  $n < 0$ . From the difference equation we see that in order to compute  $y[n]$  we need only present and past values of  $x$  and past values of  $y$ . Hence, the system does not “predict” anything.

The system does not seem to be stable, while the output starts growing 1, 2, 4, ... without bound. The stability of IIR systems has to be checked every time, and there will be easy tools for that later (poles of  $H(z)$  inside the unit circle).

The impulse response can be expressed in a closed form from the transfer function  $H(z)$  by inverse  $z$ -transform (discussed later). From the sequence  $h[n] = \{1, 0, 1, 2, 4, \dots\}$  we can guess that the closed form equation (one possible expression) is  $h[n] = \delta[n] + 2^{n-2}\mu[n - 2]$ .

- c) A set of difference equations can be written,

$$\begin{aligned} v[n] &= x[n] + a v[n - 1] \\ y[n] &= b v[n] + c v[n - 1] \end{aligned}$$

Just like in (b), the columns for temporary values are computed, and finally the first values of the impulse response are

$$h[n] = \{b, ba + c, ba^2 + ca, ba^3 + ca^2, ba^4 + ca^3, \dots\}$$

from which it can be guessed that the closed form representation for the impulse response is  $h[n] = ba^n \mu[n] + ca^{n-1} \mu[n - 1]$ .

Difference equation between  $x$  and  $y$  has been computed in Problem 23(d). It can be derived even easier when utilizing  $z$ -transform and removing temporary variable  $v$  in transform domain. See an example in Problem 64.

28. **Problem:** Step response  $s[n]$  is the response of the system to the input  $\mu[n]$ . What are the step responses of systems in Figures 32(a) and (b)?

[L0273] **Solution:** Unit step response, or shortly *step response*  $s[n]$  is the response of the system to the input  $\mu[n]$  (Mitra 2Ed Sec. 2.4.2 / 3Ed Sec. 2.4.3). Step response can be computed easily from the impulse response  $h[n]$  by cumulative sum (accumulator)

$$s[n] = \sum_{k=-\infty}^n h[k]$$

Now, in (a) the impulse response is  $h[n] = \delta[n] - 2\delta[n - 1] + \delta[n - 2]$ , and the step response is

$$s[n] = \{\dots, 0, 0, \underline{1}, -1, 0, 0, \dots\}$$

which can be also seen by feeding ones to the input and reading the output. The steady-state response (Mitra 2Ed Sec. 4.2.3 / 3Ed Sec. 3.8.5) converges quickly to zero.

In (b) the impulse response diverges  $h[n] = \delta[n] + \delta[n - 2] + 2\delta[n - 3] + 4\delta[n - 4] + \dots$ , as well as the step response

$$s[n] = \{\dots, 0, 0, \underline{1}, 2, 4, 8, \dots\}$$

29. **Problem:** Compute the linear convolution of two signals  $x_1(t)$  and  $x_2(t)$

$$y(t) = x_1(t) \otimes x_2(t) = \int_{-\infty}^{\infty} x_1(\tau) \cdot x_2(t - \tau) d\tau$$

in both cases (a) and (b) in Figure 38. The arrows in (b) are impulses  $\delta(t)$ .

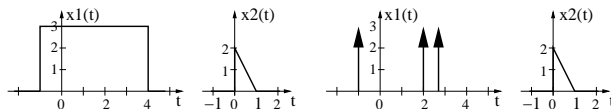


Figure 38: Problem 29: signals  $x_1(t)$  and  $x_2(t)$  to be convolved, left: (a), right: (b).

[L0272] **Solution:** **Continuous-time linear convolution** of two signals  $x_1(t)$  and  $x_2(t)$  is defined by

$$y(t) = x_1(t) \otimes x_2(t) = \int_{-\infty}^{\infty} x_1(\tau) \cdot x_2(t - \tau) d\tau$$

You can see an example of graphical convolution in Java applet in URL <http://www.jhu.edu/~signals/convolve/index.html>. Sketch the signals  $x_1(t)$  and  $x_2(t)$  of Figure 38 into the boxes. The other signal is flipped around. When sliding the flipped signal to right over the other signal, the integral of the product is computed. At certain point  $t_0$  the integral gives the convolution output  $y(t_0)$ .

The results of (a) can be seen in Figure 39(a). In (b) the arrows are impulses  $\delta(t)$  which are signals having the area of unity and being infinitely narrow, i.e. the height in infinite. Convoluting a signal with an impulse  $\delta(t)$  can be considered as copying the signal at each place where impulse lies, see Figure 39(b).

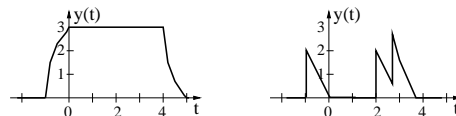


Figure 39: Problem 29: convolution results  $y(t)$ , left: (a), right: (b).

**Remark.** The continuous-time convolution contains the product of two signals and taking integral of the product. In practice, the convolution can seldom be computed in closed form. However, in (a) the signals are

$$\begin{aligned} x_1(t) &= \begin{cases} 3, & -1 \leq t < 4 \\ 0, & \text{elsewhere} \end{cases} \\ x_2(t) &= \begin{cases} 2 - 2t, & 0 \leq t < 1 \\ 0, & \text{elsewhere} \end{cases} \end{aligned}$$

The flipped signal is  $x_2(t - \tau) = 2 - 2t + 2\tau$ , and the convolution integral is  $y(t) = \int_{-\infty}^{\infty} x_1(\tau) \cdot x_2(t - \tau) d\tau$ . The convolution can be computed in five cases when sliding  $x_2$

from left to right: (1)  $t < -1$ , product of signals is zero, (2)  $-1 < t < 0$ ,  $x_2$  “penetrating”, (3)  $0 < t < 4$ , “stable” case, (4)  $4 < t < 5$ ,  $x_2$  “leaving”, (5)  $t > 5$ , again zero.

$$y(t)_{(1)} = \int_{-\infty}^{-1} 0 \cdot (2 - 2t + 2\tau) d\tau = 0, \quad t < -1$$

$$y(t)_{(2)} = \int_{-1}^t 3 \cdot (2 - 2t + 2\tau) d\tau = 3 - 3t^2, \quad -1 \leq t < 0$$

$$y(t)_{(3)} = \int_{t-1}^t 3 \cdot (2 - 2t + 2\tau) d\tau = 3, \quad 0 \leq t < 4$$

$$y(t)_{(4)} = \int_{t-1}^5 3 \cdot (2 - 2t + 2\tau) d\tau = 3t^2 - 30t + 75, \quad 4 \leq t < 5$$

$$y(t)_{(5)} = \int_5^{\infty} 0 \cdot (2 - 2t + 2\tau) d\tau = 0, \quad t \geq 5$$

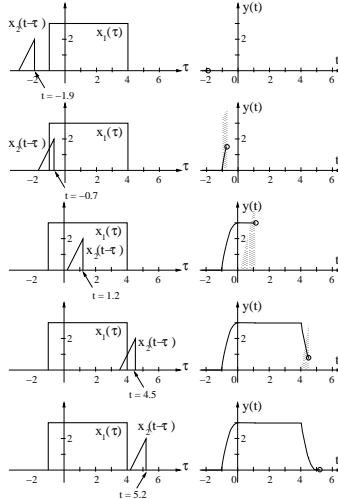


Figure 40: Remark of Problem 29: Detailed example of continuous convolution. Five figures represent examples of  $y(t)_{(i)}$ ,  $i = 1 \dots 5$ , as given in equations above. The shaded areas depict the integrand  $x_1(\tau) \cdot x_2(t - \tau)$  in the corresponding range. Note that they are not in scale in second and third figure, the peak should be at 6.

$n$	$x[n] = \delta[n] + \delta[n-1]$	$x[n-1]$	$y[n] = x[n] + x[n-1]$
-1	0	0	0 + 0 = 0
0	1	0	1 + 0 = 1
1	1	1	1 + 1 = 2
2	0	1	0 + 1 = 1
3	0	0	0 + 0 = 0
4	0	0	0 + 0 = 0

So, the result is  $x[n] \otimes h[n] = \{1, 2, 1\} = \delta[n] + 2\delta[n-1] + \delta[n-2]$ , and the length is  $L\{y[n]\} = 3$ . The starting point can be checked:  $A\{y[n]\} = A\{x[n]\} + A\{h[n]\} = 0$ .

b) Another way (on-line) is computing output values at each time moment  $n$ . Graphically this means **inverting** (flipping around) the other sequence, **sliding** it over the other, and computing the output value as a dot sum. This is also illustrated with figures in (Mitra 2Ed Ed. 2.24, p. 73-75 / 3Ed Ed. 2.26, p. 80-83).

Now when  $x_1[n] = \delta[n] + 5\delta[n-1]$  and  $x_2[n] = -\delta[n-1] + 2\delta[n-2] - \delta[n-3] - 5\delta[n-4]$ , then  $L\{x_1[n]\} = 2 + 4 - 1 = 5$  and  $A\{x_1[n]\} = 0 + 1 = 1$ . Therefore we know that the convolution result is of form  $x[n] = a_1\delta[n-1] + a_2\delta[n-2] + a_3\delta[n-3] + a_4\delta[n-4] + a_5\delta[n-5]$ .

$$n = 1: \quad x[1] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[1-k]$$

$$= 0 + \underbrace{(x_1[0] \cdot x_2[1-0])}_{-1} + \underbrace{(x_1[1] \cdot x_2[1-1])}_{0} + 0$$

$$= -1$$

$$n = 2: \quad x[2] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[2-k]$$

$$= 0 + (x_1[0] \cdot x_2[2-0]) + (x_1[1] \cdot x_2[2-1]) + 0$$

$$= 2 + (-5) = -3$$

$$n = 3: \quad x[3] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[3-k]$$

$$= 0 + (x_1[0] \cdot x_2[3-0]) + (x_1[1] \cdot x_2[3-1]) + 0$$

$$= -1 + 10 = 9$$

$$n = 4: \quad x[4] = -5 + (-5) = -10$$

$$n = 5: \quad x[5] = -25$$

The procedure is represented stepwise, and step  $n = 3$  is shown also in Figure 41. In the top line of the figure there is the sequence  $x_1[k] = \{\dots, 0, 1, 5, 0, \dots\}$ , in the second line the shifted and inverted sequence  $x_2[n-k]$ . It slides from left to right when  $n$  increases, and at  $n = 3$  it is  $x_2[3-k] = \{\dots, 0, -5, -1, 2, -1, 0, \dots\}$ . The point-wise product of sequences in top rows is shown in the third line:  $\{x_1[k]x_2[3-k]\} = \{\dots, 0, 0, (-5), 1 \cdot (-1), 5 \cdot 2, 0 \cdot (-1), 0, \dots\} = \{\underline{-1}, 10\}$ . The convolved value  $x[3]$  is the sum of values in the third row:  $x[3] = \sum_{k=-\infty}^{\infty} x_1[k]x_2[3-k] = -1 + 10 = 9$ . In the bottom line there is the result for  $n \leq 3$ , and  $n = 3$  underlined, and results for  $n > 3$  are to be computed.

c) The convolution can be computed as a sum of shifted and scaled sequences.

30. **Problem:** Linear convolution of two sequences is defined as (Mitra 2Ed Eq. 2.64a, p. 72 / 3Ed Eq. 2.73a, p. 79)

$$y[n] = h[n] \otimes x[n] = x[n] \otimes h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

- Compute  $x[n] \otimes h[n]$ , when  $x[n] = \delta[n] + \delta[n-1]$ , and  $h[n] = \delta[n] + \delta[n-1]$ . What is the length of the convolution result?
- Compute  $x_1[n] \otimes x_2[n]$ , when  $x_1[n] = \delta[n] + 5\delta[n-1]$ , and  $x_2[n] = -\delta[n-1] + 2\delta[n-2] - \delta[n-3] - 5\delta[n-4]$ . What is the length of the convolution result? Where does the output sequence start?
- Compute  $h[n] \otimes x[n]$ , when  $h[n] = 0.5^n \mu[n]$ , and  $x[n] = \delta[n] + 2\delta[n-1] - \delta[n-2]$ . What is the length of the convolution result?

[L0239] **Solution:** Discrete-time linear convolution of two sequences  $h[n]$  and  $x[n]$  is

$$y[n] = h[n] \otimes x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

The convolution is an operation for two sequences (Mitra 2Ed Sec. 2.5.1, p. 71 / 3Ed Sec. 2.5.1, 2.5.2, p. 78). There are several ways to get the convolution result. First, in (a) the convolution is considered as filtering, the other sequence is the input and the other is the impulse response of the system, and the convolution result is the output of the system. Second, in (b) a graphical way of inverting and sliding the sequences over each other is represented. In (c) the convolution is considered as a sum of shifted and scaled sequences, “tabular method” in (Mitra 3Ed Sec. 2.5.2). However, even if three ways are introduced separately, they all rely on the same (and simple) definition of the convolution.

When computing discrete-time convolution  $y[n] = x[n] \otimes h[n]$ , it is nice know a couple of rules. Let  $L\{\cdot\}$  be a length of a sequence, e.g.  $x[n] = \{3, 2, 0, 5, -2\}$ , then  $L\{x[n]\} = 5$ .

Because LTI-system is shift-invariant, the starting point of the convolution result can be determined as a sum of starting points of the convolved sequences. Let  $A\{\cdot\}$  be the index number of the first non-zero element, e.g.,  $A\{x[n]\} = -1$ .

It is easily seen that for the convolution result  $y[n]$  it holds

$$L\{y[n]\} = L\{x[n]\} + L\{h[n]\} - 1$$

$$A\{y[n]\} = A\{x[n]\} + A\{h[n]\}$$

There are also some nice convolution demos in Internet, e.g. <http://www.jhu.edu/~signals/discreteconv2/index.html>.

- Consider convolution as filtering with the input sequence  $x[n] = \delta[n] + \delta[n-1] = \{1, 1\}$ , and the impulse response  $h[n] = \delta[n] + \delta[n-1] = \{1, 1\}$ , of the system. The corresponding difference equation is  $y[n] = x[n] + x[n-1]$ , that is, the output is just the sum of the present and previous value in the input. (You can draw the flow (block) diagram for the system and verify the computation.)

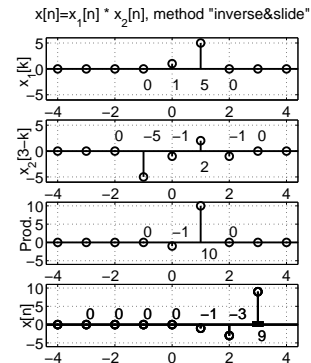


Figure 41: Problem 30(b): Linear convolution using “invert and slide”. Caption from the step  $n = 3$ , i.e. computing the value  $x[3] = 9$ . See the text for more details. There is a demo Matlab program `linconv.m` to demonstrate the computation in the course web pages.

Now,  $h[n] = 0.5^n \mu[n]$ , and  $x[n] = \delta[n] + 2\delta[n-1] - \delta[n-2]$ , in other words  $x[0] = 1$ ,  $x[1] = 2$ ,  $x[2] = -1$ , and  $x[n] = 0$ , elsewhere. The division into three parts on third line emphasizes the fact that a scalar  $x[k]$  is zero with all values of  $k$  except  $k = \{0, 1, 2\}$ .

$$y[n] = x[n] \otimes h[n]$$

$$= \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

$$= \sum_{k=-\infty}^{-1} x[k]h[n-k] + \sum_{k=0}^2 x[k]h[n-k] + \sum_{k=3}^{\infty} x[k]h[n-k]$$

$$= 0 + \sum_{k=0}^2 x[k]h[n-k] + 0$$

$$= \underbrace{x[0]}_{\text{scaling shifted seq.}} \cdot h[n-0] + x[1]h[n-1] + x[2]h[n-2]$$

$$= 1 \cdot h[n] + 2 \cdot h[n-1] - 1 \cdot h[n-2]$$

$$= 0.5^n \mu[n] + 2 \cdot 0.5^{n-1} \mu[n-1] - 0.5^{n-2} \mu[n-2]$$

$$= \delta[n] + 2.5\delta[n-1] + 0.5^n \mu[n-2] \quad | \text{ alternatively}$$

It can be seen that values of  $x[n]$  were scaling factors and sequence  $h[n]$  was shifted each time. While convolution is commutative ( $x_1[n] \otimes x_2[n] = x_2[n] \otimes x_1[n]$ ), one can compute the same using values of  $h[n]$  as scaling factors and shifting  $x[n]$ . The procedure is depicted in Figure 42. While the length of the other sequence is infinite, so is also the length of the convolution.

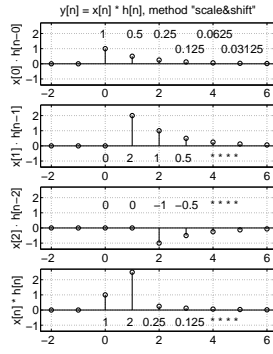


Figure 42: Problem 30(c): Linear convolution using “scaled and shifted sequences”. Top line:  $x[0] \cdot h[n - 0] = 0.5^n \mu[n]$ , second:  $x[1] \cdot h[n - 1] = 2 \cdot 0.5^{n-1} \mu[n - 1]$ , third:  $x[2] \cdot h[n - 2] = -1 \cdot 0.5^{n-2} \mu[n - 2]$ , bottom: convolution result, sum of sequences above.

31. **Problem:** Consider a LTI-system with impulse response  $h[n] = \delta[n - 1] - \delta[n - 2]$  and input sequence  $x[n] = 2\delta[n] + 3\delta[n - 2]$ .
- What is the length of convolution of  $h[n]$  and  $x[n]$  (without computing convolution itself)? Which index  $n$  is the first one having a non-zero item?
  - Compute convolution  $y[n] = h[n] \otimes x[n]$
  - Consider polynomials  $S(x) = 2 + 3x^2$  and  $T(x) = x - x^2$ . Compute the product  $U(x) = S(x) \cdot T(x)$
  - Check the result by computing the polynomial division  $T(x) = U(x)/S(x)$ .

[L0246] **Solution:** An important rule of thumb for finding length  $L\{\cdot\}$  of the linear convolution (different from circular convolution):

$$y[n] = h[n] \otimes x[n] \rightarrow L\{y[n]\} = L\{h[n]\} + L\{x[n]\} - 1$$

The index of the first non-zero item  $A\{\cdot\}$  for finite sequences:

$$y[n] = h[n] \otimes x[n] \rightarrow A\{y[n]\} = A\{h[n]\} + A\{x[n]\}$$

In this case,  $h[n] = \delta[n - 1] - \delta[n - 2]$ , which is drawn as a flow diagram in Figure 43.

- $L\{h[n]\} = 2$ ,  $L\{x[n]\} = 3 \rightarrow L\{y[n]\} = 4$ . Because  $h[n]$  is delayed by one ( $d_h = +1$ ) and  $x[n]$  starts from the origo ( $d_x = 0$ ), also their convolution is delayed by one:  $A\{h[n]\} = 1$ ,  $A\{x[n]\} = 0 \rightarrow A\{y[n]\} = 1$ .  
Now we know that the result is of form:

$$y[n] = a_1\delta[n - 1] + a_2\delta[n - 2] + a_3\delta[n - 3] + a_4\delta[n - 4]$$

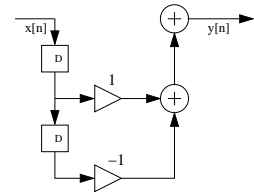


Figure 43: Problem 31: Flow diagram.

- Using values of  $h[n] = \delta[n - 1] - \delta[n - 2]$  as scaling factors

$$\begin{aligned} y[n] &= h[n] \otimes x[n] \\ &= \sum_{k=-\infty}^{\infty} h[k]x[n - k] \\ &= \sum_{k=1}^2 h[k]x[n - k] \\ &= 1 \cdot (2\delta[n - 1] + 3\delta[n - 3]) - 1 \cdot (2\delta[n - 2] + 3\delta[n - 4]) \\ &= 2\delta[n - 1] - 2\delta[n - 2] + 3\delta[n - 3] - 3\delta[n - 4] \end{aligned}$$

- $U(x) = S(x) \cdot T(x) = (2 + 3x^2) \cdot (x - x^2) = 2x - 2x^2 + 3x^3 - 3x^4$ . Notice the correspondence with the result of (b), the delay is the power of  $x$  ( $z^{-1}$  in  $z$ -transform).
- Using long division (*Mitra 2Ed Ex. 3.35 / 3Ed Ex. 6.19*). The polynomials are  $U(x) = 2x - 2x^2 + 3x^3 - 3x^4$  and  $S(x) = 2 + 3x^2$ .

$$\begin{array}{r} -x^2 + x \\ 3x^2 + 2 \overline{) -3x^4 + 3x^3 - 2x^2 + 2x} \\ \underline{-3x^4 + 6x^2} \phantom{+ 2x} \\ \phantom{-3x^4 +} 3x^3 - 2x^2 + 2x \\ \phantom{-3x^4 +} \underline{-3x^3} \phantom{+ 2x} \\ \phantom{-3x^4 +} \phantom{3x^3} - 2x \phantom{+ 2x} \end{array}$$

We get the result  $x - x^2$  as expected ( $h[n] = \delta[n - 1] - \delta[n - 2]$ ). Convolution and deconvolution operations can be computed using products and divisions of polynomials.

32. **Problem:** The impulse response  $h_1[n]$  of a LTI system is known to be  $h_1[n] = \mu[n] - \mu[n - 2]$ . It is connected in cascade (series) with another LTI system  $h_2$  as shown in Figure 44.

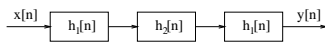


Figure 44: The cascade system of Problem 32.

Compute the impulse response  $h_2[n]$ , when it is known that the impulse response  $h[n]$  of the whole system is shown in Table 6 below.

n	< 0	0	1	2	3	4	> 4
h[n]	0	1	5	9	7	2	0

Table 6: Impulse response of the cascade system in Problem 32.

[L0241] **Solution:** There are three subsystems connected in cascade (series). They are all linear and time-invariant (LTI). The overall impulse response of the whole system is therefore

$$\begin{aligned} h[n] &= (h_1[n] \otimes h_2[n]) \otimes h_1[n] \\ h_2[n] &= (h_1[n] \otimes h_1[n]) \otimes h_2[n] \\ &= \delta[n] + 5\delta[n - 1] + 9\delta[n - 2] + 7\delta[n - 3] + 2\delta[n - 4] \\ &= \{\underline{1}, 5, 9, 7, 2\} \end{aligned}$$

Notice that  $h[n]$  and  $h_1[n]$  are known but  $h_2[n]$  is unknown. If one of the signals to be convolved is unknown and the convolution result is known, the operation to find the unknown is called *deconvolution*, inverse operation of convolution. The procedure of deconvolution is basically the same as that with convolution. If polynomial products are used, then the operation is polynomial division  $H_2(x) = H(x)/(H_1(x)H_1(x))$ .

First, compute  $h_{11}[n] = h_1[n] \otimes h_1[n]$ , with  $h_1[n] = \delta[n] + \delta[n - 1]$ , or  $h_1[0] = 1$ ,  $h_1[1] = 1$ ,

$$\begin{aligned} h_{11}[n] &= h_1[n] \otimes h_1[n] \\ &= \sum_{k=-\infty}^{+\infty} h_1[k]h_1[n - k] \\ &= \delta[n] + 2\delta[n - 1] + \delta[n - 2] \end{aligned}$$

Second, compute the length (here  $L\{\cdot\}$ ) of  $h_2[n]$ . While  $L\{h_1[n]\} = 5$ ,  $L\{h_{11}[n]\} = 3$ , and  $L\{h[n]\} = L\{h_{11}[n]\} + L\{h_2[n]\} - 1$ , the result is  $L\{h_2[n]\} = 3$ .

The index of the first non-zero element (here  $A\{\cdot\}$ ) is  $A\{h_2[n]\} = A\{h[n]\} - A\{h_{11}[n]\} = 0 - 0 = 0$ . Therefore the unknown sequence can be written as

$$h_2[n] = a\delta[n] + b\delta[n - 1] + c\delta[n - 2]$$

Third, compute the convolution, and solve the unknown constants  $a$ ,  $b$ , and  $c$ .

$$\begin{aligned} h[n] &= h_{11}[n] \otimes h_2[n] \\ &= \sum_{k=-\infty}^{+\infty} h_{11}[k]h_2[n - k] = \sum_{k=0}^2 h_{11}[k]h_2[n - k] \\ &= h_{11}[0] \cdot h_2[n - 0] + h_{11}[1] \cdot h_2[n - 1] + h_{11}[2] \cdot h_2[n - 2] \\ &= h_2[n - 0] + 2h_2[n - 1] + h_2[n - 2] \\ &= (a\delta[n] + b\delta[n - 1] + c\delta[n - 2]) \\ &\quad + 2(a\delta[n - 1] + b\delta[n - 2] + c\delta[n - 3]) \\ &\quad + (a\delta[n - 2] + b\delta[n - 3] + c\delta[n - 4]) \\ &= a\delta[n] + (b + 2a)\delta[n - 1] + (c + 2b + a)\delta[n - 2] + (2c + b)\delta[n - 3] + c\delta[n - 4] \\ &= \delta[n] + 5\delta[n - 1] + 9\delta[n - 2] + 7\delta[n - 3] + 2\delta[n - 4] \quad | \quad h_2[n] \text{ is known} \end{aligned}$$

The comparison between the last two lines from left gives  $a = 1$ , then  $(b + 2 \cdot 1) = 5 \Rightarrow b = 3$ , then  $(c + 2 \cdot 3 + 1) = 9 \Rightarrow c = 2$ , and also the rest values hold. In the end, the result is

$$h_2[n] = \delta[n] + 3\delta[n - 1] + 2\delta[n - 2]$$

which can be ensured by convolution.

**Remark.** The last convolution can be written easily in a visual way using “tabular method” (*Mitra 2Ed Sec. -, p. - / 3Ed Sec. 2.5.2, p. 83-85*).

Recall that  $h[n] = \sum_{k=0}^2 h_{11}[k]h_2[n - k]$ , where  $h_{11}[n] = \{\underline{1}, 2, 1\}$ ,  $h_2[n] = \{a, b, c\}$ , and  $h[n] = \{\underline{1}, 5, 9, 7, 2\}$ . Write down the sum into a table

n:	-1	0	1	2	3	4	5
$h_{11}[0]$ ·	0	a	b	c	0	0	0
$h_{11}[1]$ ·	0	0	a	b	c	0	0
+ $h_{11}[2]$ ·	0	0	0	a	b	c	0
= $h[n]$ =	$h[-1]$	$h[0]$	$h[1]$	$h[2]$	$h[3]$	$h[4]$	$h[5]$

Now, inserting all known values, and multiplying  $h_{11}[k]$  scaling factors in, we get

n:	-1	0	1	2	3	4	5
	0	a	b	c	0	0	0
	0	0	2a	2b	2c	0	0
+	0	0	0	a	b	c	0
= $h[n]$ =	0	1	5	9	7	2	0

We can start either from left-most non-zero column (or from right-most) to solve unknown values. First, we find out that in the column at  $n = 0$  there is  $a + 0 + 0 = 1$ , from which we get  $a = 1$ :

n:	-1	0	1	2	3	4	5
	0	1	b	c	0	0	0
	0	0	2	2b	2c	0	0
+	0	0	0	1	b	c	0
= $h[n]$ =	0	1	5	9	7	2	0

The next column at  $n = 1$  gives  $b + 2 + 0 = 5$ , from which  $b = 3$ :

n:	-1	0	1	2	3	4	5
	0	1	3	c	0	0	0
	0	0	2	6	2c	0	0
+	0	0	0	1	3	c	0
= $h[n]$ =	0	1	5	9	7	2	0

The next column at  $n = 2$  gives  $c + 6 + 1 = 9$ , from which  $c = 2$ :

$n:$	-1	0	1	2	3	4	5
$+$	0	1	3	2	0	0	0
$= h[n]$	0	1	5	9	7	2	0

Because everything holds, the deconvolution result is

$$h_2[n] = \delta[n] + 3\delta[n - 1] + 2\delta[n - 2]$$

33. **Problem:** LTI systems are commutative, distributive and associative. Determine the expression for the impulse response of each of the LTI systems shown in Figure 45.

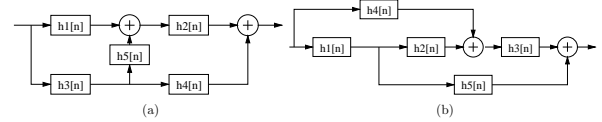


Figure 45: LTI systems in Problem 33.

[L0242] **Solution:** All subsystems are LTI. Therefore we can use sum of impulse responses for parallel systems and convolution of impulse responses for cascade systems (Mitra 2Ed Ex. 2.27 / 3Ed Ex. 2.35).

If any temporary variables are needed, they are probably best situated right after the summing units.

a) We can derive the impulse response  $h[n]$  of the whole system directly, or using a temporary variable  $v[n]$  (easier!?) shown in Figure ???. The useful position for  $v[n]$  is after summation.

$$\begin{aligned} v[n] &= (h_1[n] \otimes x[n]) + ((h_3[n] \otimes h_5[n]) \otimes x[n]) \\ y[n] &= (h_2[n] \otimes v[n]) + ((h_4[n] \otimes h_4[n]) \otimes x[n]) \\ &= ((h_2[n] \otimes h_1[n]) + (h_2[n] \otimes h_3[n] \otimes h_5[n]) + (h_3[n] \otimes h_4[n])) \otimes x[n] \\ h[n] &= (h_2[n] \otimes h_1[n]) + (h_2[n] \otimes h_3[n] \otimes h_5[n]) + (h_3[n] \otimes h_4[n]) \end{aligned}$$

b) In the same way as in (a).

$$\begin{aligned} v[n] &= (h_4[n] \otimes x[n]) + ((h_1[n] \otimes h_2[n]) \otimes x[n]) \\ y[n] &= (h_3[n] \otimes v[n]) + ((h_1[n] \otimes h_5[n]) \otimes x[n]) \\ &= ((h_3[n] \otimes h_4[n]) + (h_1[n] \otimes h_2[n] \otimes h_3[n]) + (h_1[n] \otimes h_5[n])) \otimes x[n] \\ h[n] &= (h_3[n] \otimes h_4[n]) + (h_1[n] \otimes h_2[n] \otimes h_3[n]) + (h_1[n] \otimes h_5[n]) \end{aligned}$$

34. **Problem:** The impulse response of a digital matched filter,  $h[n]$ , is the time-reversed replica of the signal to be detected (Mitra 2Ed Sec. 4.14, p. 272 / 3Ed Sec. -, p. -). The time-shift is needed in order to get a causal filter.

The (binary) signal to be detected is given by  $s[n] = \{1, 1, 1, -1, -1, 1, -1\}$ . Consider an input sequence  $x[n]$  which is a periodic sequence repeating  $s[n]$ . Determine  $h[n]$  and the result of filtering  $y[n] = h[n] \otimes x[n]$ .

[L0247] **Solution:** Matched filter. Let  $s[n]$  be a (binary) 7-bit long codeword to be detected,  $x[n]$  an input signal of repeated  $s[n]$ , and the impulse response of the matched filter  $h[n] = s[-n]$ :

$$\begin{aligned} s[n] &= \{1, 1, 1, -1, -1, 1, -1\} \\ x[n] &= \{\dots, s[n], s[n], s[n], \dots\} = \\ &= \{\dots, 1, 1, 1, -1, -1, 1, -1, 1, 1, 1, -1, -1, 1, -1, 1, 1, 1, -1, -1, 1, -1, \dots\} \\ h[n] &= s[-n] = \{-1, 1, -1, -1, 1, 1, 1\} \end{aligned}$$

The convolution result  $y[n] = h[n] \otimes x[n]$  is shown in Figure 46.

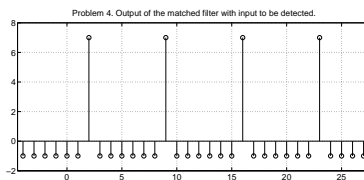


Figure 46: Convolution result of the matched filter and desired sequence in Problem 34.

The signal  $s[n]$  was chosen so, that the every seventh sample (length of  $s[n]$ ) in output is high, and all others are low. If the signal  $s[n]$  were different, there would be not so clear peaks in the convolution result.

**Remark.** Convolution and cross-correlation have a close connection (Mitra 2Ed Eq. 2.106, p. 89 / 3Ed Eq. 2.127, p. 101)

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} y[n]x[-(l-n)] = y[l] \otimes x[-l]$$

35. **Problem:** Determine the autocorrelation sequence of the sequence  $x_1[n] = \alpha^n \mu[n]$ ,  $|\alpha| < 1$  and show that it is an even sequence. What is the location of the maximum value of the autocorrelation sequence?

[L0249] **Solution:** Cross-correlation sequence  $r_{xy}[l]$  of two sequences and autocorrelation sequence  $r_{xx}[l]$  with lag  $l = 0, \pm 1, \pm 2, \dots$  are defined in (Mitra 2Ed Sec. 2.7 / 3Ed Sec. 2.9)

$$\begin{aligned} r_{xy}[l] &= \sum_{n=-\infty}^{\infty} x[n]y[n-l] & r_{xx}[l] &= \sum_{n=-\infty}^{\infty} x[n]x[n-l] \\ r_{xx}[l] &= \sum_{n=-\infty}^{\infty} x_1[n]x_1[n-l] \\ &= \sum_{n=-\infty}^{\infty} \alpha^n \mu[n] \alpha^{n-l} \mu[n-l] \\ &= \sum_{n=0}^{\infty} \alpha^{2n-l} \mu[n-l] \\ &= \begin{cases} \sum_{n=0}^{\infty} \alpha^{2n-l} = \frac{\alpha^{-l}}{1-\alpha^2}, & \text{for } l < 0 \\ \sum_{n=l}^{\infty} \alpha^{2n-l} = \frac{\alpha^{-l}}{1-\alpha^2} = \frac{\alpha^l}{1-\alpha^2}, & \text{for } l \geq 0 \end{cases} \end{aligned}$$

Note for the lag  $l \geq 0$ ,  $r_{xx}[l] = \frac{\alpha^l}{1-\alpha^2}$ , and for  $l < 0$ ,  $r_{xx}[l] = \frac{\alpha^{-l}}{1-\alpha^2}$ .

Replacing  $l$  with  $-l$  in the second expression we get  $r_{xx}[-l] = \frac{\alpha^{-(-l)}}{1-\alpha^2} = r_{xx}[l]$ .

Hence,  $r_{xx}[l]$  is an even function of  $l$ .

Maximum value of  $r_{xx}[l]$  occurs at  $l = 0$  since  $\alpha^l$  is a decaying function for increasing  $l$  when  $|\alpha| < 1$ .

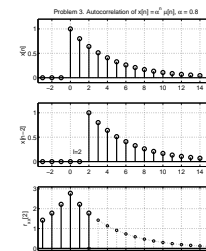


Figure 47: Autocorrelation sequence in Problem 35. Top:  $x[n]$ , middle:  $x[n - 2]$ , bottom:  $r_{xx}[l], r_{xx}[2] = \sum_k x[k]x[k - 2]$ .



36. **Problem:** Compute continuous-time Fourier transform (CTFT) of the following analog signals using the definition

$$X(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt$$

- a)  $x_1(t) = e^{-3t}\mu(t)$
- b)  $x_2(t) = e^{-j3t}$
- c)  $x_3(t) = e^{-j3t} + e^{j3t}$

[L0275] **Solution:** The continuous-time Fourier transform (CTFT) of a continuous-time signal  $x_a(t)$  is given by (Mitra 2Ed Eq. -, p. - / 3Ed Eq. 3.1, p. 118) below. The variable is the angular frequency  $\Omega = 2\pi f \in \mathbb{R}$ , in range  $-\infty < \Omega < \infty$ .

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\Omega t} dt$$

a) Now  $x_1(t) = e^{-3t}\mu(t) \in \mathbb{R}$ , see Figure 48(a). Note that the unit step function  $\mu(t)$  sets the low border of integration to zero. See also (Mitra 2Ed Ex. -, p. - / 3Ed Ex. 3.1, p. 118).

$$\begin{aligned} X_1(j\Omega) &= \int_{-\infty}^{\infty} x_1(t) e^{-j\Omega t} dt = \int_0^{\infty} e^{-3t} e^{-j\Omega t} dt = \int_0^{\infty} e^{-(3+j\Omega)t} dt \\ &= -\frac{1}{3+j\Omega} \int_0^{\infty} e^{-(3+j\Omega)t} dt = -\frac{1}{3+j\Omega} \cdot (0-1) \\ &= \frac{1}{3+j\Omega} \end{aligned}$$

b) Now  $x_2(t) = e^{-j3t} \in \mathbb{C}$ . The signal is complex-valued and runs clock-wise around unit circle with angular frequency  $\Omega = -3$  (rad/s), see Figure 48(b).

$$\begin{aligned} X_2(j\Omega) &= \int_{-\infty}^{\infty} x_2(t) e^{-j\Omega t} dt = \int_{-\infty}^{\infty} e^{-j3t} e^{-j\Omega t} dt = \int_{-\infty}^{\infty} e^{-j(3+\Omega)t} dt \\ &= 2\pi\delta(\Omega+3) \end{aligned}$$

There is a peak of height  $2\pi$  at  $\Omega = -3$ , because  $\int \delta(t)a(t)dt = a(t)|_{t=0}$ . The signal is complex and therefore the spectrum is not symmetric around y-axis.

c) Now  $x_3(t) = e^{-j3t} + e^{j3t} = 2\cos(3t) \in \mathbb{R}$  using Euler's formula. There are two peaks of height  $2\pi$  at frequencies  $\Omega = \pm 3$ .

$$X_3(j\Omega) = 2\pi(\delta(\Omega-3) + \delta(\Omega+3))$$

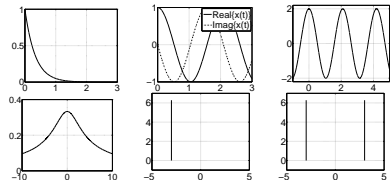


Figure 48: Problem 36: Top row (a), (b), (c): signals  $x_1(t)$ ,  $x_2(t)$ , and  $x_3(t)$ , time  $t$  in x-axis. Bottom row (d), (e), (f): corresponding CTFTs, amplitude spectra  $|X_1(j\Omega)|$ ,  $|X_2(j\Omega)|$ ,  $|X_3(j\Omega)|$ , angular frequency  $\Omega$  in x-axis. In case of real signals  $x(t) \in \mathbb{R}$  the spectrum is symmetric around y-axis.

37. **Problem:** Sketch the following signals in time-domain and their (amplitude) spectra in frequency-domain.

- a)  $x_1(t) = \cos(2\pi 500 t)$
- b)  $x_2(t) = 4\cos(2\pi 200 t) + 2\sin(2\pi 300 t)$
- c)  $x_3(t) = e^{-j(2\pi 250)t} + e^{j(2\pi 250)t}$
- d)  $x_4(t) = x_1(t) + x_2(t) + x_3(t)$

[L0271] **Solution:** The continuous-time Fourier transform (CTFT) decomposes the signal to its frequency components. Cosine and exponential function have a close relationship via Euler's formula:

$$\cos(\Omega t) = 0.5 \cdot (e^{j\Omega t} + e^{-j\Omega t})$$

Ideally, each real cosine component  $x_i(t) = A_i \cos(2\pi f_i t + \theta_i)$  is a peak at frequency  $f_i$  in an one-sided spectrum or a peak pair at frequencies  $-f_i$  and  $f_i$  in a two-sided spectrum. So, if the signal  $x(t)$  is real-valued, then the two-sided spectrum  $|X(j\Omega)|$  is symmetric. The amplitude  $A_i$  expresses how strong the cosine component is.

- a) A pure cosine at 500 Hz. Figure 49(a).
- b) A sum of two cosines. Peaks at 200 and 300 Hz. Figure 49(b).
- c) Two complex exponentials with the same amplitude and opposite frequencies can be combined to a cosine using Euler's formula. A peak at 250 Hz. Figure 49(c).
- d) The sum signal contains all components in time domain as well as in frequency domain. Figure 49(d).

**Remark.** Typically, when computing spectra numerically ( $x[n]$  instead of  $x(t)$ ) with computer, the peaks "spread". There is the of signal  $x_4[n] \leftarrow x_4(t)$  in Figure 50. DFT using  $N=40$  points in (a), and DFT using  $N=41$  points in (b), and both having the sampling frequency  $f_s = 2000$  Hz. So, in (a) the resolution  $f_0$  of the frequency is exactly 50 Hz, whereas in (b) it is  $2000 \text{ Hz} / 41 = 48.78$  Hz. The components of the signal are multiples of 50 Hz ( $4 \cdot 50 = 200$ , etc.) but not multiples of 48.78 Hz. In practice, the former case is very rare – normally all possible peaks are spread. This example was executed using the command `fft` in Matlab.

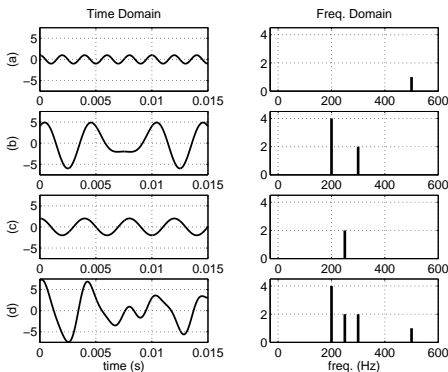


Figure 49: Problem 37: Signals and their one-sided spectra (CTFT).

When analyzing spectra in any commercial software, the sequence is first "cut" with a window (Hamming, Hanning, Blackman, etc.). Windows and their effect on spectra are discussed later in FIR filter design.

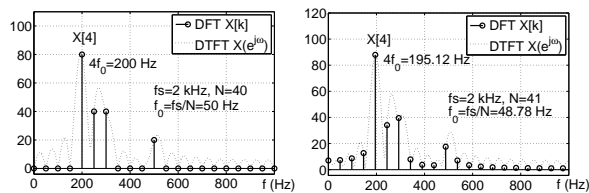


Figure 50: Remark in Problem 37. Discrete Fourier Transform (DFT) of the same signal (now discrete-time  $x_4[n]$ ) as in Figure 49(d): (a) signal components (200, 250, 300, 500 Hz) are multiples of the frequency resolution  $f_0 = 2000 \text{ Hz} / 40$ , (b) signal components are not any more multiples of  $f_0 = 2000 \text{ Hz} / 41$ . Actually there are only four frequency components in the signal, but this cannot be observed in (b). Fourier component  $X_4[4]$  is highlighted in both figures. In (a) its frequency is  $4f_0 = 200$  Hz, while in (b) it is  $4f_0 \approx 195$  Hz. Dashed line is the result of discrete-time Fourier transform (DTFT) where the frequency is continuous-valued.

38. **Problem:** Compute discrete-time Fourier transform (DTFT) for each of the following sequences using the definition

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

- a)  $x_1[n] = \delta[n-2]$
- b)  $x_2[n] = 0.5^n \mu[n]$
- c)  $x_3[n] = a[n] \cdot \cos(\frac{\pi}{5}n)$ , where  $a[n]$  is a real-valued sequence whose DTFT is  $A(e^{j\omega})$ .

[L0258] **Solution:** The discrete-time Fourier transform (DTFT) of sequence  $x[n]$  is defined

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

a)  $x_1[n] = \delta[n-2]$

$$\begin{aligned} X_1(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x_1[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \delta[n-2]e^{-j\omega n} \\ &= e^{-j2\omega} \end{aligned}$$

b)  $x_2[n] = 0.5^n \mu[n]$

$$\begin{aligned} X_2(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x_2[n]e^{-j\omega n} = \sum_{n=0}^{\infty} 0.5^n \mu[n]e^{-j\omega n} \\ &= \sum_{n=0}^{\infty} (0.5 \cdot e^{-j\omega})^n \\ &= \frac{1}{1 - 0.5 \cdot e^{-j\omega}} \end{aligned}$$

c)  $x_3[n] = a[n] \cos(\frac{\pi}{5}n)$ . Using Euler's formula we can express

$$\cos(\frac{\pi}{5}n) = 0.5 \cdot (e^{j\frac{\pi}{5}n} + e^{-j\frac{\pi}{5}n})$$

Hence,

$$\begin{aligned} X_3(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x_3[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} a[n] \cos(\frac{\pi}{5}n)e^{-j\omega n} \\ &= 0.5 \sum_{n=-\infty}^{\infty} a[n] (e^{j\frac{\pi}{5}n} + e^{-j\frac{\pi}{5}n})e^{-j\omega n} \\ &= 0.5 \sum_{n=-\infty}^{\infty} a[n] (e^{-j(\omega-\frac{\pi}{5})n} + e^{-j(\omega+\frac{\pi}{5})n}) \\ &= 0.5(A(e^{j(\omega-\frac{\pi}{5})}) + A(e^{j(\omega+\frac{\pi}{5})})) \end{aligned}$$

where  $A(e^{j\omega})$  is DTFT of  $a[n]$ . Signal  $a[n]$  is modulated with  $\omega = \pi/5$ . In the frequency domain the spectrum  $A(e^{j\omega})$  is "copied" (and scaled) at normalized angular frequencies  $\omega = \pm\pi/5$ . See Problem 39(a) for solving the same problem with the transform table.

39. **Problem:** Consult the transform table and find the DTFTs of sequences

- a)  $x_3[n] = a[n] \cdot \cos(0.2\pi n)$
- b)  $x_4[n] = 0.25^{n-1} \mu[n-1]$
- c)

$$x_5[n] = \begin{cases} 0, & n < -1 \vee n \geq 6 \\ 2, & -1 \leq n < 1 \\ 3, & 1 \leq n < 4 \\ 1, & 4 \leq n < 6 \end{cases}$$

[L0282] **Solution:** The discrete-time Fourier transform (DTFT) is defined as

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

but can often be solved via the transform table. There are a set of transform pairs and some properties listed in the table.

- a) There are two following lines in the transform table related to this problem. The first one is for *modulation* (product in time domain) and the latter Fourier-transform of a cosine sequence:

$$\begin{aligned} x_1[n] \cdot x_2[n] &\leftrightarrow \frac{1}{2\pi} \int_{2\pi} X_1(e^{j\theta}) \cdot X_2(e^{j(\omega-\theta)}) d\theta \\ \cos(\omega_0 n) &\leftrightarrow \pi \sum_l (\delta(\omega - \omega_0 + 2\pi l) + \delta(\omega + \omega_0 + 2\pi l)) \end{aligned}$$

Now an unknown sequence  $a[n]$ , whose DTFT is  $A(e^{j\omega})$ , is modulated with a cosine sequence with  $\omega_0 = 0.2\pi$ . Transform of the cosine is

$$x_1[n] = \cos((\pi/5)n) \leftrightarrow \pi \sum_l (\delta(\omega - \pi/5 + 2\pi l) + \delta(\omega + \pi/5 + 2\pi l)) = X_1(e^{j\omega})$$

which is an (infinite-length) impulse train. Convolution a spectrum  $X_2(e^{j\omega}) = A(e^{j\omega})$  with the impulse train  $X_1(e^{j\omega})$  over one period, that is  $l = 0$ , and multiplying with  $\frac{1}{2\pi}$  we get

$$\begin{aligned} X_3(e^{j\omega}) &= \frac{1}{2\pi} \int_{2\pi} X_1(e^{j\theta}) \cdot X_2(e^{j(\omega-\theta)}) d\theta \\ &= \frac{1}{2\pi} \cdot \pi \cdot \left( \int_{2\pi} \delta(\theta - \pi/5) \cdot A(e^{j(\omega-\theta)}) d\theta + \int_{2\pi} \delta(\theta + \pi/5) \cdot A(e^{j(\omega-\theta)}) d\theta \right) \\ &= \frac{1}{2} \cdot (A(e^{j(\omega-\pi/5)}) + A(e^{j(\omega+\pi/5)})) \end{aligned}$$

where we have utilized *Dirac's delta* function, see Formulas  $\int_{-\infty}^{\infty} \delta(t - t_0)x(t) dt = x(t_0)$ .

An example can be seen in Figure 51. The time-domain modulation corresponds frequency-domain convolution.<sup>1</sup> Hence, the original spectrum is "copied" at each  $\delta(\cdot)$  frequency. In this case cosine  $\cos((\pi/5)n) = 0.5(e^{j(\pi/5)n} + 0.5(e^{j(-\pi/5)n})$ , from which the normalized angular frequencies are  $-\pi/5$  and  $\pi/5$ .

<sup>1</sup>Java applet for convolution of spectra <http://www.jhu.edu/~signals/convolve/index.html>.

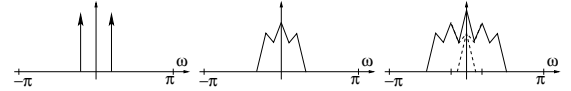


Figure 51: Problem 39(a): (a)  $X_1(e^{j\omega})$  (Dirac's) impulse train from a cosine, depicted only in range  $(-\pi \dots \pi]$ , (b)  $X_2(e^{j\omega}) = A(e^{j\omega})$  spectrum from an arbitrary sequence  $a[n]$ , (c)  $X_3(e^{j\omega})$  convolution of spectra  $X_1$  and  $X_2$ , which corresponds modulation  $x_1[n] \cdot x_2[n]$  in time-domain. The spectrum  $X_2$  is "copied" at  $-\pi/5$  and  $\pi/5$ , and overlapping parts are summed together (assuming zero-phases signals). All scaling factors (y-axis) have been omitted in these graphs. Discrete-time spectra  $X_1, X_2$ , and  $X_3$  are all  $2\pi$ -periodic.

- b) From table we see that  $a^n \mu[n] \leftrightarrow 1/(1 - ae^{-j\omega})$  and  $x[n - k] \leftrightarrow e^{-jk\omega} X(e^{j\omega})$ . In this case we have  $a = 0.25$  and there is a time shift by  $k = 1$ , from which we get

$$X_4(e^{j\omega}) = e^{-j\omega} \cdot 1/(1 - 0.25e^{-j\omega})$$

- c) The sequence  $x_5[n]$  can be converted directly with the pair  $\delta[n - k] \leftrightarrow e^{-jk\omega}$  and keeping in mind that transform is linear  $\sum_i c_i \cdot \delta[n - k_i] \leftrightarrow \sum_i c_i \cdot e^{-jk_i\omega}$ . In this way

$$X_5(e^{j\omega}) = 2e^{j\omega} + 2 + 3e^{-j\omega} + 3e^{-2j\omega} + 3e^{-3j\omega} + e^{-4j\omega} + e^{-5j\omega}$$

However, this time we want to practice transform tables. Note that  $x_5[n]$  can be constructed as a sum from two "boxes"  $x_5[n] = x_{51}[n] + x_{52}[n]$ , see Figure 52(b)-(d)

$$\begin{aligned} x_{51}[n] &= 2, \quad -1 \leq n < 4 \\ x_{52}[n] &= 1, \quad 1 \leq n < 6 \end{aligned}$$

we can utilize the time shifting property and a transform of a rectangle

$$\begin{aligned} x[n - k] &\leftrightarrow e^{-jk\omega} X(e^{j\omega}) \\ x_{rect}[n] &= \begin{cases} 1, & |n| \leq N_1 \\ 0, & |n| > N_1 \end{cases} \leftrightarrow \frac{\sin(\omega(N_1 + 0.5))}{\sin(\omega/2)} = X_{rect}(e^{j\omega})|_{N_1} \end{aligned}$$

Now we can see that  $x_{51}[n] = 2x_{rect}[n - 1]$  with length parameter  $N_1 = 2$  and delay  $k = 1$ , as shown in Figure 52(a)-(b). For the other rectangle  $x_{52}[n] = x_{rect}[n - 3]$  we have  $N_1 = 2$  and  $k = 3$ . Next, because of linearity, we can transform  $x_{51}$  and  $x_{52}$  separately and add the spectra together. Hence,

$$\begin{aligned} X_5(e^{j\omega}) &= X_{51}(e^{j\omega}) + X_{52}(e^{j\omega}) \\ &= 2e^{-j\omega} X_{rect}(e^{j\omega})|_{N_1=2} + e^{-3j\omega} X_{rect}(e^{j\omega})|_{N_1=2} \\ &= (2e^{-j\omega} + e^{-3j\omega}) \left( \frac{\sin(2.5\omega)}{\sin(0.5\omega)} \right) \end{aligned}$$

When computing at  $\omega = 0$ , we get  $X_5(e^{j0}) = 15$ , because  $\sin(2.5\omega)/\sin(0.5\omega) \rightarrow 5$ , when  $\omega \rightarrow 0$  using *Hospital's rule*: If  $\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{0}{0}$ , then  $\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)}$ . Now  $\lim_{\omega \rightarrow 0} \frac{f'(\omega)}{g'(\omega)} = \lim_{\omega \rightarrow 0} \frac{2.5 \cos(2.5\omega)}{0.5 \cos(0.5\omega)} = 5$ .

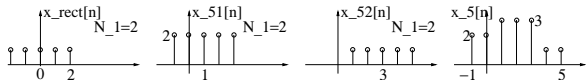


Figure 52: Problem 39(b): (a)  $x_{rect}[n]$ , a rectangular sequence, whose transform  $X_{rect}|_{N_1}$  is known, (b)  $x_{51}[n] = 2x_{rect}[n - 1]|_{N_1=2}$ , which is a rectangle of height 2, of length 2, i.e.,  $N_1 = 2$ , and delayed by  $k = 1$  from origo, (c)  $x_{52}[n] = x_{rect}[n - 3]|_{N_1=2}$ , height 1, length 2, delay 3, and (d)  $x_5[n] = x_{51}[n] + x_{52}[n]$ .

40. **Problem:** Suppose that a real sequence  $x[n]$  and its discrete-time Fourier transform (DTFT)  $X(e^{j\omega})$  are known. The sampling frequency is  $f_s$ . At normalized angular frequency  $\omega_c = \pi/4$ :  $X(e^{j(\pi/4)}) = 3 + 4j$ . Determine

- a)  $|X(e^{j(\pi/4)})|$
- b)  $\angle X(e^{j(\pi/4)})$
- c)  $X(e^{j(-\pi/4)})$
- d)  $X(e^{j(\pi/4+2\pi)})$
- e) If  $f_s = 4000$  Hz, what is  $f_c$

[L0257] **Solution:** The *discrete-time Fourier transform (DTFT)* is always  $2\pi$ -periodic:

$$X(e^{j(\omega+2\pi k)}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j(\omega+2\pi k)n} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \underbrace{e^{-j2\pi kn}}_{=1} = X(e^{j\omega})$$

Complex-valued DTFT can be considered in polar coordinates

$$\begin{aligned} X(e^{j\omega}) &= |X(e^{j\omega})| \cdot e^{j\angle X(e^{j\omega})} \\ z &= r \cdot e^{j\theta} \end{aligned}$$

where  $|X(e^{j\omega})|$  is (amplitude) spectrum and  $\angle X(e^{j\omega})$  phase spectrum.

The value of DTFT was given at  $\omega_c = \pi/4$ :  $X(e^{j(\pi/4)}) = 3 + 4j$ .

- a)  $|X(e^{j(\pi/4)})| = 5$
- b)  $\angle X(e^{j(\pi/4)}) = \arctan(4/3) \approx 0.927$
- c)  $X(e^{j(-\pi/4)}) = 3 - 4j$
- d)  $X(e^{j(\pi/4+2\pi)}) = 3 + 4j$
- e) Normalized angular sampling frequency is  $\omega_c = 2\pi$ . The interesting frequency can be obtained from the ratio  $(\omega_c/\omega_s) = (f_c/f_s)$ . If the sampling frequency  $f_s = 4000$  Hz, then

$$f_c = \frac{4000 \text{ Hz} \cdot (\pi/4)}{2\pi} = 500 \text{ Hz.}$$

41. **Problem:** The magnitude response function  $|X(e^{j\omega})|$  of a discrete-time sequence  $x[n]$  is shown in Figure 53 in normalized angular frequency axis. Sketch the magnitude response for the range  $-\pi \leq \omega < \pi$ . Is the signal  $x[n]$  real or complex valued?

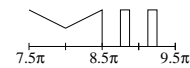


Figure 53:  $|X(e^{j\omega})|$  of Problem 41.

[L0276] **Solution:** Discrete-time Fourier transform (DTFT) is always  $2\pi$ -periodic:

$$X(e^{j(\omega+2\pi k)}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j(\omega+2\pi k)n} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \underbrace{e^{-j2\pi kn}}_{=1} = X(e^{j\omega})$$

The spectrum in range  $(7.5\pi \dots 9.5\pi]$  can be repeated. Borders correspond  $7.5\pi - (2 \cdot 4)\pi = -0.5\pi$  and  $9.5\pi - (2 \cdot 4)\pi = 1.5\pi$ . When origo taken as a central point, it can be seen that the spectrum is symmetric around y-axis. See Figure 54.

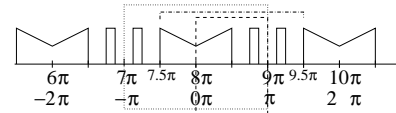


Figure 54: Problem 41: Discrete-time spectrum is periodic with  $2\pi$ . The bottom label line is shifted by  $8\pi$ . The two-sided spectrum in range  $(-\pi \dots \pi]$  in a dashed rectangle.

In case of a real valued sequence  $x[n]$  the following symmetry relations hold (*Mitra 2Ed Sec. 3.1.4, p. 127 / 3Ed Sec. 3.2.3, p. 128*):

$$\begin{aligned} X(e^{j\omega}) &= X^*(e^{-j\omega}) \\ X_{re}(e^{j\omega}) &= X_{re}(e^{-j\omega}) \\ X_{im}(e^{j\omega}) &= -X_{im}(e^{-j\omega}) \\ |X(e^{j\omega})| &= |X(e^{-j\omega})| \\ \angle X(e^{j\omega}) &= -\angle X(e^{-j\omega}) \end{aligned}$$

Equivalently, because now our magnitude spectrum is symmetric ( $|X(e^{j\omega})| = |X(e^{-j\omega})|$ ), then  $x[n] \in \mathbb{R}$ . For real sequences  $x[n]$  it is normal to draw the spectrum only in range  $\omega \in [0 \dots \pi]$ , as in Figure 55.

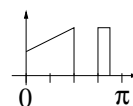


Figure 55: Problem 41: One-sided spectrum.

42. **Problem:** A LTI filter is characterized by its difference equation

$$y[n] = 0.25x[n] + 0.5x[n - 1] + 0.25x[n - 2]$$

- Draw the block diagram
- What is the impulse response  $h[n]$
- Determine the frequency response  $H(e^{j\omega}) = \sum_k \frac{d_k e^{-j\omega k}}{\sum_k d_k e^{-j\omega k}}$
- Determine the amplitude response  $|H(e^{j\omega})|$
- Determine the phase response  $\angle H(e^{j\omega})$
- Determine the group delay  $\tau(\omega) = -\frac{d\angle H(e^{j\omega})}{d\omega}$

[L0277] **Solution:** LTI system can be characterized by a linear constant coefficient difference equation of form (Mitra 2Ed Sec. 4.2.7, 4.2.6 / 3Ed Sec. 3.8.3, 3.9.1)

$$\sum_k d_k y[n - k] = \sum_k p_k x[n - k]$$

and its corresponding frequency response can be derived by Fourier transform ( $a \cdot x[n - k] \leftrightarrow a \cdot e^{-j\omega k} X(e^{j\omega})$ ) or directly

$$H(e^{j\omega}) = \frac{\sum_k p_k e^{-j\omega k}}{\sum_k d_k e^{-j\omega k}}$$

Finite-length sequences with certain geometric symmetry have an important role in digital signal processing (Mitra 2Ed Sec. 4.4.3 / 3Ed Sec. 5.5.2). A length- $N$  sequence is said to be symmetric if

$$x[n] = x[N - 1 - n]$$

and a length- $N$  sequence is said to be antisymmetric if

$$x[n] = -x[N - 1 - n]$$

For example,  $x_1[n] = \{-2, 3, -2\}$  is symmetric around  $n_c = 1$ , and  $x_2[n] = \{1, -2, 2, -1\}$  is antisymmetric around  $n_c = -0.5$ .

**Remark.** If the length of the impulse response  $h[n]$  is  $N + 1$ , then the order of filter is  $N$ . For instance, two-point moving average  $y[n] = 0.5 \cdot (x[n] + x[n - 1])$ , whose impulse response  $h[n] = 0.5 \cdot (\delta[n] + \delta[n - 1])$  is of length 2, and has the frequency response  $H(e^{j\omega}) = 0.5 \cdot (1 + e^{-j\omega})$ , which is 1st order filter according to the order of the polynomial of  $e^{j\omega}$ . There is a risk of confusing length and order, because  $N$  is typically used for both of them.

- The block (flow) diagram of the filter with the difference equation  $y[n] = 0.25x[n] + 0.5x[n - 1] + 0.25x[n - 2]$  is drawn in Figure 56(a). There is no feedback loops in the system, i.e., there is no delayed  $y[n - k]$  terms in the difference equation. Hence, the system is FIR.
- The impulse response  $h[n]$  is computed by assigning  $x[n] \leftarrow \delta[n]$ , which gives in the output  $y[n] = 0.25\delta[n] + 0.5\delta[n - 1] + 0.25\delta[n - 2]$ . Because the system response to an impulse is called the impulse response, we write

$$h[n] = 0.25\delta[n] + 0.5\delta[n - 1] + 0.25\delta[n - 2]$$

The length of the impulse response is  $L\{h[n]\} = 3 < \infty$ . So, the system is FIR (= Finite (length) Impulse Response). While  $h[n]$  is of finite length, the *stability condition*  $\sum_n |h[n]|$  always converges and therefore FIR filters are always stable.

- Recall, that any complex number  $z$  can be represented in the polar coordinate system as  $z = r \cdot e^{j\theta}$ , where  $r = |\sqrt{z}|$  and  $\theta = \angle z = \arctan \text{Imag}(z) / \text{Real}(z)$ .

The frequency response  $H(e^{j\omega})$  can be expressed by its amplitude response (= magnitude response)  $|H(e^{j\omega})|$  and the phase response  $\angle H(e^{j\omega})$  (Mitra 2Ed Sec. 4.2.1 / 3Ed Sec. 3.8.1)

$$H(e^{j\omega}) = |H(e^{j\omega})| \cdot e^{j\angle H(e^{j\omega})}$$

Now using the expression above,

$$H(e^{j\omega}) = \frac{0.25 + 0.5e^{-j\omega} + 0.25e^{-j2\omega}}{1}$$

In this particular case, when  $h[n]$  is symmetric around  $n_c = 1$ , we can take  $e^{-j\omega n_c}$  as a common factor, utilize Euler's formula, and write the frequency response as follows

$$\begin{aligned} H(e^{j\omega}) &= 0.25 + 0.5e^{-j\omega} + 0.25e^{-j2\omega} \\ &= e^{-j\omega} (0.25e^{j\omega} + 0.5 + 0.25e^{-j\omega}) \\ &= e^{-j\omega} (0.5 \cos(\omega) + 0.5) \end{aligned}$$

- Recall that absolute value of product  $A$  and  $B$  is the same as product of absolute values  $A$  and  $B$ :  $|A \cdot B| = |A| \cdot |B|$ .

When computing values for the amplitude response in range  $[0 \dots \pi]$  we will get the curve which says if the filter is lowpass / highpass / bandpass / bandstop

$$|H(e^{j\omega})| = \underbrace{|e^{-j\omega}|}_{=1} \cdot |(0.5 \cos(\omega) + 0.5)| = |0.5 \cos(\omega) + 0.5|$$

The amplitude response is plotted in Figure 56(b).

- Recall that angle of product  $A$  and  $B$  is the same as sum of angles of  $A$  and  $B$ :  $\angle(A \cdot B) = \angle A + \angle B$ .

Now, in this case when  $h[n]$  is finite-length and symmetric, the phase response can be expressed easily in closed form (Mitra 2Ed Sec. 4.4.3 / 3Ed Sec. 7.3)

$$\angle H(e^{j\omega}) = \angle e^{-j\omega} + \underbrace{\angle(0.5 \cos(\omega) + 0.5)}_{=0} = -\omega$$

The phase response is linear and plotted in Figure 56(c).

- In case of linear phase response, the group delay is constant for all frequencies (Mitra 2Ed Sec. 4.2.6 / 3Ed Sec. 3.9). That is, all frequency components are delayed by the same amount (of time). In this case the output sequence is delayed by 1 in the filter.

$$\tau(\omega) = -\frac{d}{d\omega} \angle H(e^{j\omega}) = 1$$

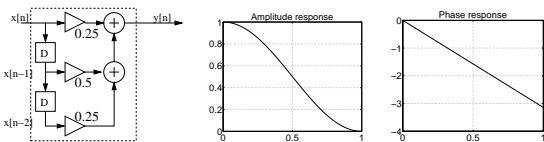


Figure 56: Problem 42: (a) block diagram, (b) amplitude response  $|H(e^{j\omega})|$ , (c) phase response  $\angle H(e^{j\omega})$ , which in this case is linear because of symmetric properties of finite-length  $h[n]$ .  $x$ -axis in (b) and (c)  $\omega/\pi$ , y-axis in (b) linear and in (c) radians.

43. **Problem:** Consider a continuous-time signal  $x(t)$  which consists of two cosine components:

$$x(t) = A_1 \cos(2\pi f_1 t + \theta_1) + A_2 \cos(2\pi f_2 t + \theta_2)$$

In Figure 57 there are the sinusoidal components in the first two rows and the sum signal in bottom. The scale in x-axis is about 0...0.1 seconds.

- Sketch the amplitude spectrum  $|X(j\Omega)|$  of the signal  $x(t)$ .
- What is the smallest sampling frequency, with which there is no aliasing in  $x(t)$ ?
- Create a sequence  $x[n]$  by taking samples with sampling frequency  $f_T = 100$  Hz.
- Sketch the waveform of the reconstructed signal  $x_r(t)$  in the time domain.

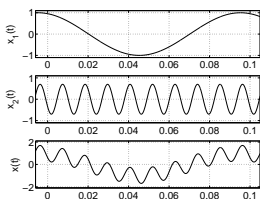


Figure 57: Problem 43: Sinusoidal components  $x_1(t)$  and  $x_2(t)$  and their sum  $x(t)$ .

[L0422] **Solution:** The signal  $x(t)$  consists of two sinusoids whose frequencies can be computed from Figure 57. We see that the period of  $x_1(t)$  is  $T_1 = 0.1$  s, and it oscillates 10 times a second, that is,  $f_1 = 10$  Hz. There are 9 oscillations of  $x_2(t)$  in 0.1 seconds, i.e.,  $T_2 = 0.0111 \dots$  s, and 90 periods a second,  $f_2 = 90$  Hz. Amplitudes can be read from the extreme points,  $A_1 \approx 1$  and  $A_2 \approx 0.8$ . The phase shifts  $\theta_i$  can also be approximated if needed.

We can compute the fundamental frequency  $f_0$  of the signal  $x(t)$  by finding greatest common divisor (GCD) of  $f_1 = 10$  and  $f_2 = 90$ . That is  $f_0 = 10$  Hz, because  $f_1 = 1 \cdot f_0$  and  $f_2 = 9 \cdot f_0$ , see Problem 20.

- The one-sided spectrum of a sinusoidal signal is a peak at corresponding frequency. Thus, the spectral components are a peak at 10 Hz with relative height of 1 and another peak at 90 Hz with height of 0.8. This is sketched in Figure 58.  $X(j\Omega)$  can be computed via *continuous-time Fourier transform (CTFT)*, where the spectrum  $X(j\Omega)$  is not periodic, see Problem 37.

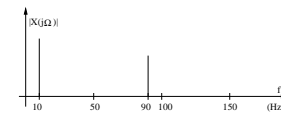


Figure 58: Problem 43: Spectrum  $|X(j\Omega)|$ .

- The smallest *sampling frequency* according to *Shannon sampling theorem* is at least two times higher than highest frequency in the bandlimited signal (Mitra 2Ed Sec. 5.2.1 / 3Ed Sec. 4.2.1). In this case,  $f_T \geq 2 \cdot 90$  Hz = 180 Hz.
- Analog signal  $x(t)$  is sampled with a sampling frequency of  $f_T = 100$  Hz. This means taking samples every  $T = 1/f = 0.01$  s. Therefore  $t$  is substituted by  $nT$

$$x[n] = x(nT) = x(n/f_T) = A_1 \cos(2\pi(f_1/f_T)n + \theta_1) + A_2 \cos(2\pi(f_2/f_T)n + \theta_2)$$

Two examples of sampling are given in Figure 59. Notice that the component at 10 Hz can be recovered whereas the component at  $90 > f_T/2$  Hz is aliased to a lower frequency.

- After sampling the shape of  $x[n]$  can be seen in bottom subfigures of Figure 59. It is a pure sinusoidal curve with 10 Hz. This is because the high-frequency 90 Hz component is aliased down to exactly 10 Hz,  $t \leftarrow n/100$ :

$$\cos(2\pi 90(n/100)) = \cos(1.8\pi n) = \cos(-0.2\pi n) = \cos(2\pi 10(n/100))$$

see more examples in Problem 47. The same can be found in frequency-domain either by copying the base band ( $0 \dots f_T/2$ ) spectrum at each multiple of  $f_T$  (Problem 46) or by flipping each component down to the base band (Problem 48).

The ideal reconstruction returns the base band (corresponding  $0 \dots \pi$ ) back to analog domain. In the reconstructed spectrum  $X_r(j\Omega)$  there is only one peak at 10 Hz, which corresponds one single sinusoid, see the dash curves in bottom subfigures of Figure 59.

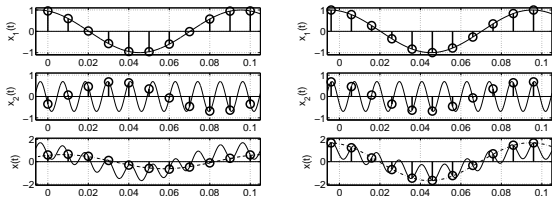


Figure 59: Problem 43: Sinusoidal components  $x_1(t)$ ,  $x_2(t)$ , their sum  $x(t)$ , and corresponding sampled sequences  $x_1[n]$ ,  $x_2[n]$ ,  $x[n]$ . The amplitude of  $x[n]$  depends on the phases  $\theta_i$ . The dash curve in bottom is reconstructed analog signal  $x_r(t)$ .

44. **Problem:** Show that the periodic impulse train  $p(t)$

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

can be expressed as a Fourier series

$$p(t) = \frac{1}{T} \sum_{k=-\infty}^{\infty} e^{j(2\pi/T)kt} = \frac{1}{T} \sum_{k=-\infty}^{\infty} e^{j\Omega_T kt}$$

where  $\Omega_T = 2\pi/T$  is the sampling angular frequency.

[L0400] **Solution:** Since  $p(t)$

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

is a periodic function of time  $t$  with a period  $T$  (time between samples), it can be represented as Fourier series (F-series for periodic, F-transform for non-periodic signals):

$$p(t) = \sum_{n=-\infty}^{\infty} c_n e^{j(2\pi n t/T)}$$

where Fourier coefficients (note,  $p(t)$  over one period  $T$ )

$$c_n = \frac{1}{T} \int_T p(t) e^{-j(2\pi n t/T)} dt$$

The unit impulse function (continuous-time) has properties

- (1)  $\int_{-\infty}^{\infty} \delta(t) dt = 1$ , and
- (2)  $\int_{-\infty}^{\infty} \delta(t) a(t) dt = a(t)|_{t=0}$ .

Therefore Fourier series coefficients are:

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) e^{-j(2\pi n t/T)} dt = \frac{1}{T} e^{-j(2\pi n t/T)}|_{t=0} = \frac{1}{T}$$

Hence

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j(2\pi n t/T)}$$

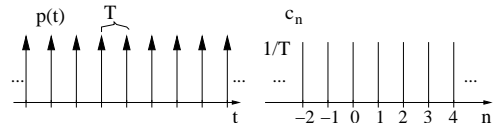


Figure 60: Problem 44: impulse train  $p(t)$  left, and its Fourier series coefficients  $c_n$  right.

45. **Problem:** Impulse train in Problem 44 can be also expressed as a Fourier transform

$$P(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s)$$

Sampling can be modeled as multiplication in time domain  $x[n] = x_p(t) = x(t)p(t)$ . What is  $X_p(j\Omega)$  for an arbitrary input spectrum  $X(j\Omega)$ ?

Hints: Fourier transform of a periodic signal (Fourier series)

$$X(j\Omega) = \sum_{n=-\infty}^{\infty} 2\pi a_n \delta(\Omega - k\Omega_0)$$

Multiplication of signals in time domain corresponds to convolution of transforms in frequency domain:

$$x_1(t) \cdot x_2(t) \leftrightarrow \frac{1}{2\pi} [X_1(j\Omega) \otimes X_2(j\Omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_1(j\theta) \cdot X_2(j(\Omega - \theta)) d\theta$$

[L0415] **Solution:** The Fourier series of a continuous-time signal can be expressed

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\Omega_0 t}$$

where  $a_k$  are Fourier coefficients and  $\Omega_0$  is fundamental angular frequency. Fourier transform of a periodic signal can be written in form of

$$X(j\Omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\Omega - k\Omega_0)$$

So, the impulse train  $p(t)$  of Problem 44 with all coefficients  $a_k = 1/T_s$  and fundamental angular frequency  $\Omega_s$  can be written as

$$P(j\Omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s)$$

Sampling in time and frequency domain can be modeled  $x[n] = x(t) \cdot p(t) \leftrightarrow \frac{1}{2\pi} [X(j\Omega) \otimes$

$P(j\Omega)]$ , which finally gives

$$\begin{aligned} \frac{1}{2\pi} [P(j\Omega) \otimes X(j\Omega)] &= \frac{1}{2\pi} \int_{-\infty}^{\infty} P(j\theta) X(j(\Omega - \theta)) d\theta \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\theta - k\Omega_s) X(j(\Omega - \theta)) d\theta \\ &= \frac{1}{T_s} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(\theta - k\Omega_s) X(j(\Omega - \theta)) d\theta \quad | \quad \int \delta(t) x(t) dt = x(t)|_{t=0} \\ &= \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s)) \end{aligned}$$

In other words, the spectrum  $X(e^{j\omega})$  of the discrete-time signal  $x[n]$  can be obtained by summing the shifted spectra  $X(j\Omega)$  of the corresponding analog signal  $x(t)$ . Spectra  $X(j\Omega)$  are scaled by  $(1/T_s)$  and copied at every sampling (angular) frequency.

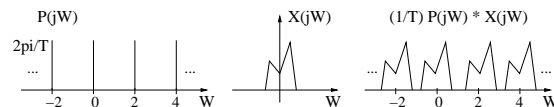


Figure 61: Problem 45: Left, an example of a spectrum  $P(j\Omega)$  of an impulse train, middle, a spectrum  $X(j\Omega)$  of an arbitrary signal, and their convolution in right. Notice that  $X(j\Omega)$  is not symmetric, which means that  $x(t)$  is complex-valued.

46. **Problem:** Suppose that a continuous-time signal  $x(t)$  and its spectrum  $|X(j\Omega)|$  in Figure 62 are known.

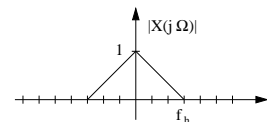


Figure 62: Spectrum  $X(j\Omega)$  in Problem 46.

The highest frequency component in the signal is  $f_h$ . The signal is sampled with frequency  $f_s$ , i.e. the interval between samples is  $T_s = 1/f_s$ ;  $x[n] = x(nT_s)$ . Sketch the spectrum  $|X(e^{j\omega})|$  of the discrete-time signal, when

- a)  $f_h = 0.25 f_s$
- b)  $f_h = 0.5 f_s$
- c)  $f_h = 0.75 f_s$

[L0416] **Solution:** The spectrum  $X(j\Omega)$  of a real analog signal is symmetric around  $\omega$ -axis. When sampling, the spectrum  $X(e^{j\omega})$  is  $2\pi$ -periodic (corresponds sampling frequency  $f_s$ )

$$x[n] = x_p(nT_s) = x(t)p(t) \leftrightarrow X(e^{j\omega}) = X_p(j\Omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s))$$

where  $\omega = 2\pi\Omega/\Omega_s = 2\pi f/f_s$ .

a) Figure 63. The highest component of  $x(t)$  is only  $0.25 \cdot f_s \Rightarrow$  No aliasing.

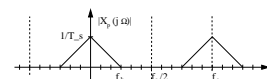


Figure 63:  $f_h = 0.25 f_s$ , no aliasing in Problem 46(a).

b) Figure 64. Case: Nyquist frequency, half of the sampling frequency.

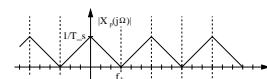


Figure 64:  $f_h = 0.5 f_s$ , critical sampling in Problem 46(b).

c) Figure 65. Aliasing takes place.  $X(e^{j\omega})$  is the sum of all folded analog spectra. The spectrum  $X(e^{j\omega})$  is depicted in Figure 65 with a thick continuous line.

$$X(e^{j\omega}) = \frac{1}{T_s} \left( \dots + X(j(\Omega - \Omega_s)) + X(j\Omega) + X(j(\Omega + \Omega_s)) + \dots \right)$$

47. **Problem:** Consider a continuous-time signal

$$\tilde{x}(t) = \begin{cases} \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t), & t \geq 0 \\ 0, & t < 0 \end{cases}$$

where  $f_1=100$  Hz,  $f_2=300$  Hz and  $f_3=750$  Hz. The signal is sampled using frequency  $f_s$ . Thus, a discrete signal  $x[n] = \tilde{x}(nT_s) = \tilde{x}(n/f_s)$  is obtained.

Sketch the magnitude of the Fourier spectrum of  $x[n]$ , the sampled signal, when  $f_s$  equals to (i) 1600 Hz (ii) 800 Hz (iii) 400 Hz.

Use an ideal reconstruction lowpass filter whose cutoff frequency is  $f_s/2$  for each case. What frequency components can be found in reconstructed analog signal  $x_r(t)$ ?

[L0417] **Solution:** There is a continuous-time signal

$$x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t)$$

Let  $f_1 = 100$  Hz,  $f_2 = 300$  Hz and  $f_3 = 750$  Hz.

It is possible directly to express the Fourier transform of a periodic signal using transform pairs (or see Page 78). In this case using frequencies  $f$  and a constant  $A$

$$X(f) = A \cdot (\delta[f + 750] + \delta[f + 300] + \delta[f + 100] + \delta[f - 100] + \delta[f - 300] + \delta[f - 750])$$

The signal is sampled with sampling frequency  $f_s$ , ( $T = 1/f_s$ ).

$$x[n] = x(nT) = x\left(\frac{n}{f_s}\right) = \left( \cos\left(2\pi \frac{f_1}{f_s} n\right) + \cos\left(2\pi \frac{f_2}{f_s} n\right) + \cos\left(2\pi \frac{f_3}{f_s} n\right) \right)$$

In the frequency domain the discrete-time spectrum  $G_p(j\Omega)$  can be seen as a sum of shifted and scaled replicas of the analog spectrum  $G_a(j\Omega)$  as shown in Problems 45 and 46 (Mitra 2Ed Eq. 5.9, p. 302 / 3Ed Eq. 4.10, p. 174):

$$G_p(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G_a(j(\Omega - k\Omega_T))$$

Alternatively, sampling can be considered as flipping the analog spectrum around each half of the sampling frequency down to the band  $0 \dots f_s/2$ .

Reconstruction means converting a digital sequence back to analog signal. An ideal lowpass filter with the passband up to half of the sampling frequency is used. When reconstructing signals we can only observe frequencies up to Nyquist frequency.<sup>2</sup> If there

<sup>2</sup>There is variation in using "Nyquist frequency" in the literature. It is either (1) half of the sampling frequency (Mitra 2Ed p. 302 / 3Ed p. 174) or (2) the highest frequency in the signal (Mitra 2Ed p. 304 / 3Ed p. 176). The first one is much more common. The reader should not confuse with this.

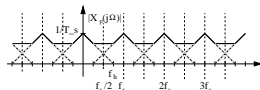


Figure 65:  $f_s = 0.75 f_3$ , aliasing in Problem 46(c).

are frequencies over the Nyquist frequency in the original signal, those frequencies are aliased into low frequencies.

In this problem  $X(j\Omega)$  is sampled with three different sampling frequencies  $f_s$  of 1600 Hz, 800 Hz and 400 Hz. The Nyquist frequency is the half of the sampling frequency  $f_s/2$ , 800 Hz, 400 Hz, and 200 Hz, respectively. Let  $f_m$  (in this case 750 Hz) be the biggest frequency found in the input signal. If the sampling frequency is less than  $2f_m = 1500$  Hz, then there will be aliasing.

In the following figures for i, ii and iii, the scale and magnitude values for aliased frequencies are not exactly correct. Phase shifts in input signal cause that a pure addition of magnitudes will not hold. (The sum of two cosines with same frequency and phase shift of  $\pi$  is zero. However, in practice, this is rarely significant.)

i)  $f_s = 1600$  Hz, highest frequency component  $f_m = 750$  Hz. The inequality  $1600 > 2 \cdot 750$  holds, hence, there is no aliasing. All three frequencies can be recovered. See Figure 66.

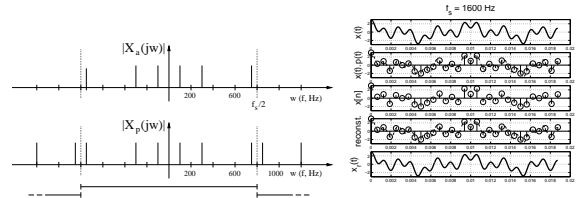


Figure 66: Sampling in Problem 47(i) with sampling frequency 1600 Hz: original analog spectrum  $X(j\Omega)$  (left top), and spectrum  $X(e^{j\omega})$  of the discrete-time signal (left bottom). Time domain view (right), top down  $x(t)$ , sampling  $x(t) \cdot p(t)$  sampled sequence  $x[n]$  to be processed with DSP, reconstruction, and reconstructed continuous-time signal  $x_r(t)$ . Again, in this case no aliasing, i.e.  $x(t) \equiv x_r(t)$ .

ii)  $f_s = 800$  Hz, highest frequency component  $f_m = 750$  Hz. The inequality  $800 > 2 \cdot 750$  does not hold, hence, there is aliasing. All frequencies over 400 Hz are missed (750 Hz in this case); they cannot be observed. There is a new alias component at frequency 50 Hz. See Figure 67.

Before going further, there is a short demonstration on the aliasing signal component  $x_3(t)$  ( $f_i = 750$  Hz) of the signal  $x(t)$  in Figure 68. The figures are depicted in time-domain: (a) original  $x(t)$  with period  $T = 1/f = (1/750) = 1.333$  ms, (b) samples  $x[n]$  using interval  $T_s = 1/f_s = (1/800) = 1.250$  ms, (c) reconstructed signal  $x_{3r}(t)$ , whose period  $T_r = (1/50) = 20$  ms. The same aliasing effect can be shown using the cosine function, which is  $2\pi$ -periodic ( $\cos(\omega n) \equiv \cos(\omega n + 2\pi)$ ) and even ( $\cos(-\omega n) \equiv \cos(\omega n)$ ). The highest component  $x_3(t)$  of 750 Hz aliases in the sampling and reconstructing process to 50 Hz:

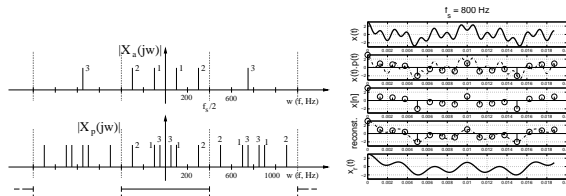


Figure 67: Sampling in Problem 47(ii) with sampling frequency 800 Hz. Aliasing occurs,  $x(t) \neq x_r(t)$ , compare the top and bottom axis in the figure right.

$$\begin{aligned} x_3(t) &= \cos(2\pi 750t) && | \text{original: 750 Hz} \\ x_3[n] &= x_3(n/f_s) = \cos(2\pi(750/800)n) = \cos(2\pi(750/800)n - 2\pi n) && | 2\pi\text{-periodicity} \\ &= \cos(2\pi(-50/800)n) = \cos(2\pi(50/800)n) && | \text{even function} \\ x_{3r}(t) &= \cos(2\pi 50t) && | \text{reconstructed: 50 Hz} \end{aligned}$$

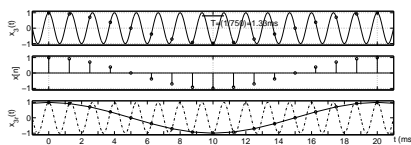


Figure 68: Problem 47(ii): Demonstration of aliasing of a single cosine. In top figure there is a continuous cosine  $x_3(t) = \cos(2\pi 750t)$  whose period is  $T = 1/750 \text{ s} \approx 1.33$  ms. Cosine is sampled with  $f_s = 800$  Hz, i.e.,  $T_s = 1/800 \text{ s} = 1.25$  ms. The sampling rate is too small in order to capture the cosine of 750 Hz, because  $750 \text{ Hz} > (800/2) \text{ Hz}$ . The sampled sequence  $x_3[n]$  is plotted in middle figure. Then the sequence is ideally reconstructed back to analog, that is, "slowest" possible cosine which fits the samples. In bottom figure 50 Hz cosine  $x_{3r}(t) = \cos(2\pi 50t)$  is plotted. Hence, cosine of 750 Hz is folded down (aliased) to 50 Hz. Note that any cosine  $x(t) = \cos(2\pi(\pm 50 + k \cdot f_s)t)$ ,  $k \in \mathbb{Z}$  goes through the points  $x_3[n] = \cos(2\pi(50/800)n) = \cos((1/80)\pi n)$ .

iii)  $f_s = 400$  Hz, highest frequency component  $f_m = 750$  Hz. The inequality  $400 > 2 \cdot 750$  does not hold, hence, there is aliasing. All frequencies over 200 Hz are missed (300 and 750 Hz). There are new alias components at 50 and 100 Hz. See Figure 69.

$$\begin{aligned} \cos\left(2\pi \frac{750}{400} n\right) &= \cos\left(2\pi \frac{750}{400} n - 4\pi n\right) = \cos\left(2\pi \frac{-50}{400} n\right) = \cos\left(2\pi \frac{50}{400} n\right) \\ \cos\left(2\pi \frac{300}{400} n\right) &= \cos\left(2\pi \frac{300}{400} n - 2\pi n\right) = \cos\left(2\pi \frac{-100}{400} n\right) = \cos\left(2\pi \frac{100}{400} n\right) \end{aligned}$$

After ideal reconstruction  $[x[n] \rightarrow x_r(t)]$  there are the following components left:  
(i) original 100, 300, 750 Hz.  
(ii) original 100, 300 Hz, and an alias 50 Hz.  
(iii) original 100 Hz, and aliases 50, 100 Hz.

There is a sampling (aliasing) demo `demosampling.m` in the course web pages.

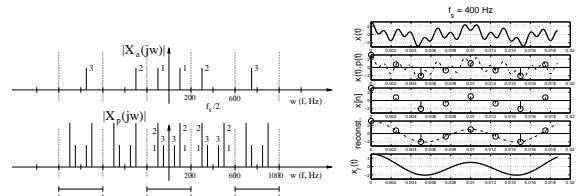


Figure 69: Sampling in Problem 47(iii) with sampling frequency 400 Hz. Aliasing occurs again,  $x(t) \neq x_r(t)$ .

48. **Problem:** Real analog signal  $x(t)$ , whose spectrum  $|X(j\Omega)|$  is drawn in Figure 70, is sampled with sampling frequency  $f_s = 8000$  Hz into a sequence  $x[n]$ .

- In the sampling process aliasing occurs. What would have been smallest sufficient sampling frequency, with which no aliasing would not happen?
- Analogue signal  $x(t)$  is 0.2 seconds long. How many samples are there in the sequence  $x[n]$ ?
- Sketch the spectrum  $|X(e^{j\omega})|$  of sampled sequence  $x[n]$ .
- Sequence  $x[n]$  is filtered with a LTI system, whose pole-zero plot is shown in Figure 70. After that filtered sequence  $y[n]$  is reconstructed (ideally) to continuous-time  $y_r(t)$ . Sketch the spectrum  $|Y_r(j\Omega)|$  in range  $f = [0 \dots 20]$  kHz.

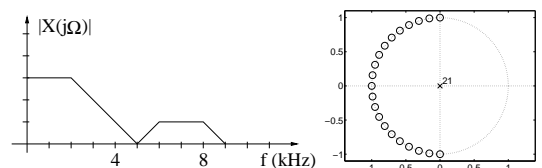


Figure 70: Problem 48: Spectrum left. Pole-zero plot right.

[L0419] **Solution:** The Shannon (Nyquist) sampling theorem is discussed in (Mitra 2Ed Sec. 5.2, p. 302 / 3Ed Sec. 4.2, p. 176).

- The highest frequency component in the analog signal is  $f_h = 9$  kHz. Thus the required sampling frequency would be  $f_r = 2f_h = 18$  kHz.
- The sampling frequency is 8 kHz, which means that there are 8000 samples each second. Now

$$\frac{x}{0.2 \text{ s}} = \frac{8000}{1 \text{ s}}$$

gives 16000 samples for 0.2 seconds.

- c) Due to too low sampling frequency aliasing occurs. All frequency components above  $f_T/2 = 4$  kHz are aliased into low frequencies between  $0 \dots 4$  kHz. Sampling in the frequency domain can be considered as copying the original analog spectrum at each multiple of the sampling frequency (Mitra 2Ed Eq. 5.9, p. 302 / 3Ed Eq. 4.10, p. 174)

$$G_{\text{discrete}}(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G_{\text{analog}}(j(\Omega + k\Omega_T))$$

Here we show another way of finding the discrete-time spectrum. If the analog signal is **real-valued** and the amplitude spectrum therefore symmetric around y-axis, the sampling can also be thought as flipping each  $f_T/2$  wide band of the analog spectrum downwards into the fundamental range  $0 \dots f_T/2$ .

First, divide the spectrum to a set of bands, whose width is  $f_T/2$ . In this case we have  $B_0 = [0 \dots 4000]$  Hz,  $B_1 = [4000 \dots 8000]$  Hz, and  $B_2 = [8000 \dots 12000]$  Hz.

Flip the band  $B_2$  around 8000 Hz (mirror) down to  $B_1$ . Spectral components of  $B_1$  and the mirrored  $B_2$  are summed together. See Figure 71 left top.

Finally, the band  $B_1$  is mirrored around 4000 Hz down to the fundamental band  $B_0$ , and the spectral components are summed together. See Figure 71 left bottom and the final result in Figure 71 right. Scaling factors (height) are here ignored.

To be exact,  $G_{\text{discrete}} = a \sum_k G_{\text{analog},k}$  is different from  $|G_{\text{discrete}}| = a \sum_k |G_{\text{analog},k}|$ , and therefore the procedure with amplitude spectra  $|X|$  shown in Figure 71 does not hold. However, in practice (and in any DSP book) this can be considered as a good approximation.

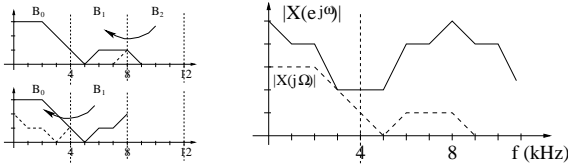


Figure 71: Problem 48: (a) Two steps when mirroring spectral bands down to the fundamental band  $0 \dots 4$  kHz. (b) Discrete-time spectrum  $|X(e^{j\omega})|$ . Scaling factors are ignored.

- d) If pole-zero plots are not familiar, consult (Mitra 2Ed Sec. 4.3.4 / 3Ed Sec. 6.7.4) or Problem 54, p. 96.

The sampling frequency is 8 kHz. In the pole-zero plot the whole circle  $2\pi$  corresponds 8 kHz. The upper part of the circle from  $\omega = 0$  to  $\omega = \pi$  is  $0 \dots 4$  kHz. There are zeros on the unit circle from  $2 \dots 4$  kHz. Hence, the filter is a lowpass filter, whose cut-off frequency is about at 2 kHz.

The discrete-time signal  $x[n]$  is filtered with a lowpass filter. Here we use a rough approximation of the filter depicted with a pole-zero plot, an ideal lowpass filter  $H(e^{j\omega})$  with cut-off at 2 kHz. The spectrum of filtered signal is  $Y(e^{j\omega}) = H(e^{j\omega}) \cdot X(e^{j\omega})$  shown in Figure 72(a). **Note!** Filter with zeros like in Figure 70 is not ideal. You can plot its real amplitude response using Matlab.

After reconstructing ideally (ideal lowpass filter with cut-off at  $f_T/2$ ) the sequence  $y[n]$  back to analog  $y_r(t)$ , the spectrum  $|Y_r(j\Omega)|$  is plotted in in Figure 72(b).

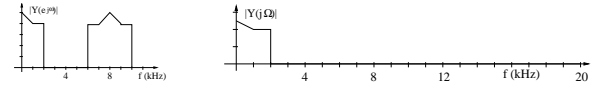


Figure 72: Problem 48: (a) Spectra  $|Y(e^{j\omega})|$  and (b)  $|Y_r(j\Omega)|$ . Discrete-time spectrum  $|Y(e^{j\omega})|$  is periodic every  $2\pi$  (or every  $f_T$ ), whereas continuous-time spectrum  $|Y_r(j\Omega)|$  is not periodic. Scaling factors are not taken into account.

49. **Problem:** Suppose that there is an analog signal which will be sampled with 8 kHz. The interesting band is  $0 \dots 2$  kHz. Sketch specifications for an anti-aliasing filter. Determine the order of the filter when using Butterworth approximation and minimum stopband attenuation is 50 dB. The variables in Table 7:  $\Omega_p$  is the passband edge frequency (interesting part),  $\Omega_T$  is the sampling frequency, and  $\Omega_0$  is the frequency after which the aliasing components are small enough.

$\Omega_0 =$	$2\Omega_p$	$3\Omega_p$	$4\Omega_p$
Attenuation (dB)	$6.02N$	$9.54N$	$12.04N$
$\Omega_T =$	$3\Omega_p$	$4\Omega_p$	$5\Omega_p$

Table 7: Approximate minimum stopband attenuation of a Butterworth lowpass filter (Mitra 2Ed Table 5.1, p. 336 / 3Ed Table 4.1, p. 210). See the text in Problem 49 for details.

[L0418] **Solution:** An is an analog lowpass filter used in order to remove components, which cause aliasing when sampling (Mitra 2Ed Sec. 5.6 / 3Ed Sec. 4.6). Consider an analog signal  $x(t)$  and its spectrum  $X(j\Omega)$  depicted in Figure 73.

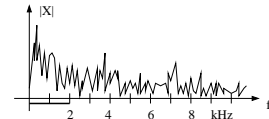


Figure 73: Spectrum  $X(j\Omega)$  in Problem 49.

In the following, notations of (Mitra 2Ed Fig. 5.28 / 3Ed Fig. 4.34) are used,  $\Omega_p$  for passband edge frequency,  $\Omega_0 = \Omega_T - \Omega_p$  for stopband edge frequency, and  $\Omega_T$  for sampling frequency. Now that the interesting band stops at  $\Omega_p = 2$  kHz and the sampling frequency is  $\Omega_T = 8$  kHz, we can set the edge frequency for the stopband to be at  $\Omega_0 = (8 - 2) = 6$  kHz (see Figure 74). After sampling there will be aliasing components in  $2 \dots 4$  kHz, but we are not interested in them, i.e. we use digital filtering for that band.

When the specifications are not so tight as they normally (cut-off at 4 kHz) are, also the order of the anti-aliasing filter is lower. The design of the anti-aliasing filter can be made even easier by increasing sampling frequency with analog circuits (order of anti-aliasing filter decreases), and afterwards decrease sampling frequency using multirate techniques (Mitra 2Ed Sec. 10 / 3Ed Sec. 13).

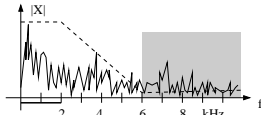


Figure 74: Problem 49: Spectrum  $X(j\Omega)$ , specifications for a LP filter (dashed line), frequency components that would alias in  $0 \dots 2$  kHz without anti-aliasing filtering (gray).

Calculations using (Mitra 2Ed Table 5.1 / 3Ed Table 4.1) or Table 7:  $\Omega_0/\Omega_p = 3 \rightarrow N = \lceil 50/9.54 \rceil = 6$ . Note that if the passband ended at 2 kHz and the stopband started at 4 kHz, the required order of the filter would be 10.

50. **Problem:** The exponent term in DFT/IDFT is commonly written  $W_N = e^{-j2\pi/N}$ .

- Compute and draw in complex plane values of  $W_3^k$
- Compute 3-DFT for the sequence  $x[n] = \{1, 3, 2\}$ .

[L0259] **Solution:** Discrete Fourier transform (DFT), left, and Inverse Fourier transform (IDFT), right, using  $N$  points are defined with  $W_N = e^{-j\frac{2\pi}{N}}$  as

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-kn}$$

DFT is discrete both in time and frequency, and can be calculated directly using computers. For example, in Matlab a spectrum of a signal is computed in discrete frequency points  $X[k]$ , but so densely, that it seems to be continuous  $X(e^{j\omega})$ .

- $W_N = e^{-j\frac{2\pi}{N}}$ , now  $N = 3$ .

$$W_3^0 = e^{-j\frac{2\pi}{3} \cdot 0} = 1$$

$$W_3^1 = e^{-j\frac{2\pi}{3} \cdot 1} = -0.5 - \frac{\sqrt{3}}{2}j$$

$$W_3^2 = e^{-j\frac{2\pi}{3} \cdot 2} = -0.5 + \frac{\sqrt{3}}{2}j$$

Notice that the exponent in  $W_N$  defines the uniform angle jump in clockwise direction.  $W_N$  is periodic with every  $N$ , e.g.  $W_3^{2 \cdot 2} = W_3^4 = e^{-j\frac{2\pi}{3} \cdot 4} = e^{-j\frac{2\pi}{3} \cdot 3} \cdot e^{-j\frac{2\pi}{3} \cdot 1} =$

$W_3^1$ . We can compute values of  $W_3^{kn}$ , when  $k = 0 \dots 2$  and  $n = 0 \dots 2$  into a table below

$k, n$	0	1	2
0	$W_3^{(0 \cdot 0)} = W_3^0$	$W_3^{(0 \cdot 1)} = W_3^0$	$W_3^{(0 \cdot 2)} = W_3^0$
1	$W_3^{(1 \cdot 0)} = W_3^0$	$W_3^{(1 \cdot 1)} = W_3^1$	$W_3^{(1 \cdot 2)} = W_3^2$
2	$W_3^{(2 \cdot 0)} = W_3^0$	$W_3^{(2 \cdot 1)} = W_3^2$	$W_3^{(2 \cdot 2)} = W_3^1$

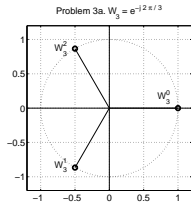


Figure 75: Values of  $W_N^{kn}$  in Problem 50.

b) The sequence  $x[n] = \{1, 3, 2\}$  is of length 3.

$$\begin{aligned}
 X[0] &= \sum_{n=0}^2 x[n]W^{0 \cdot n} \\
 &= 1 + 3 + 2 = 6 \\
 X[1] &= \sum_{n=0}^2 x[n]W^{1 \cdot n} \\
 &= 1 \cdot W^0 + 3 \cdot W^1 + 2 \cdot W^2 \\
 &= 1 + (-1.5 - \frac{3\sqrt{3}}{2}j) + (-1 + \frac{2\sqrt{3}}{2}j) = -1.5 - \frac{\sqrt{3}}{2}j \\
 X[2] &= \sum_{n=0}^2 x[n]W^{2 \cdot n} \\
 &= 1 \cdot W^0 + 3 \cdot W^2 + 2 \cdot W^4 \\
 &= 1 + (-1.5 + \frac{3\sqrt{3}}{2}j) + (-1 - \frac{2\sqrt{3}}{2}j) = -1.5 + \frac{\sqrt{3}}{2}j
 \end{aligned}$$

DFT is therefore  $X[k] = \{6, -1.5 - \frac{\sqrt{3}}{2}j, -1.5 + \frac{\sqrt{3}}{2}j\}$ .

**Remark.** Notice that

- DFT is discrete in frequency domain (DTFT is continuous)
- if having sampling frequency  $f_r$  Hz, then the spectral resolution  $\Delta f = f_r/N$
- N-point DFT of a real signal is generally complex-valued
- if N-point DFT is real-valued then  $x[n]$  has to be "symmetric"  $x[n] = x[-n]$
- each value of  $X[k]$  is a dot product of  $x[n]$  and  $W_N$  with some constant angle jump ( $nk$ )
- $X[0]$  is the sum of values of  $x[n]$  (DC-component)
- values of  $X[k]$  are N-periodic:  $X[k] = X[k + N] = X[k + 2N] = \dots$
- absolute values (amplitude spectrum) are even, e.g.,  $|X[1]| = |X[-1]|$
- angle values are odd, e.g.,  $\angle X[1] = -\angle X[-1]$

**Remark.** Discrete Fourier transform is a linear operation. It can be calculated in matrix form as (Mitra 2Ed Sec. 3.2.2 / 3Ed Sec. 5.2.2) and Problem 13

$$X = D_N x$$

where  $X$  is a column vector of the  $N$  frequency-domain DFT samples,  $x$  is a column vector of  $N$  time-domain input samples, and  $D_N$  is the  $N \times N$  DFT matrix (dfmtx in Matlab)

$$\begin{aligned}
 X &= [X[0] \ X[1] \ \dots \ X[N-1]]^T \\
 x &= [x[0] \ x[1] \ \dots \ x[N-1]]^T \\
 D_N &= \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^1 & W_N^2 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix}
 \end{aligned}$$

In this problem dfmtx gives

$$D_3 = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & -0.5000 - 0.8660i & -0.5000 + 0.8660i \\ 1.0000 & -0.5000 + 0.8660i & -0.5000 - 0.8660i \end{bmatrix}$$

For instance, when  $x = [1, 3, 2]^T$ , then  $X = D_3 x = [6, -1.5 - 0.8660j, -1.5 + 0.8660j]^T$  as earlier computed.

In the inverse DFT  $x = D_N^{-1} X$  the matrix  $D_N^{-1}$  is

$$D_N^{-1} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \dots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \dots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \dots & W_N^{-(N-1)(N-1)} \end{bmatrix}$$

It can be seen that  $D_N^{-1} = (1/N)D_N^H$ .

51. **Problem:** Consider a sequence

$$x_1[n] = 2\delta[n] + 8\delta[n-3] = \{2, 0, 0, 8\}$$

and its DFT-4

$$X_1[k] = 10\delta[k] + (2 + 8j)\delta[k-1] - 6\delta[k-2] + (2 - 8j)\delta[k-3] = \{10, 2 + 8j, -6, 2 - 8j\}$$

Having a circular shift (delay) of one,  $x_2[n] = x_1[\langle n-1 \rangle_4]$ , what is the DFT  $X_2[k]$ ? Here  $\langle \cdot \rangle_N$  computes modulo  $N$ .

[L0296] **Solution:** The circular shift of a sequence, see (Mitra 2Ed Sec. 3.4.1 / 3Ed Sec. 5.4.1, 5.7) and Problem 7(g), gives

$$x_2[n] = 8\delta[n-2] + 2\delta[n-3] = \{0, 0, 8, 2\}$$

Instead of computing DFT again we are applying circular time-shifting which has a DFT pair in (Mitra 2Ed Table 3.5, p. 141 / 3Ed Table 5.3, p. 264)

$$x[\langle n - n_0 \rangle_N] \leftrightarrow W_N^{-kn_0} X[k]$$

where  $W_N = e^{-j2\pi/N}$  (typo error in 2nd Edition).

Now we have  $N = 4$  and  $n_0 = 1$ . We compute the values  $X_2[k] = W_N^{-kn_0} X_1[k] = W_4^{-k} X_1[k]$  in a table below.

$k$	$X_1[k]$	$W_N^{-kn_0}$	$X_2[k]$
0	10	$W_4^0 = 1$	10
1	$2 + 8j$	$W_4^{-1} = j$	$-8 + 2j$
2	-6	$W_4^{-2} = -1$	6
3	$2 - 8j$	$W_4^{-3} = -j$	$-8 - 2j$

52. **Problem:** Let  $h[n]$  and  $x[n]$  be two finite-length sequences given below:

$$h[n] = \begin{cases} 5, & \text{for } n = 0, \\ 2, & \text{for } n = 1, \\ 4, & \text{for } n = 2 \end{cases} \quad x[n] = \begin{cases} -3, & \text{for } n = 0, \\ 4, & \text{for } n = 1, \\ 0, & \text{for } n = 2, \\ 2, & \text{for } n = 3 \end{cases}$$

- a) Determine the linear convolution  $y_L[n] = h[n] \otimes x[n]$ .
- b) Extend  $h[n]$  to length-4 sequence  $h_e[n]$  by zero-padding and compute the circular convolution  $y_C[n] = h_e[n] \circledast x[n]$ .
- c) Extend both sequences to length-6 sequences by zero-padding and compute the circular convolution  $y_C[n] = h_e[n] \circledast x_e[n]$ . Show that now  $y_C[n] = y_L[n]$ !

[L0270] **Solution:** In this problem linear convolution  $y_L[n]$  (Mitra 2Ed Sec. 2.5.1 / 3Ed Sec. 2.5.1) and circular convolution  $y_C[n]$  (Mitra 2Ed Sec. 3.4.2 / 3Ed Sec. 5.4.2) are computed using sequences  $h[n] = \{5, 2, 4\}$  and  $x[n] = \{-3, 4, 0, 2\}$ .

Linear convolution  $y[n] = h[n] \otimes x[n]$  can be computed using "flip and slide" method in Figure 76(a).  $x[n]$  is flipped and at each  $n$  the items are multiplied and finally all summed together. In the figure, when  $n = 1$ , it gives  $y_L[1] = h[0]x[1-0] + h[1]x[1-1] + h[2]x[1-2] = 20 - 6 + 0 = 14$ .

Computation of circular convolution  $y_C[n] = h[n] \circledast x[n]$  can be illustrated with "a circular buffer" of length  $N$  in Figure 76(b) and (c).  $x[n]$  is flipped and at each  $n$  the items are multiplied. There are always  $N$  terms to be added to get the result at  $n$ . In the figure, when  $N = 4$  and  $n = 1$ , it gives  $y_C[1] = h[0]x[1] + h[1]x[0] + h[2]x[3] + h[3]x[2] = 20 - 6 + 8 + 0 = 22$ .



Figure 76: Problem 52: Convolution depicted with "flip and slide" method, (a) left, linear convolution, (b) right, circular convolution with  $N = 4$ . As an example, in both cases the convolution sum is computed at  $n = 1$ .

a) Linear convolution:  $y_L[n] = \sum_{k=0}^2 h[k]x[n-k]$ . Its length will be  $L\{h[n]\} + L\{x[n]\} - 1 = 6$ . Using "flip around and slide":

$$\begin{aligned}
 y_L[0] &= h[0]x[0] = 5 \cdot (-3) = -15 \\
 y_L[1] &= h[0]x[1] + h[1]x[0] = 5 \cdot 4 + 2 \cdot (-3) = 14 \\
 y_L[2] &= h[0]x[2] + h[1]x[1] + h[2]x[0] = 5 \cdot 0 + 2 \cdot 4 + 4 \cdot (-3) = -4 \\
 y_L[3] &= h[0]x[3] + h[1]x[2] + h[2]x[1] = 5 \cdot 2 + 2 \cdot 0 + 4 \cdot 4 = 26 \\
 y_L[4] &= h[1]x[3] + h[2]x[2] = 2 \cdot 2 + 4 \cdot 0 = 4 \\
 y_L[5] &= h[2]x[3] = 4 \cdot 2 = 8
 \end{aligned}$$

Therefore,

$$y_L[n] = \{-15, 14, -4, 26, 4, 8\}$$

b) Circular convolution is computed in  $N = 4$  points

$$y_C[n] = h_e[n] \circledast x[n] = \sum_{k=0}^3 h_e[k] x[\langle n - k \rangle_4]$$

where  $h_e[n] = \{ \underline{5}, 2, 4, 0 \}$  is zero-extended version of  $h[n]$ , and  $\langle n \rangle_4$  is modulo-4 operation. Hence,  $h[\langle n - 5 \rangle_4] = h[\langle n - 1 \rangle_4]$ , i.e. the sequence can be thought to be periodic with the period  $\{h[0], h[1], h[2], h[3]\}$ .

$$\begin{aligned} y_C[0] &= h_e[0]x[\langle 0 - 0 \rangle_4] + h_e[1]x[\langle 0 - 1 \rangle_4] + h_e[2]x[\langle 0 - 2 \rangle_4] + h_e[3]x[\langle 0 - 3 \rangle_4] \\ &= h_e[0]x[0] + h_e[1]x[3] + h_e[2]x[2] + h_e[3]x[1] \\ &= 5 \cdot (-3) + 2 \cdot 2 + 4 \cdot 0 + 0 \cdot 4 = -11 \\ y_C[1] &= h_e[0]x[1] + h_e[1]x[0] + h_e[2]x[3] + h_e[3]x[2] \\ &= 5 \cdot 4 + 2 \cdot (-3) + 4 \cdot 2 + 0 \cdot 0 = 22 \\ y_C[2] &= h_e[0]x[2] + h_e[1]x[1] + h_e[2]x[0] + h_e[3]x[3] \\ &= 5 \cdot 0 + 2 \cdot 4 + 4 \cdot (-3) + 0 \cdot 2 = -4 \\ y_C[3] &= h_e[0]x[3] + h_e[1]x[2] + h_e[2]x[1] + h_e[3]x[0] \\ &= 5 \cdot 2 + 2 \cdot 0 + 4 \cdot 4 + 0 \cdot (-3) = 26 \end{aligned}$$

Thus,

$$y_C[n] = \{ -11, 22, -4, 26 \}$$

c) Circular convolution using  $N = 6$  points

$$y_C[n] = h_e[n] \circledast x_e[n] = \sum_{k=0}^5 h_e[k] x_e[\langle n - k \rangle_6]$$

where  $h_e[n] = \{ \underline{5}, 2, 4, 0, 0, 0 \}$ , and  $x_e[n] = \{ \underline{-3}, 4, 0, 2, 0, 0 \}$  are zero-padded versions. Computing like in (b) the result is

$$y_C[n] = \{ -15, 14, -4, 26, 4, 8 \} \equiv y_L[n]$$

If  $N$  in circular convolution is chosen so that  $N \geq L\{y_L[n]\}$ , then  $y_C[n] = y_L[n]$ .

**Remark.** Circular convolution has a close connection to Discrete Fourier Transform (DFT). For example, in (b)

$$y_C[n] = h_e[n] \circledast x[n] \quad \text{DFT-4} \quad H_e[k] \cdot X[k] = Y_C[k] \quad \text{IDFT-4} \quad y_C[n]$$

53. **Problem:** Consider a LTI system depicted in Figure 77 with registers having initial values of zero and the input sequence  $x[n] = (-0.8)^n \mu[n]$ .

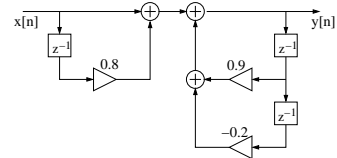


Figure 77: LTI system of Problem 53.

- What is the difference equation of the system?
- Compute  $X(z)$  using the definition of z-transform or consult the z-transform table.
- Apply z-transform to the difference equation. What is the transfer function  $H(z) = Y(z)/X(z)$ ? Where are the constant multipliers of the system seen in Figure 77 in difference equation and in transfer function? Hint: the z-transform of  $K w[n - n_0]$  is  $K z^{-n_0} W(z)$ .
- Now it is possible to compute the output  $y[n]$  without convolution in time-domain using the convolution theorem

$$y[n] = h[n] \otimes x[n] \quad \leftrightarrow \quad Y(z) = H(z) \cdot X(z)$$

Write down the equation for  $Y(z)$ , use partial fraction expansion in order to achieve rational polynomials of first order, and then use the inverse z-transform (equation in (b)).

[L0326] **Solution:**

- The input-output-relation is  $y[n] - 0.9y[n - 1] + 0.2y[n - 2] = x[n] + 0.8x[n - 1]$ . Notice that the coefficients in the diagram are also present in the difference equation (past output values maybe as opposite numbers).
- If computing using the definition, see Problem 38(b). From the z-transform table directly:

$$\begin{aligned} Z\{a^n \mu[n]\} &= \frac{1}{1 - az^{-1}} \quad \text{ROC: } |z| > |a| \\ (-0.8)^n \mu[n] &\leftrightarrow \frac{1}{1 + 0.8z^{-1}} \quad |z| > 0.8 \end{aligned}$$

- Using the z-transform pair  $K \cdot w[n - n_0] \leftrightarrow K \cdot z^{-n_0} W(z)$ :

$$\begin{aligned} y[n] - 0.9y[n - 1] + 0.2y[n - 2] &= x[n] + 0.8x[n - 1] \quad | \text{z-transform} \\ Y(z) - 0.9z^{-1}Y(z) + 0.2z^{-2}Y(z) &= X(z) + 0.8z^{-1}X(z) \\ Y(z)(1 - 0.9z^{-1} + 0.2z^{-2}) &= X(z)(1 + 0.8z^{-1}) \\ Y(z) &= X(z) \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \quad | / X(z) \\ H(z) = Y(z)/X(z) &= \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \quad \text{ROC: } |z| > 0.5 \end{aligned}$$

The flow (block) diagram was given in direct form (DF) (Mitra 2Ed Sec. 6.4.1 / 3Ed Sec. 8.4.1). The coefficients of the diagram are that of the difference equation and transfer function. Coefficients in the loop (IIR subfilter) are in the denominator polynomial and coefficients of the FIR part can be found in the numerator polynomial.

d) Using convolution theorem

$$\begin{aligned} Y(z) &= H(z) \cdot X(z) \\ &= \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \cdot \frac{1}{1 + 0.8z^{-1}} \\ &= \frac{1}{1 - 0.9z^{-1} + 0.2z^{-2}} \quad | \text{partial fraction expansion} \\ &= \frac{5}{1 - 0.5z^{-1}} + \frac{-4}{1 - 0.4z^{-1}} \quad | \text{inverse z-transform} \\ y[n] &= 5 \cdot (0.5)^n \mu[n] - 4 \cdot (0.4)^n \mu[n] \end{aligned}$$

54. **Problem:** Consider the pole-zero plots in Figure 78.

- What is the order of each transfer function?
- Are they FIR or IIR?
- Sketch the amplitude response for each filter.
- What could be the transfer function of each filter?

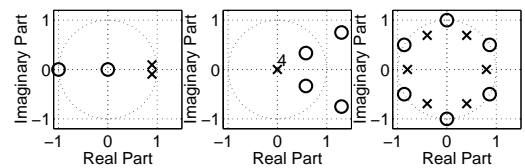


Figure 78: Pole-zero plots of LTI systems in Problem 54.

[L0327] **Solution:** The z-transform of the impulse response  $h[n]$  of the LTI system is the transfer function  $H(z)$  (with certain regions of convergence, ROCs, see Problem 56). It can be written as a rational function in  $z^{-1}$  as follows

$$\begin{aligned} H(z) = \frac{B(z)}{A(z)} &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \\ &= K \cdot \frac{(1 - z_1 z^{-1}) \cdot (1 - z_2 z^{-1}) \cdot \dots \cdot (1 - z_M z^{-1})}{(1 - p_1 z^{-1}) \cdot (1 - p_2 z^{-1}) \cdot \dots \cdot (1 - p_N z^{-1})} = K \cdot \frac{\prod_{k=1}^M (1 - z_k z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})} \end{aligned}$$

where  $b_i$  are the coefficients of the numerator polynomial  $B(z)$ , and  $a_i$  are the coefficients of the denominator polynomial  $A(z)$ . The filter order, that is, order of  $H(z)$  is  $\max\{M, N\}$ . Those points  $z_i$  where  $B(z) = 0$  are called "zeros", and points  $p_i$  where  $A(z) = 0$  are called "poles". The figure with zeros (circles) and poles (crosses) plotted in the complex plane is called "pole-zero plot" (diagram) of the transfer function.

**The rules of thumb** for determining amplitude response from the pole-zero-diagram (Mitra 2Ed Sec. 4.3.4 / 3Ed Sec. 6.7.4)

- Examine the frequencies  $\omega \in (0 \dots \pi)$ , in other words, the observation point moves on the unit circle counterclockwise from  $(1, 0j)$  to  $(-1, 0j)$ . In each point the amplitude response  $|H(e^{j\omega})|$  is estimated. A "simple" function  $H(e^{j\omega})$  has a smooth response.
- The amplification is big, when a pole is close to unit circle (a small factor in denominator) or a zero is far from unit circle. The closer the pole is to unit circle, the narrower the amplification is in frequency area.
- The amplification is small, when a pole is far from the unit circle (big factors in denominator) or there is a zero close to unit circle.
- The amplification is zero, if a zero is on the unit circle at observation frequency.
- Poles or zeros in the origo do not affect at all because the distance is always 1.



6. The amplification cannot be found from pole-zero plot. Normally  $H(e^{j\omega})$  is scaled so that the maximum value is set to be 1:  $H(e^{j\omega}) \leftarrow H(e^{j\omega}) / \max\{|H(e^{j\omega})|\}$ .
- a) The order is the maximum of the number of poles or zeros (not in origo).  
So, (i) 2 poles, 1 zero: 2nd order; (ii) 4 zeros: 4th order; (iii) 6 poles, 6 zeros: 6th order.
- Note, in analog  $H(s)$  there are only poles, but in digital  $H(z)$  there can be both poles and zeros.
- b) If there is any pole (cross in the graph) outside the origo, it means that there is at least first-order polynomial in the denominator in  $H(z) \Leftrightarrow$  there is a feedback in the system  $\Leftrightarrow$  IIR.  
Hence, (i) IIR; (ii) FIR; (iii) IIR.
- c) The analysis with graphs is done below for each case separately.  
Shortly, (i) lowpass with narrow passband; (ii) highpass; (iii) a comb filter.
- d) The transfer function can be constructed from zeros  $z_i$  and poles  $p_i$

$$H(z) = K \cdot \frac{\prod_{k=1}^M (1 - z_k z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})}$$

However, the scaling factor  $K$  cannot be seen from the pole-zero-plot. Therefore  $K$  is set so that  $\max\{|H(e^{j\omega})|\} = 1$ .

Next, a closer look at (c) and (d) is given for each filter.

- i) Without computing any exact values of the amplitude response, it is possible to approximate it by looking at the positions of zeros and poles. The angular frequency gets values from 0 to  $\pi$ , and the observation is done on a unit circle counterclockwise. Poles are close to unit circle at  $\omega = \pm\pi/30$  in Figure 79(a). Therefore the amplitude response gets the maximum approximately at that frequency and the filter is lowpass type, see the sketch in Figure 79(b). The closer the poles are the unit circle, the narrower the maximum area is. The value at  $\omega = \pi$  is zero.
- In this case the exact locations of poles and zeros were known ( $z_1 = -1, p_1 = 0.8950 + 0.0947i, p_2 = 0.8950 - 0.0947i$ ). The actual transfer function is  $H(z) = K \cdot (1 + z^{-1}) / (1 - 1.79z^{-1} + 0.81z^{-2})$  from which the actual frequency response is received by  $z \leftarrow e^{j\omega}$ . Some values in range  $0 \dots \pi$  are computed below, and  $K$  is chosen so that the maximum of  $|H(z)|$  is one. Figures are plotted using Matlab in both linear scale and in logarithmic scale in Figure 79(c) and (d), respectively.
- | $\omega$ | $H(e^{j\omega})$  | $ H(e^{j\omega}) $ | $\omega$ | $H(e^{j\omega})$  | $ H(e^{j\omega}) $ |
|----------|-------------------|--------------------|----------|-------------------|--------------------|
| 0        | 1                 | 1                  | $3\pi/4$ | $-0.008 + .0023j$ | .0025              |
| $\pi/4$  | $-.0277 + .0210j$ | .0348              | $\pi$    | 0                 | 0                  |
| $\pi/2$  | $-.0049 + .0061j$ | .0078              |          |                   |                    |
- ii) There are four zeros in Figure 80(left). At  $\omega \approx \pi/6$  the zeros are closest to the observation point, and the minimum of the response is probably reached (bandstop). At  $\omega = \pi$  the zeros are much further away than at  $\omega = 0$ , so the attenuation is much stronger at low frequencies (highpass). Notice that  $|H(e^{j0})| \neq 0$ , because there is not a zero on the unit circle at  $\omega = 0$ . The filter can be a highpass or bandstop FIR filter.
- Actually,  $H(z) = 1 - 3.753z^{-1} + 5.694z^{-2} - 3.753z^{-3} + z^{-4}$ . Filter coefficients have a certain symmetry as well as the zeros lie in a certain symmetry, which implies a

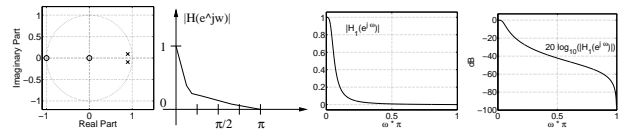


Figure 79: Problem 54(i): (a) Pole-zero-diagram, (b) an example of approximated amplitude response, (c) actual amplitude response  $|H_1(e^{j\omega})|$  in linear scale, (d) actual amplitude response  $|H_1(e^{j\omega})|$  in decibels.

linear-phase filter, see Problem 59. The minimum of  $|H(e^{j\omega})| \approx 0.0114$  (scaled) at  $\omega \approx 0.11\pi$ , which is different from  $\pi/6$  estimated earlier. All “zero vectors” affect to the response, see the remark text below.

- iii) Zeros are on the unit circle at uniform intervals forcing the amplification drop down to zero, see Figure 80(right). This type of periodic filter is often called a comb filter. The maximum is scaled to one. Note that all poles and zeros affect, so that if there were not exactly same intervals between poles and zeros, the amplitude response would also turn out to be non-symmetric.

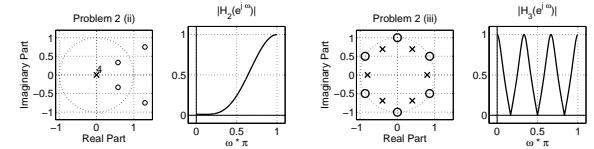


Figure 80: Problem 54(ii),(iii): Pole-zero-diagram and corresponding amplitude response of  $|H_2(e^{j\omega})|$  left, and  $|H_3(e^{j\omega})|$  right.

**Remark.** Determining amplitude response from the pole-zero-diagram, theory in background.

Any transfer function  $H(z)$  can be expressed in form of

$$H(z) = \frac{p_0 (1 - z_1 z^{-1})(1 - z_2 z^{-1}) \dots (1 - z_M z^{-1})}{d_0 (1 - p_1 z^{-1})(1 - p_2 z^{-1}) \dots (1 - p_N z^{-1})}$$

In order to achieve this, all zeros ( $z_i$ ) and poles ( $p_i$ ) of  $H(z)$  has to be computed. Zeros are the roots of the numerator polynomial and poles are the roots of the denominator polynomial. Numerator part is “FIR part” (always stable,  $y[n]$  depends only on values of  $x[n - k_i]$ ), denominator is “IIR part” (feedback, in order to compute  $y[n]$  some old values of it has to be used).

Frequency response is the transfer function computed on unit circle, i.e. substitution  $z = e^{j\omega}$ :

$$H(e^{j\omega}) = \frac{p_0 (1 - z_1 e^{-j\omega})(1 - z_2 e^{-j\omega}) \dots (1 - z_M e^{-j\omega})}{d_0 (1 - p_1 e^{-j\omega})(1 - p_2 e^{-j\omega}) \dots (1 - p_N e^{-j\omega})}$$

We are interested in amplitude response  $|H(e^{j\omega})|$ . Because the expression is in a product form, the absolute value of  $|H(e^{j\omega})|$  can be computed with its first order blocks. Let

$K = |p_0|/|d_0|$ ,  $B_i$  be the length of a first order block in numerator polynomial, and  $A_i$  the length of a first order block in denominator polynomial:

$$|H(e^{j\omega})| = K \cdot \frac{\underbrace{|(1 - z_1 e^{-j\omega})|}_{B_1}}{\underbrace{|(1 - p_1 e^{-j\omega})|}_{A_1}} \frac{\underbrace{|(1 - z_2 e^{-j\omega})|}_{B_2}}{\underbrace{|(1 - p_2 e^{-j\omega})|}_{A_2}} \dots \frac{\underbrace{|(1 - z_M e^{-j\omega})|}_{B_M}}{\underbrace{|(1 - p_N e^{-j\omega})|}_{A_N}} = K \cdot \prod_{k=1}^M \frac{B_k}{A_k}$$

The frequency axis lies on the unit circle from  $\omega = 0$ , which is a complex point  $e^{j\omega}|_{\omega=0} = 1$  to  $\omega = \pi$ , which is situated at  $e^{j\omega}|_{\omega=\pi} = -1$ . The observation frequency  $\omega_0$  gets values  $0 \dots \pi$ .

$B_i$  is called a “zero vector”, i.e. it is the length from the observation point  $\omega_0$  to zero  $i$ .  $A_i$  is a “pole vector” correspondingly.

Any small  $A_i$  (pole close to unit circle) gives big value of  $|H(e^{j\omega})|$ . Any small  $B_i$  (zero close to unit circle) decreases  $|H(e^{j\omega})|$ . However, it should be noticed that  $|H(e^{j\omega})|$  is a product of **all** zero vectors and **all** pole vectors.

For example, in Figure 81(a)  $M = 2$  and  $N = 2$ :

$$|H(e^{j\omega})| = K \cdot \frac{\underbrace{|(1 - z_1 e^{-j\omega})|}_{B_1}}{\underbrace{|(1 - p_1 e^{-j\omega})|}_{A_1}} \frac{\underbrace{|(1 - z_2 e^{-j\omega})|}_{B_2}}{\underbrace{|(1 - p_2 e^{-j\omega})|}_{A_2}}$$

It can be roughly estimated that the filter is highpass, because around  $\omega = 5\pi/6$   $A_1$  is smallest and therefore  $|H(e^{j\omega})|$  is at maximum. Actually the maximum might be at  $\omega = \pi$ , where  $A_1 \cdot A_2$  is probably smaller. The rough estimate of the amplitude response ( $0 \dots \omega_0 \dots \pi$ ) is given in Figure 81(b).

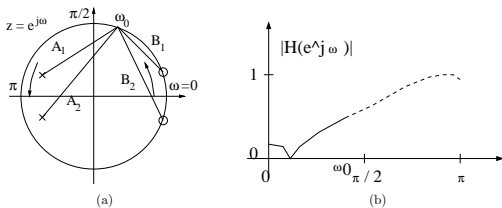


Figure 81: (a) Zero vectors  $B_k$  and pole vectors  $A_k$ .  $\omega$  runs  $0 \dots \pi$ . (b) Amplitude response roughly from the pole-zero-diagram.

**The rules of thumb** were given on page 96.

It can also be seen that the frequency response in discrete-time domain is always  $2\pi$ -periodic. Because  $|H(e^{j\omega})|$  is an even function, it is only necessary to draw angles  $0 \dots \pi$ .

55. **Problem:** Consider the filter described in Figure 82.

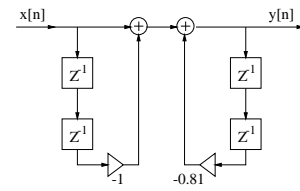


Figure 82: LTI system of Problem 55.

- Derive the difference equation of the system.
- Calculate the transfer function  $H(z)$ .
- Calculate the zeros and poles of  $H(z)$ . Sketch the pole-zero plot. Is the system stable and/or causal?
- If the region of convergence (ROC) of  $H(z)$  includes the unit circle, it is possible to derive frequency response  $H(e^{j\omega})$  by applying  $z = e^{j\omega}$ . Do this!
- Sketch the magnitude (amplitude) response  $|H(e^{j\omega})|$  roughly. Which frequency gives the maximum value of  $|H(e^{j\omega})|$ ? (If you want to calculate magnitude response explicitly, calculate  $|H(e^{j\omega})|^2 = H(e^{j\omega})H(e^{-j\omega})$  and use Euler’s formula.)
- Compute the equation for the impulse response  $h[n]$  using partial fraction expansion and inverse z-transform.

[L0328] **Solution:** Notice that the same filter can be represented (i) as a block diagram, (ii) with a difference equation, (iii) with a transfer function (and ROC), (iv) with an impulse response, (v) with poles, zeros and gain.

- Difference equation:  $y[n] = x[n] - x[n - 2] - 0.81y[n - 2]$
- Transfer function  $H(z)$  can be obtained from  $h[n]$  using z-transform pairs:

$$\begin{aligned} Z\{x[n]\} &= X(z) \\ Z\{a \cdot x[n - n_0]\} &= a \cdot z^{-n_0} \cdot X(z) \quad \text{ROC: } |z| > |a| \end{aligned}$$

Hence,

$$\begin{aligned} y[n] &= x[n] - x[n - 2] - 0.81y[n - 2] \\ Y(z) &= X(z) - z^{-2}X(z) - 0.81z^{-2}Y(z) \\ (1 + 0.81z^{-2}) \cdot Y(z) &= (1 - z^{-2}) \cdot X(z) \\ Y(z) &= X(z) \cdot \frac{1 - z^{-2}}{1 + 0.81z^{-2}} \\ H(z) = \frac{Y(z)}{X(z)} &= \frac{1 - z^{-2}}{1 + 0.81z^{-2}} \quad \text{ROC: } |z| > 0.9 \end{aligned}$$

- c) Zeros are the points, where the numerator of transfer function  $H(z)$  is zero:  
 $1 - z^{-2} = 0 \Leftrightarrow z^2 = 1 \Leftrightarrow z = \pm 1$ .  
 Poles are the points, where the denominator of transfer function  $H(z)$  is zero:  
 $1 + 0.81z^{-2} = 0 \Leftrightarrow z^2 = -0.81 \Leftrightarrow z = \pm 0.9j$   
 The pole-zero plot of the system is (a common notation is to use a  $\circ$  for a zero and a  $\times$  for a pole) in Figure 83(a).  
 The system is **causal**, because current output does not depend on future values of  $x[n]$  and  $y[n]$  (time-domain view). The system is **stable**, because the impulse response  $h[n]$  is absolutely summable (time-domain view).  
 On the other hand, if all poles in the pole-zero plot are inside the unit circle, i.e., the region of convergence (ROC) includes both the unit circle and the infinity, the filter is causal and stable (see Problem 56).
- d) Frequency response of the system  $H(e^{j\omega})$  (continuous systems  $H(j\Omega)$ ) is obtained by applying  $z = e^{j\omega}$  (continuous  $s = j\Omega$ ). If the unit circle is contained in the ROC, it is possible to apply  $H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$ :

$$H(e^{j\omega}) = \frac{1 - e^{-2j\omega}}{1 + 0.81e^{-2j\omega}}$$

- e) The amplitude response can be computed as exact as wanted using the mathematical functions. It can be computed also in specific points using calculator or computer. These will be explained after the roughest way, which is graphical approximation from poles and zeros.

The sketch of the magnitude (amplitude) response  $|H(e^{j\omega})|$  can be drawn by using pole-zero plot. There are zeros at  $z = 1$  and  $z = -1$ . The corresponding angular frequencies are  $0$  and  $\pi$ , because  $e^{j0} = 1 + 0j$  and  $e^{j\pi} = -1 + 0j$ . Hence, amplitude response is zero when  $\omega = 0$  and  $\omega = \pi$ . It is also clear that the maximum value is at  $\omega = \pi/2$ , where the pole is closest to the unit circle. A sketch is given in Figure 83(b).

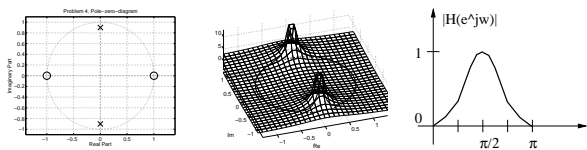


Figure 83: Problem 55: (a) Pole-zero plot of  $H(z)$  and (b)  $|H(z)|$  as a grid and  $|H(e^{j\omega})|$  with solid curve plotted with Matlab. (c) A rough sketch of amplitude response by hands using pole-zero.

Second, the amplitude response  $H(e^{j\omega}) = (1 - e^{-2j\omega}) / (1 + 0.81e^{-2j\omega})$  can be calculated in certain points. More points, more exact amplitude response. Start with points  $\omega = \{0, \pi/4, \pi/2, 3\pi/4, \pi\}$ , and calculate more if it seems to be appropriate. If your calculator does not support complex exponentials, decompose them by Euler's formula. (Notice that in Matlab you can use directly function `exp`.) A new sketch is drawn in Figure 84.

$\omega$	$ H(e^{j\omega}) $	$ H(e^{j\omega}) $	$\omega$	$ H(e^{j\omega}) $	$ H(e^{j\omega}) $
0	0	0	$5\pi/8$	0.6352 + 2.5067j	2.5859
$\pi/8$	0.0199 - 0.4568j	0.4573	$3\pi/4$	0.1147 + 1.0929j	1.0989
$\pi/4$	0.1147 - 1.0929j	1.0989	$7\pi/8$	0.0199 + 0.4568j	0.4573
$3\pi/8$	0.6352 - 2.5067j	2.5859	$\pi$	0	0
$\pi/2$	10.5263 - 0.0000j	10.5263			

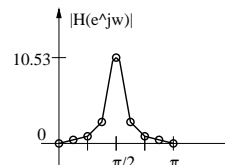


Figure 84: A sketch of amplitude response after computing several values in Problem 55(e).

Third, the magnitude response can (only sometimes) be simplified. For example, this time the simplified version is relatively simple. Simplification is sometimes needed to some proofs, etc.

$$\begin{aligned} |H(e^{j\omega})|^2 &= H(e^{j\omega})H^*(e^{j\omega}) = H(e^{j\omega})H(e^{-j\omega}) \quad | \text{ complex conjugate} \\ &= \frac{1 - e^{-2j\omega}}{1 + 0.81e^{-2j\omega}} \cdot \frac{1 - e^{+2j\omega}}{1 + 0.81e^{+2j\omega}} \\ &= \frac{1 + 1 - (e^{2j\omega} + e^{-2j\omega})}{1 + 0.81^2 + 0.81(e^{2j\omega} + e^{-2j\omega})} \\ &= \frac{2 - 2 \cos(2\omega)}{1.6561 + 1.62 \cos(2\omega)} \quad | \text{ square} \end{aligned}$$

$|H(e^{j\omega})|$  gets the maximum value at frequency  $\omega = \frac{\pi}{2}$ . The maximum value is

$$|H(e^{j\omega})|_{\max} = |H(e^{j\frac{\pi}{2}})| \approx 10.53$$

- f) Notice that the partial fraction expansion can be written in various forms, see Problem 10, for instance. The transform pair  $a^n \mu[n] \leftrightarrow \frac{1}{1 - az^{-1}}$  is applied again.

$$\begin{aligned} H(z) &= \frac{1 - z^{-2}}{1 + 0.81z^{-2}} \\ &= \frac{1}{1 + 0.81z^{-2}} - z^{-2} \cdot \frac{1}{1 + 0.81z^{-2}} \quad | \text{ part. frac. exp.} \\ &= \left[ \frac{0.5}{1 - 0.9jz^{-1}} + \frac{0.5}{1 + 0.9jz^{-1}} \right] - z^{-2} \left[ \frac{0.5}{1 - 0.9jz^{-1}} + \frac{0.5}{1 + 0.9jz^{-1}} \right] \\ h[n] &= 0.5 \cdot ((0.9j)^n \mu[n] + (-0.9j)^n \mu[n]) - \\ &\quad 0.5 \cdot ((0.9j)^{n-2} \mu[n-2] + (-0.9j)^{n-2} \mu[n-2]) \\ &\approx \{ \underline{1.0000}, 0, -1.8100, 0, 1.4661, 0, -1.1875, \dots \} \end{aligned}$$

Matlab (`residuez`) may give a different form of the same sequence:

$$h[n] \approx -1.2346 \cdot \delta[n] + 1.1173 \cdot (0.9j)^n \mu[n] + 1.1173 \cdot (-0.9j)^n \mu[n]$$

56. **Problem:** The transfer function of a filter is

$$H(z) = \frac{1 - z^{-1}}{1 - 2z^{-1} + 0.75z^{-2}}$$

- Compute the zeros and poles of  $H(z)$ .
- What are the three different regions of convergence (ROC)?
- Determine the ROC and the impulse response  $h[n]$  so that the filter is causal.
- Determine the ROC and the impulse response  $h[n]$  so that the filter is stable.

[L0329] **Solution:** Let us begin by reviewing some properties (Mitra 2Ed Sec. 3.8 / 3Ed Sec. 6.3)

- The filter is causal  $\Leftrightarrow \infty$  belongs to the region of convergence (ROC).
- The filter is stable  $\Leftrightarrow$  unit circle belongs to ROC,  $H(z)$  converges on the unit circle.
- ROC on  $z$ -plane must not contain any poles; it may be a ring between two poles, the disc limited by the closest pole to origin or the plane outside the most distant pole from origin.
- It is easiest to do the the inverse  $z$ -transform (here) by calculating first the fractional expansion of the  $H(z)$  and then inverting each part of it individually using the sum of a geometric series.
- The sum of a geometric series is

$$\sum_{k=0}^{\infty} q^k = \frac{1}{1 - q}, \quad |q| < 1$$

- The  $z$ -transform of  $h[n]$  is

$$\sum_{n=-\infty}^{\infty} h[n]z^{-k}$$

- First we have to solve the poles and zeros:

$$H(z) = \frac{1 - z^{-1}}{1 - 2z^{-1} + 0.75z^{-2}} = \frac{z(z - 1)}{z^2 - 2z + 0.75}$$

Poles:

$$z^2 - 2z + 0.75 = 0 \Leftrightarrow z = \frac{2 \pm \sqrt{4 - 4 \cdot 0.75}}{2} \Leftrightarrow z_1 = 0.5, z_2 = 1.5$$

Zeros:

$$z(z - 1) = 0 \Leftrightarrow z_1 = 0, z_2 = 1$$

- Now we may answer to the questions about stability and causality using different ROCs, see Figure 85:

- If we require causality, the region of convergence has to include  $z = \infty$ . Thus, the region of convergence has to be "outside" the pole  $z = 1.5$ , that is  $|z| > 1.5$ .
- If we require stability, the unit circle has to be on the region of convergence. Thus the region is a ring between the poles:  $0.5 < |z| < 1.5$
- If ROC is the inner circle  $|z| < 0.5$ , we will have a noncausal and unstable filter.

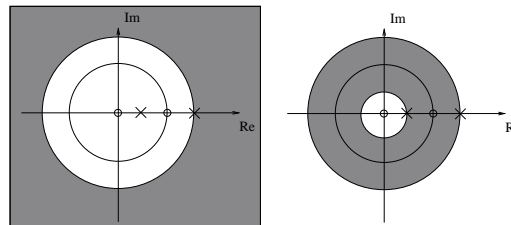


Figure 85: Region of convergence (ROC) in gray in Problem 56: (i)  $\infty$  belongs to ROC - causal filter, (ii) unit circle belongs to ROC - stable filter.

Note, that in this case we cannot have a filter that is both causal and stable.

At this point, when we calculate the impulse response  $h[n]$ , we have to do an inverse  $z$ -transformation for the transfer function  $H(z)$ . To do this we express the  $H(z)$  as a partial fraction expansion as then we may apply the formula of the sum of a geometric series.

Using the poles and zeros we may write the transfer function as follows:

$$\begin{aligned} H(z) &= \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} = \frac{1 - z^{-1}}{(1 - 0.5z^{-1})(1 - 1.5z^{-1})} \\ &\Leftrightarrow H(z) = \frac{A}{1 - 0.5z^{-1}} + \frac{B}{1 - 1.5z^{-1}} \\ &\Leftrightarrow 1 - z^{-1} \equiv A(1 - 1.5z^{-1}) + B(1 - 0.5z^{-1}) \end{aligned}$$

We solve  $A$  and  $B$  by letting  $z \rightarrow 0.5$  and  $z \rightarrow 1.5$

$$\begin{aligned} z \rightarrow 0.5: \quad 1 - 0.5^{-1} &= A(1 - 1.5 \cdot 0.5^{-1}) + B(1 - 0.5 \cdot 0.5^{-1}) \\ &\Rightarrow A = 0.5 \\ z \rightarrow 1.5: \quad 1 - 1.5^{-1} &= A(1 - 1.5 \cdot 1.5^{-1}) + B(1 - 0.5 \cdot 1.5^{-1}) \\ &\Rightarrow B = 0.5 \end{aligned}$$

Now we may write the expansion

$$H(z) = \frac{0.5}{1 - 0.5z^{-1}} + \frac{0.5}{1 - 1.5z^{-1}}$$

- Causal filter  $\Rightarrow$  we know that  $|z| > 1.5$ . We notice that both fractions in

$$H(z) = \frac{0.5}{1 - 0.5z^{-1}} + \frac{0.5}{1 - 1.5z^{-1}}$$

represent a sum of a geometric series, as  $|0.5z^{-1}| < 1$  and  $|1.5z^{-1}| < 1$  as required. We conclude

$$h_{\text{causal}}[n] = Z^{-1}\{H(z)\} = 0.5 \cdot 0.5^n \mu[n] + 0.5 \cdot 1.5^n \mu[n]$$

See Figure 86(a), the impulse response grows to infinity, i.e. it is not absolutely summable, and therefore the filter is not stable with the criterion  $\sum_n |h[n]| < \infty$ .

d) Stable filter  $\Rightarrow$  we know that  $0.5 < |z| < 1.5$ . We note that  $\sum_{n=0}^{\infty} 1.5^n z^{-n}$  does not converge as  $|\frac{1.5}{z}| \geq 1$ . We have to convert the expression to higher terms in order to get the denominator to suitable form:

$$\begin{aligned} H_{p2}(z) &= \frac{1}{2} \cdot \frac{1}{1 - (3/2)z^{-1}} \quad | \quad \cdot (-2/3)z / (-2/3)z \\ &= -\frac{1}{3} z \frac{1}{1 - (2/3)z} \\ &= -\frac{1}{3} z \sum_{n=0}^{\infty} \left(\frac{2}{3}\right)^n z^n \\ &= -\frac{1}{3} \sum_{n=-\infty}^{\infty} \left(\frac{2}{3}\right)^n \mu[n] z^{n+1} \quad | \quad \text{let } -m = n + 1 \\ &= -\frac{1}{3} \sum_{m=-\infty}^{\infty} \left(\frac{2}{3}\right)^{-m-1} \mu[-m-1] z^{-m} \end{aligned}$$

Thus, the inverse transform of  $H(z)$  is

$$h_{stable}[n] = 0.5 \cdot 0.5^n \mu[n] - \frac{1}{3} \left(\frac{2}{3}\right)^{-n-1} \mu[-n-1]$$

which is plotted in Figure 86(b). The impulse response is non-zero for indices  $n < 0$ , and the filter is not causal with criterion  $h[n] < 0, n < 0$ . The filter is stable.

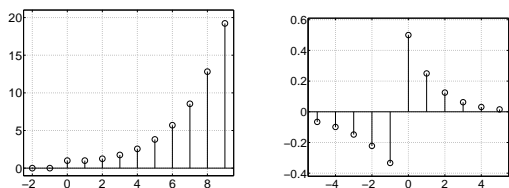


Figure 86: Problem 56: Left, 56(c) where ROC:  $|z| > 1.5 \Leftrightarrow$  filter is causal but not stable. Right, 56(d) where ROC:  $0.5 < |z| < 1.5 \Leftrightarrow$  filter is not causal but stable.

**Remark.** In practice, we operate with causal and stable filters, which means that all poles should be inside the unit circle.

58. **Problem:** Show that the filter  $H(z) = 1 - 2z^{-1} + 2z^{-2} - z^{-3}$  has linear phase response.

[L0510] **Solution:** If  $N$ -length  $h[n]$  is either symmetric ( $h[n] = h[N - 1 - n]$ ) or antisymmetric ( $h[n] = -h[N - 1 - n]$ ), then the filter has linear phase response (Mitra 2Ed Sec. 4.4.3 / 3Ed Sec. 5.5.2). At the same time, the locations of zeros are mirror-imaged with respect to the unit circle ( $r \cdot e^{\pm j\omega} \leftrightarrow (1/r) \cdot e^{\pm j\omega}$ ).

All possible linear-phase filters:

- Type 1: symmetric odd length,
- Type 2: symmetric even length,
- Type 3: antisymmetric odd length,
- Type 4: antisymmetric even length.

Zero-phase filter is a special case when the sequence is symmetric around origo, i.e., delay is zero and the filter is noncausal.

Some examples of four types of linear-phase filters are given in Figure 88.

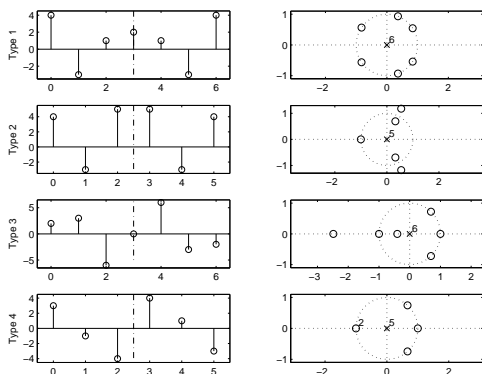


Figure 88: Problem 58: Examples of four types of linear-phase filters with impulse response  $h[n]$  in left and pole-zero-plot in right.

The inverse transform of  $H(z)$  is the impulse response  $h[n]$  of even length  $N = 4$

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

which is clearly antisymmetric around  $n_c = 1.5$ , see Figure 89(a).

We take the term  $e^{-jn_c\omega}$  as a common factor in frequency response

$$\begin{aligned} H(e^{j\omega}) &= e^{-j1.5\omega} \cdot (e^{j1.5\omega} - 2e^{j0.5\omega} + 2e^{-j0.5\omega} - e^{-j1.5\omega}) \\ &= e^{-j1.5\omega} \cdot ((e^{j1.5\omega} - e^{-j1.5\omega}) - 2(e^{j0.5\omega} - e^{-j0.5\omega})) \\ &= e^{-j1.5\omega} \cdot (2j \sin(1.5\omega) - 4j \sin(0.5\omega)) \end{aligned}$$

57. **Problem:** Magnitude specifications are normally expressed in normalized form. The maximum of the amplitude response is scaled to one, and the frequency axis is scaled up to half of the sampling frequency,  $0 \dots \pi$ . The first term of the denominator polynomial should also be 1.

Consider the following digital lowpass filter of type Chebyshev II:

$$H(z) = K \cdot \frac{0.71 - 0.36z^{-1} - 0.36z^{-2} + 0.71z^{-3}}{1 - 2.11z^{-1} + 1.58z^{-2} - 0.40z^{-3}}$$

Normalize the maximum of the amplitude response to the unity (0 dB).

[L0612] **Solution:** Chebyshev II approximation is monotonic in the passband, see Figure 87.

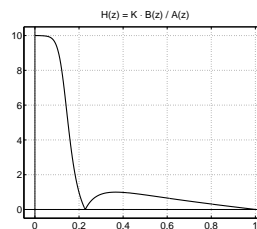


Figure 87: Problem 57,  $H(z) = K \cdot B(z)/A(z)$  without magnitude scaling.

Therefore the maximum value of the amplitude response of the lowpass Chebyshev II filter is at  $\omega = 0$ . The gain  $K$  can be computed also in  $z$ -plane using  $z = e^{j\omega}|_{\omega=0} = 1$ .

$$\begin{aligned} |H(z)| &= \left| K \frac{0.71 - 0.36z^{-1} - 0.36z^{-2} + 0.71z^{-3}}{1 - 2.11z^{-1} + 1.58z^{-2} - 0.40z^{-3}} \right| \\ |H(z)|_{z=1} &= \left| K \frac{0.71 - 0.36z^{-1} - 0.36z^{-2} + 0.71z^{-3}}{1 - 2.11z^{-1} + 1.58z^{-2} - 0.40z^{-3}} \right| = 1 \\ &= K \frac{0.70}{0.07} = 1 \\ \Rightarrow K &= 0.1 \end{aligned}$$

In a similar way if the maximum is at  $\omega = \pi$  (highpass), compute  $z = e^{j\omega}|_{\omega=\pi} = -1$ , etc.

**Remark.** When  $|H(z)|_{max} = 1$ , then the maximum reference level is in (power) desibels  $|H(z)|_{max} = 20 \log_{10}(1) = 0$  dB.

using Euler's formula  $e^{j\omega} = \cos(\omega) + j \sin(\omega)$ . Recall that  $\angle(A \cdot B) = \angle A + \angle B$ . When computing phase  $\angle H(e^{j\omega})$ , we get

$$\begin{aligned} \angle H(e^{j\omega}) &= \angle e^{-j1.5\omega} + \angle j + \angle(2 \sin(1.5\omega) - 4 \sin(0.5\omega)) \\ &= -1.5\omega + 0.5\pi + \angle C(\omega) \end{aligned}$$

where  $C(\omega) = 2 \sin(1.5\omega) - 4 \sin(0.5\omega)$ , and  $\angle C(\omega)$  gets values  $0$  or  $\pi$ , when  $C(\omega) > 0$  or  $C(\omega) < 0$ , respectively. Now  $\angle H(e^{j\omega})$  is piecewise linear function of  $\omega$ . The phase response is depicted in Figure 89(c).

Hence,  $H(e^{j\omega})$  is said to have linear phase response<sup>3</sup>. All frequency components are delayed by the same 1.5. Therefore the waveform is preserved as much as possible for a given amplitude response.

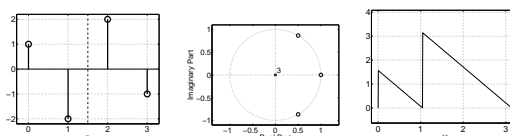


Figure 89: Problem 58: (a) One type of antisymmetric impulse response  $h[n]$ , (b) pole-zero plot, and (c) linear phase response  $\angle H(e^{j\omega})$ . Notice the points of discontinuity ("jumps") at  $0$  and  $\pm\pi/3$ , when there is a zero on the unit circle. This kind of discontinuities (by  $\pi$ ) are allowed in the definition of linear-phase filters.

<sup>3</sup>More precisely "linear phase interrupted by discontinuities" (see [http://ccrma.stanford.edu/~jos/mdft/Linear\\_Phase\\_Terms.html](http://ccrma.stanford.edu/~jos/mdft/Linear_Phase_Terms.html), referred 2007-03-16)

59. **Problem:** Examine the following five filters and connect them at least to one of the following categories (a) zero-phase, (b) linear-phase, (c) allpass, (d) minimum-phase, (e) maximum-phase.

$$\begin{aligned}
 h_1[n] &= -\delta[n+1] + 2\delta[n] - \delta[n-1] \\
 H_2(z) &= \frac{1+3z^{-1}+2.5z^{-2}}{1-0.5z^{-1}} \\
 y_3[n] &= 0.5y_3[n-1] + x[n] + 1.2x[n-1] + 0.4x[n-2] \\
 H_4(z) &= \frac{0.2-0.5z^{-1}+z^{-2}}{1-0.5z^{-1}+0.2z^{-2}} \\
 H_5(e^{j\omega}) &= -1 + 2e^{-j\omega} - e^{-2j\omega}
 \end{aligned}$$

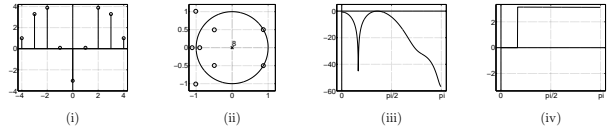


Figure 90: An example of a zero-phase transfer function in Problem 59. Subfigures (for Figures 90..95), (i) impulse response  $h[n]$ , (ii) pole-zero plot, (iii) amplitude response  $|H(e^{j\omega})|$ , x-axis  $(0 \dots \pi)$ , (iv) phase response  $\angle H(e^{j\omega})$ , x-axis  $(0 \dots \pi)$ .

[L0331] **Solution:** Types of transfer functions are explained in (*Mitra 2Ed Sec. 4.4, 4.6, 4.7, 4.8 / 3Ed Sec. 7.1, 7.2, 7.3*). After some work at least the following pairs can be mentioned: (a)  $h_1[n]$ , (b)  $H_5(e^{j\omega})$ , (c)  $H_4(e^{j\omega})$ , (d)  $y_3[n]$ , and (e)  $H_2(z)$ .

If the coefficients of the transfer function are real-valued (as they are in this course), then the pole and zero pairs must be complex conjugates:  $z_1 = re^{j\theta}$ ,  $z_2 = re^{-j\theta}$ .

If the coefficients of the FIR filter are symmetric, Type I, II, III, and IV, (*Mitra 2Ed Sec. 4.4.3, 4.4.4 / 3Ed Sec. 7.3*) and (*Mitra 2Ed Fig. 4.14, 4.16 / 3Ed Fig. -, 7.17*), then the filter has linear phase response (or even zero-phase). The group delay ( $\tau(\omega) = -d/d\omega \angle H(e^{j\omega})$ ) of linear-phase filters is constant for all frequencies.

Another important term is mirror-symmetry respect to the unit circle. In this case the connection between poles or zeros is:  $z_1 = re^{j\theta}$ ,  $z_2 = (1/r)e^{j\theta}$  (and their complex conjugates).

For each filter type there is also another example. There are four figures a row for each example, (i) impulse response, (ii) pole-zero-diagram, (iii) amplitude response in decibels and x-axis in range  $0 \dots \pi$ , (iv) phase response.

h1) This noncausal FIR filter has zero phase. The impulse response  $h_1 = -\delta[n+1] + 2\delta[n] - \delta[n-1]$  is symmetric around the origo in the time-domain. The frequency response can be written

$$\begin{aligned}
 H_1(e^{j\omega}) &= -e^{j\omega} + 2 - e^{-j\omega} = 2 - 2\cos(\omega) \\
 |H_1(e^{j\omega})| &= |2 - 2\cos(\omega)| \geq 0 \quad | \text{ ampl.resp. } \in \mathbf{R} \\
 \angle H_1(e^{j\omega}) &= 0 \quad | \text{ phase resp.} \\
 -\frac{d}{d\omega} \angle H_1(e^{j\omega}) &= 0 \quad | \text{ no delay at all}
 \end{aligned}$$

from which it can be seen that  $H_1(e^{j\omega})$  is real-valued. The phase response and group delay ( $\tau(\omega) = -d/d\omega \angle H(e^{j\omega})$ ) is therefore zero (or 180 degrees for negative values of  $H(e^{j\omega})$ ) for all frequencies, in other words, the filter is zero-phase (*Mitra 2Ed Sec. 4.4.2 / 3Ed Sec. 7.2.1*) and the signal is not delayed in the filter. Matlab command `filtfilt` can be applied instead of `filter`.

Another example, see Figure 90. The zeros are situated mirror-symmetrically according to the unit circle, and the impulse response and the transfer function are

$$\begin{aligned}
 h_2[n] &= \{1, 3.2893, 3.8875, 0.0884, -3.0407, 0.0884, 3.8875, 3.2893, 1\} \\
 H(z) &= \sum_n h_2[n]z^{-n}
 \end{aligned}$$

H2) When all zeros are outside the unit circle, the filter has maximum phase. The filter is IIR, the two zeros are outside the unit circle. When plotting the amplitude response, it can be noticed that the filter is lowpass (LP). The filter  $H_2(z)$  is at least maximum-phase.

Another example on a maximum-phase transfer function (*Mitra 2Ed Sec. 4.7 / 3Ed Sec. 7.2.3*), whose all zeros lie outside the unit circle in Figure 91

$$H(z) = \frac{1 - 2.773z^{-1} + 3.108z^{-2} - 3.125z^{-3}}{1 + 1.559z^{-1} + 0.81z^{-2}}$$

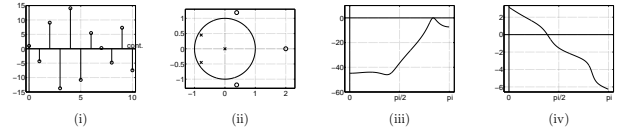


Figure 91: An example of a maximum-phase transfer function in Problem 59.

y3) When all zeros are inside the unit circle, the filter has minimum phase. From the difference equation we get

$$H_3(z) = \frac{1 + 1.2z^{-1} + 0.4z^{-2}}{1 - 0.5z^{-1}}$$

The transfer function is similar to  $H_2(z)$ , but the zeros are now mirror-symmetric to those. Therefore the amplitude response is the same, but the filter is minimum-phase (*Mitra 2Ed Sec. 4.7 / 3Ed Sec. 7.2.3*).

Another example on a minimum-phase transfer function whose all zeros lie inside the unit circle in Figure 92

$$H(z) = \frac{1 - 0.9944z^{-1} + 0.8872z^{-2} - 0.32z^{-3}}{1 + 1.559z^{-1} + 0.81z^{-2}}$$

A minimum-phase transfer function can be converted to a maximum-phase transfer function (or vice versa) by mirroring the zeros respect to the unit circle. This can be done using an appropriate allpass function.

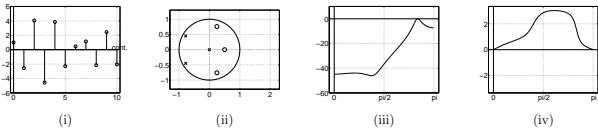


Figure 92: An example of a minimum-phase transfer function in Problem 59.

H4) If the amplitude response ( $z \leftarrow e^{j\omega}$ ) is  $|H(e^{j\omega})| = 1$  for all frequencies, then the filter is allpass (*Mitra 2Ed Sec. 4.6 / 3Ed Sec. 7.1.3*). The phase response differs from filter to filter. Allpass-filters contain both zeros and poles mirror-symmetrically, and there is a certain symmetry in the coefficients of numerator and denominator polynomials, too. Note that gain cannot be seen from the pole-zero plot.

In Figure 93 an allpass transfer function

$$H(z) = -3.4722 \cdot \frac{-0.288 + 0.4785z^{-1} - 0.007771z^{-2} - 0.09443z^{-3} + z^{-4}}{1 - 0.09443z^{-1} - 0.007771z^{-2} + 0.4785z^{-3} - 0.288z^{-4}}$$

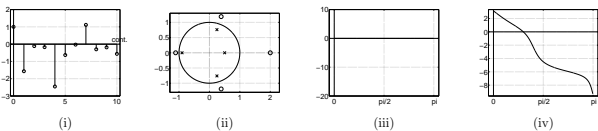


Figure 93: An example of an allpass transfer function in Problem 59.

**Remark.** A complementary transfer function (*Mitra 2Ed Sec. 4.8 / 3Ed Sec. 7.5*) can be obtained using allpass filters. An example of a lowpass filter

$$\begin{aligned}
 H_{LP}(z) &= 0.5(A_0(z) + A_1(z)) \\
 &= 0.5\left(1 + \frac{-a + z^{-1}}{1 - az^{-1}}\right) \\
 &= 0.5\left(\frac{1 - a + z^{-1} - az^{-1}}{1 - az^{-1}}\right)
 \end{aligned}$$

where  $A_0(z)$  and  $A_1(z)$  are allpass transfer functions and its power-complementary highpass filter

$$\begin{aligned}
 H_{HP}(z) &= 0.5(A_0(z) - A_1(z)) \\
 &= 0.5\left(1 - \frac{-a + z^{-1}}{1 - az^{-1}}\right) \\
 &= \frac{1+a}{2} \cdot \frac{1-z^{-1}}{1-az^{-1}}
 \end{aligned}$$

In Figure 94(iii) is shown that  $|H_{LP}(z)|^2 + |H_{HP}(z)|^2 = 1$ , as expected by the definition of power-complementary transfer functions.

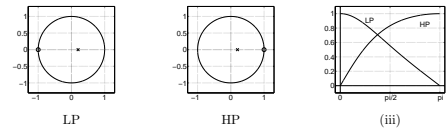


Figure 94: An example of power-complementary LP and HP filters in Problem 59.

H5) Linear-phase. This impulse response is a shifted (delayed) version of  $h_1[n]$ . The frequency response is not any more real-valued, but still the phase response is linear and the group delay constant.

$$\begin{aligned}
 H_5(e^{j\omega}) &= e^{-j\omega} \cdot H_1(e^{j\omega}) \\
 |H_5(e^{j\omega})| &= |H_1(e^{j\omega})| = |2 - 2\cos(\omega)| \\
 \angle H_5(e^{j\omega}) &= -\omega \quad | \text{ linear} \\
 -\frac{d}{d\omega} \angle H_5(e^{j\omega}) &= 1 \quad | \text{ constant}
 \end{aligned}$$

There are four types of linear-phase transfer functions (*Mitra 2Ed Sec. 4.4.3 / 3Ed Sec. 7.3*). Impulse response of Type 1 is symmetric and odd-length. Type 2 is symmetric and even-length. Type 3 is antisymmetric and odd-length. Type 4 is antisymmetric and even-length. The zeros have mirror-image symmetry respect to the unit circle.

In Figure 95 there is a Type 1 (length: 9, order: 8) impulse response, which is a shifted version of the filter in Figure 90.

$$\begin{aligned}
 h_5[n] &= \{1, 3.2893, 3.8875, 0.0884, -3.0407, 0.0884, 3.8875, 3.2893, 1\} \\
 H(z) &= \sum_n h_5[n]z^{-n}
 \end{aligned}$$

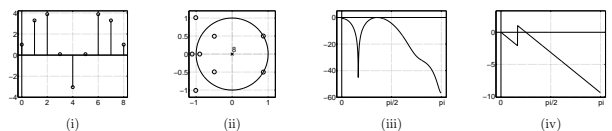


Figure 95: An example of a linear-phase transfer function in Problem 59.

60. **Problem:** A second-order FIR filter  $H_1(z)$  has exactly at  $z = 2 \pm j$ .

- a) Derive a minimum-phase FIR filter with exactly same amplitude response (Mitra 2Ed Sec. 4.7, p. 246 / 3Ed Sec. 7.2.3, p. 365).
- b) Derive an inverse filter of the minimum-phase FIR filter computed in (a) (Mitra 2Ed Sec. 4.9, p. 253 / 3Ed Sec. 7.6, p. 396).

[L0332] **Solution:** Minimum-phase filter has all zeros inside the unit circle whereas maximum-phase filter has all zeros outside the unit circle. A filter with zeros inside and outside the unit circle is often called a mixed-phase filter (Mitra 2Ed Sec. 4.7, p. 246 / 3Ed Sec. 7.2.3, p. 365).

Two causal LTI discrete-time systems with impulse responses  $h_1[n]$  and  $h_2[n]$  are inverses if  $h_1[n] \otimes h_2[n] = \delta[n]$ . After  $z$ -transform  $H_1(z) \cdot H_2(z) = 1$ , or  $H_1(z) = 1/H_2(z)$ . If  $H_1(z) = B(z)/A(z)$ , then  $H_2(z) = A(z)/B(z)$ , that is, all zeros are replaced by poles, and vice versa. If the filter is minimum-phase FIR with all zeros inside the unit circle, then its inverse is stable. Inverse filtering can be used, e.g., in recovering a signal which has been distorted in an imperfect transmission channel (Mitra 2Ed Sec. 4.9, p. 253 / 3Ed Sec. 7.6, p. 396). In the case of non-minimum-phase FIR filter the situation is more complex (Mitra 2Ed Sec. -, p. - / 3Ed Sec. 7.6.2, p. 398).

- a) A second-order FIR filter  $H_1(z)$  with zeros at  $z = 2 \pm j$  is drawn in Figure 96(a). It is clearly a high-pass filter. Its transfer function can be written

$$H_1(z) = (1 - (2 + j)z^{-1}) \cdot (1 + (2 - j)z^{-1}) = 1 - 4z^{-1} + 5z^{-2}$$

Consider a stable noncausal all-pass filter in Figure 96(b)

$$A(z) = \frac{5 - 4z^{-1} + z^{-2}}{1 - 4z^{-1} + 5z^{-2}} \quad \text{ROC: } |z| < \sqrt{5}$$

which has poles at  $p = 2 \pm j$  and zeros at  $z = 0.4 \pm 0.2j$ , and  $|A(z)| \equiv 1$  for all frequencies. Poles and zeros are mirror-images  $p_i = re^{j\theta}$ , and  $z_i = (1/r)e^{j\theta}$ . Note also that the inverse of a pole gives a complex conjugate of a mirrored zero,  $\frac{1}{2+j} = \frac{2-j}{(2+j)(2-j)} = \frac{2-j}{4+2j-2j+1} = 0.4 - 0.2j$ . Now, the minimum-phase FIR filter  $H_2(z)$  is received by  $H_2(z) = H_1(z) \cdot A(z)$ , see Figure 96(c)

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

The amplitude response of  $H_2(z)$  is exactly the same as that of  $H_1(z)$ , because  $|H_2(z)| = |H_1(z)| \cdot |A(z)|$ , where  $|A(z)| = 1$  by definition of all-pass filter. The phase responses of  $H_1(z)$  and  $H_2(z)$  differ.  $H_2(z)$  has two zeros at  $z = 0.4 \pm 0.2j$ .

- b) The inverse filter is now received directly  $H_3(z) = 1/H_2(z)$

$$H_3(z) = \frac{1}{5 - 4z^{-1} + z^{-2}} = \frac{0.2}{1 - 0.8z^{-1} + 0.2z^{-2}} \quad \text{ROC: } |z| > \sqrt{0.2}$$

$H_3(z)$  is a stable lowpass all-pole filter with poles at  $p = 0.4 \pm 0.2j$ , see Figure 96(d).

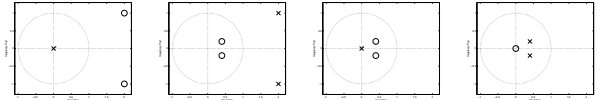


Figure 96: Problem 60: (a)  $H_1(z)$ , (b)  $A(z)$ , (c)  $H_2(z) = H_1(z) \cdot A(z)$ , (d)  $H_3(z) = 1/H_2(z)$ .

61. **Problem:** Consider a stable and causal discrete-time LTI system  $S_1$ , whose zeros  $z_i$  and poles  $p_i$  are

$$\begin{aligned} \text{zeros: } & z_1 = 1, \quad z_2 = 1 \\ \text{poles: } & p_1 = 0.18, \quad p_2 = 0 \end{aligned}$$

Add a LTI FIR filter  $S_2$  in parallel with  $S_1$  as shown in Figure 97 so that the whole system  $S$  is causal second-order bandstop filter, whose minimum is approximately at  $\omega \approx \pi/2$  and whose maximum is scaled to one. What are transfer functions  $S_2$  and  $S$ ?

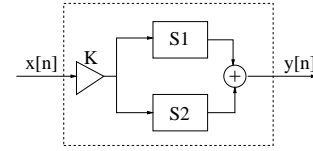


Figure 97: Problem 61: Filter  $S$  constructed from LTI subsystems  $S_1$  and  $S_2$ .

[L0345] **Solution:** Denote transfer functions of the system  $S_1$  by  $H_1(z) = B_1(z)/A_1(z)$ ,  $S_2$  by  $H_2(z) = B_2(z)/A_2(z)$ , and the total system  $S$  by  $H(z) = K \cdot B(z)/A(z)$ . The system  $S_2$  is FIR, so  $A_2(z) = 1$ , and therefore  $H_2(z) = B_2(z)$ . The subsystems are parallel which gives

$$H(z) = K \cdot (H_1(z) + H_2(z)) = K \cdot \left( \frac{B_1(z)}{A_1(z)} + B_2(z) \right) = K \cdot \frac{B(z)}{A(z)}$$

The system  $S_1$  is clearly a high-pass filter, see Figure 98(a),(b).

$$H_1(z) = \frac{(1 - z_1z^{-1}) \cdot (1 - z_2z^{-1})}{1 - p_1z^{-1}} = \frac{(1 - z^{-1}) \cdot (1 - z^{-1})}{1 - 0.18z^{-1}} = \frac{1 - 2z^{-1} + z^{-2}}{1 - 0.18z^{-1}}$$

We would like to have a bandstop filter  $H(z)$  whose minimum is approximately at  $\omega_m = \pi/2$ . Zeros do not necessarily need to lie on the unit circle, but with the angle  $\omega_m$  and  $-\omega_m$ , see Figure 98(c),(d). The numerator polynomial of  $H(z)$ , i.e.,  $B(z)$  is of form

$$B(z) = (1 - rjz^{-1}) \cdot (1 + rjz^{-1}) = 1 + r^2z^{-2}$$

Next we will compute two different solutions for bandstop filters. In the first case, zeros lie inside the unit circle ( $r < 1$ ), and in the second case they are on the unit circle ( $r = 1$ ).

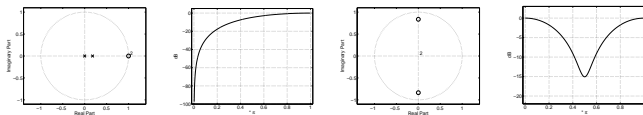


Figure 98: Problem 61: (a),(b) Known  $H_1(z)$ , (c),(d)  $B(z)$  of the whole system.

Adding  $H_1(z)$  and  $H_2(z)$  results to  $(H'(z))$  is  $H(z)$  without scaling factor  $K$

$$\begin{aligned} H'(z) &= \frac{B_1(z)}{A_1(z)} + B_2(z) \\ &= \frac{B_1(z)}{A_1(z)} + \frac{A_1(z) \cdot B_2(z)}{A_1(z)} \\ &= \frac{(1 - 2z^{-1} + z^{-2}) + (1 - 0.18z^{-1}) \cdot B_2(z)}{1 - 0.18z^{-1}} \end{aligned}$$

It can be seen that the order of  $B_2(z)$  cannot be more than 1 because  $B(z)$  has to be second-order at most. We can write  $B_2(z) = a + bz^{-1}$ , and

$$\begin{aligned} H'(z) &= \frac{(1 - 2z^{-1} + z^{-2}) + (1 - 0.18z^{-1}) \cdot (a + bz^{-1})}{1 - 0.18z^{-1}} \\ &= \frac{(1 + a) + (-2 - 0.18a + b)z^{-1} + (1 - 0.18b)z^{-2}}{1 - 0.18z^{-1}} \end{aligned}$$

Now we can simply choose  $a = 0$  and  $b = 2$ , i.e.,  $B_2(z) = 2z^{-1}$ , in order to get a required form of  $B(z) = 1 + 0.64z^{-2}$ . In this case zeros are at  $z_1 = 0.8j$  and  $z_2 = -0.8j$ . The pole-zero plot and the (scaled) magnitude response of

$$H'(z) = \frac{1 + 0.64z^{-2}}{1 - 0.18z^{-1}}$$

are given in Figure 99(a),(b).

The only pole lies at  $p_1 = 0.18$ , which is closer to 1 than  $-1$ , and the maximum is therefore at  $\omega = 0$  ( $z = 1$ ). The scaling constant  $K$ :

$$|H(z=1)| = K \cdot \frac{|1 + 0.64|}{|1 - 0.18|} = 2K = 1$$

which gives  $K = 0.5$  and the final results:

$$\begin{aligned} H_2(z) &= 2z^{-1} \\ H(z) &= 0.5 \cdot \frac{1 + 0.64z^{-2}}{1 - 0.18z^{-1}} \end{aligned}$$

Another solution is to compute other values for  $a$  and  $b$ . By demanding  $B(z = j) = 0$  and  $B(z = -j) = 0$ , i.e., zeros on the unit circle, we get the following two equations with two unknowns. Note that  $(1/j) = -j$ .

$$\begin{aligned} B(z = j) &= (1 + a) + (-2 - 0.18a + b)(-j) + (1 - 0.18b)(-1) = 0 \\ B(z = -j) &= (1 + a) + (-2 - 0.18a + b)(j) + (1 - 0.18b)(-1) = 0 \end{aligned}$$

$$\begin{aligned} 1 + a - 1 + 0.18b &= 0 \quad | \text{real part} \\ -2 - 0.18a + b &= 0 \quad | \text{imaginary part} \end{aligned}$$

Computing the unknowns gives  $a \approx -0.35$  and  $b \approx 1.94$ , leading to  $B_2(z) = -0.35 + 1.94z^{-1}$  and

$$H'(z) = 0.65 \cdot \frac{1 + z^{-2}}{1 - 0.18z^{-1}}$$

whose pole-zero plot and (scaled) magnitude response are plotted in Figure 99(c),(d).

Now the scaling constant  $K$  is:

$$|H(z=1)| = K \cdot 0.65 \cdot \frac{|1+1|}{|1-0.18|} \approx 1.59K = 1$$

which gives  $K = 0.63$  and the final results:

$$\begin{aligned} H_2(z) &= -0.35 + 1.94z^{-1} \\ H(z) &= 0.41 \cdot \frac{1 + z^{-2}}{1 - 0.18z^{-1}} \end{aligned}$$

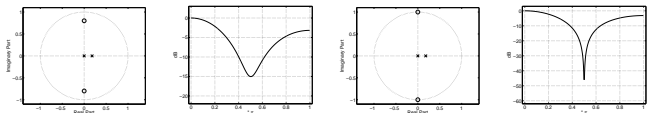


Figure 99: Problem 61: (a),(b)  $H(z)$  of first solution, (c),(d)  $H(z)$  of second solution.

62. **Problem:** Derive the transfer function of the feedback system shown in Figure 100.

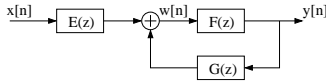


Figure 100: System in Problem 62.

[L0500] **Solution:** Systems in parallel, see Figure 101:  $H_p(z) = H_1(z) + H_2(z)$  in frequency domain and  $h_p[n] = h_1[n] + h_2[n]$  in time domain. Systems in cascade, see Figure 102:

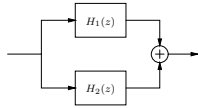


Figure 101: Systems in parallel, example in Problem 62.

$H_c(z) = H_1(z)H_2(z)$  in frequency domain and  $h_c[n] = h_1[n] \otimes h_2[n]$  in time domain. The flow diagram of the system being investigated with temporary variable  $w[n]$  is in

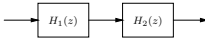


Figure 102: Systems in cascade, example in Problem 62.

Figure 103. We get the following equations:

$$\begin{cases} Y(z) = F(z)W(z) \\ W(z) = E(z)X(z) + G(z)Y(z) \end{cases}$$

$$\begin{aligned} Y(z) &= F(z)(E(z)X(z) + G(z)Y(z)) \\ Y(z)(1 - F(z)G(z)) &= (F(z)E(z))X(z) \end{aligned}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{F(z)E(z)}{1 - F(z)G(z)}$$

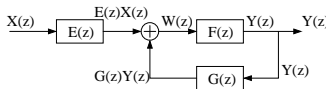


Figure 103: System in Problem 62.

63. **Problem:** The filter in Figure 104 is in canonic direct form II (DF II). Draw it in DF I. What is the transfer function  $H(z)$ ?

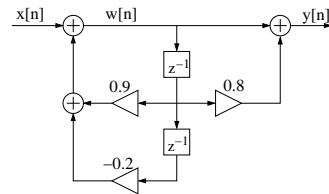


Figure 104: The block diagram of direct form II of Problem 63.

[L0508] **Solution:** Direct form structure means that the coefficients of the block diagram are the same (or negative values) as in the difference equation and transfer function. There are also other structures, e.g. lattice. The transfer function for any direct form (I, II, and transposes  $I_T, II_T$ , respectively, see Page 120) is the same. Some differences (may) occur when working with finite word length. There are also differences in computational load and memory storage.

a) The block diagram in Figure 105 is in canonic direct form II.

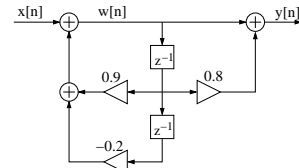


Figure 105: The block diagram of direct form II in Problem 63.

If we want to convert it into direct form I without any calculations (done below in (b)), we can duplicate the registers. The same signal  $w[n]$  goes into the both branches. See Figure 106(a).

Then we can denote the part in left as an "IIR subsystem" and the structure in right as an "FIR subsystem". Because both of them are LTI, we can change the order of them, as in any LTI system, for example, using impulse responses

$$h[n] = h_{IIR}[n] \otimes h_{FIR}[n] \equiv h_{FIR}[n] \otimes h_{IIR}[n]$$

Now we have direct form I in Figure 106(b), and the difference equation and the transfer function can be obtained directly without any temporal variables! However, there are now three registers instead of two.

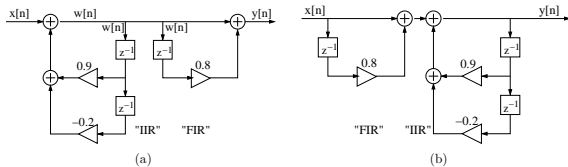


Figure 106: From direct form II to direct form I in Problem 63(a).

b) The transfer function and difference equation can be derived directly from the filter in Figure 105:

$$\begin{aligned} y[n] &= w[n] + 0.8w[n-1] \\ w[n] &= x[n] + 0.9w[n-1] - 0.2w[n-2] \end{aligned}$$

Using z-transform

$$\begin{aligned} Y(z) &= W(z) + 0.8z^{-1}W(z) = W(z)(1 + 0.8z^{-1}) \\ W(z) &= X(z) + 0.9z^{-1}W(z) - 0.2z^{-2}W(z) \end{aligned}$$

From the latter one,  $W(z) = X(z)/(1 - 0.9z^{-1} + 0.2z^{-2})$ , and substituting into the first one, we get

$$\begin{aligned} Y(z) &= X(z) \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \\ H(z) = Y(z)/X(z) &= \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \end{aligned}$$

Using inverse z-transform we get difference equation which can be easily drawn as direct form I block diagram:

$$\begin{aligned} Y(z)/X(z) &= \frac{1 + 0.8z^{-1}}{1 - 0.9z^{-1} + 0.2z^{-2}} \\ Y(z)(1 - 0.9z^{-1} + 0.2z^{-2}) &= X(z)(1 + 0.8z^{-1}) \\ y[n] - 0.9y[n-1] + 0.2y[n-2] &= x[n] + 0.8x[n-1] \end{aligned}$$

**Remark.** Direct Forms.

(Mitra 2Ed Sec. 6.4.1 / 3Ed Sec. 8.4.1) Direct form: coefficients of difference equation or transfer function can be found in block diagram. (This is not the case, for example, in lattice form.) Common in all forms is that they have the same transfer function, but the "implementation" is different.

Let the transfer function be

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + 0.5z^{-1}}{1 - 0.2z^{-1} + 0.4z^{-2}}$$

In the top numerator polynomial  $1 + 0.5z^{-1}$  refers to "FIR part"  $P(z)$  and in the bottom denominator polynomial  $1 - 0.2z^{-1} + 0.4z^{-2}$  "IIR part"  $D(z)$ :

$$H(z) = P(z) \frac{1}{D(z)}$$

How to get difference equation and block diagram from transfer function, z-transform  $ax[n - n_0] \leftrightarrow a z^{-n_0} X(e^{j\omega})$ :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + 0.5z^{-1}}{1 - 0.2z^{-1} + 0.4z^{-2}}$$

$$Y(z) = \frac{X(z)[1 + 0.5z^{-1}]}{1 - 0.2z^{-1} + 0.4z^{-2}}$$

$$\begin{aligned} Y(z)[1 - 0.2z^{-1} + 0.4z^{-2}] &= X(z)[1 + 0.5z^{-1}] \\ Y(z) - 0.2z^{-1}Y(z) + 0.4z^{-2}Y(z) &= X(z) + 0.5z^{-1}X(z) \\ y[n] - 0.2y[n-1] + 0.4y[n-2] &= x[n] + 0.5x[n-1] \\ y[n] &= 0.2y[n-1] - 0.4y[n-2] + x[n] + 0.5x[n-1] \end{aligned}$$

**Direct form I** can be drawn directly  $H(z) = P(z) \cdot \frac{1}{D(z)}$ , first "FIR" and then "IIR" (Figure 107).

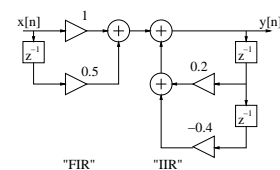


Figure 107: Direct form I. You may connect FIR and IIR parts in the middle sum line.

**When transposing** (Figure 108) transfer function stays, but structure changes. "Rules" for transposing:

- 1 Change directions
- 2 Nodes to sums
- 3 Sums to nodes
- 4 Flip the whole structure

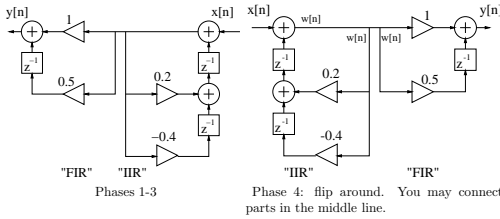


Figure 108: Transposed direct form I.

**Direct form II** contains minimum number of delay registers. Draw in order “IIR” and then “FIR”. Think the transfer function in order  $H(z) = \frac{1}{D(z)} \cdot P(z)$ . Because LTI, the order of subfilters can be changed. Connect the delay registers, because there are the same signals (see Book). So you get **canonic** form, where the number of delays is the same as order of the filter (Figure 109).

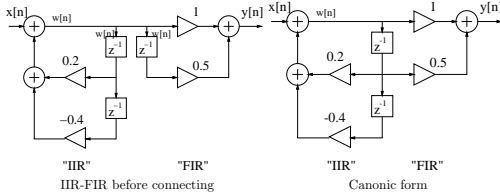


Figure 109: Direct form II.

Corresponding transposing  $II_T$ , see Figure 110.

**Example** on direct form, cascade and parallel system. Consider a second order transfer function

$$H(z) = \frac{1}{(1 + \frac{1}{3}z^{-1})(1 - \frac{1}{4}z^{-1})} = \frac{1}{1 + \frac{1}{12}z^{-1} - \frac{1}{12}z^{-2}}$$

with difference equation

$$y[n] = -\frac{1}{12}y[n-1] + \frac{1}{12}y[n-2] + x[n]$$

Cascade form can be written as

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

and parallel form using partial fraction (draw!)

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

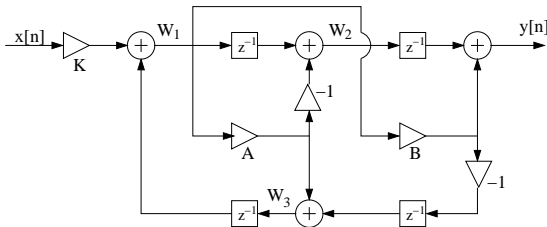


Figure 112: The filter with temporary signals  $w_1, w_2$ , and  $w_3$  in Problem 64.

Next, substituting second line in fourth line we get

$$Y = [z^{-1}(z^{-1} - A) + B]W_1$$

Finally, we get rid of the last temporary variable  $W_1$ , and get

$$H(z) = \frac{Y(z)}{X(z)} = K \cdot \frac{B - Az^{-1} + z^{-2}}{1 - Az^{-1} + Bz^{-2}}$$

- It is LTI. There are only multiplications by constants, delays, and sums of sequences.
- Since the structure employs 4 unit delays to implement a second-order transfer function, it is not canonic.

**Canonic structure:** the number of registers, i.e. delay components, is the same as the filter order. Direct form I is not canonic, but it is intuitive and its difference equation is easy to obtain. Direct form II is canonic. It is more efficient to use canonic structures. (Consider, for example, Problem 71. If canonic structure is used, there are only 8 storage locations instead of 10.)

- 

$$\begin{aligned} H(z)H(z^{-1}) &= K^2 \left(\frac{B - Az^{-1} + z^{-2}}{1 - Az^{-1} + Bz^{-2}}\right) \left(\frac{B - Az^1 + z^2}{1 - Az^1 + Bz^2}\right) \Big|_{z \rightarrow z^{-2}} \\ &= K^2 \left(\frac{B - Az^{-1} + z^{-2}}{1 - Az^{-1} + Bz^{-2}}\right) \left(\frac{Bz^{-2} - Az^{-1} + 1}{z^{-2} - Az^{-1} + B}\right) \\ &= K^2 \end{aligned}$$

Therefore  $|H(e^{j\omega})| = K$  for all values of  $\omega$  and hence  $|H(e^{j\omega})| = 1$  if  $K = 1$ .  $H(z)$  is an allpass transfer if  $K = 1$ .

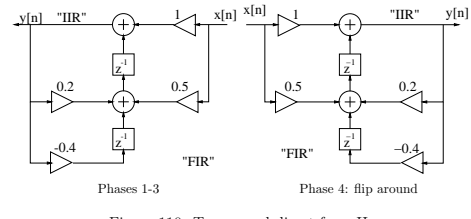


Figure 110: Transposed direct form II.

64. **Problem:** Analyze the digital filter structure shown in Figure 111 and determine its transfer function  $H(z) = Y(z)/X(z)$ .

- Is the system LTI?
- Is the structure canonic with respect to delays?
- Compute  $H(z)H(z^{-1})$  (the squared amplitude response). What is the type of this filter (lowpass/highpass/bandpass/bandstop/allpass)?

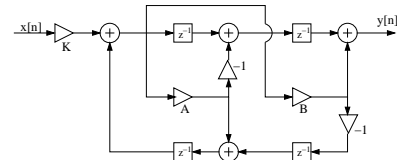


Figure 111: The flow diagram of the system in Problem 64.

[L0502] **Solution:** Let us use three temporary signals  $w_1[n], w_2[n]$ , and  $w_3[n]$ , in the following locations in Figure 112.

From the figure we get the following expressions in transform-domain ( $W \equiv W(z)$ ):

$$\begin{aligned} W_1 &= KX + z^{-1}W_3 \\ W_2 &= (z^{-1} - A)W_1 \\ W_3 &= AW_1 - Bz^{-1}W_1 = (A - Bz^{-1})W_1 \\ Y &= z^{-1}W_2 + BW_1 \end{aligned}$$

Substituting the equation from third line to first line we get

$$\begin{aligned} W_1 &= KX + z^{-1}(A - Bz^{-1})W_1 \\ (1 - Az^{-1} + Bz^{-2})W_1 &= KX \end{aligned}$$

65. **Problem:** Develop a polyphase realization of a length-9 FIR transfer function given by

$$H(z) = \sum_{n=0}^8 h[n]z^{-n}$$

with (a) 2 branches and (b) 4 branches.

[L0501] **Solution:** Polyphase realizations (Mitra 2Ed Sec. 6.3.3 / 3Ed Sec. 8.3.3) can be used in multirate techniques.

- Two branches

$$\begin{aligned} H(z) &= \sum_{n=0}^8 h[n]z^{-n} \\ &= h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + \\ &\quad h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8} \\ &= (h[0] + h[2]z^{-2} + h[4]z^{-4} + h[6]z^{-6} + h[8]z^{-8}) + \\ &\quad z^{-1}(h[1] + h[3]z^{-2} + h[5]z^{-4} + h[7]z^{-6}) \\ &= H_0(z^2) + z^{-1}H_1(z^2) \end{aligned}$$

where

$$\begin{aligned} H_0(z) &= h[0] + h[2]z^{-1} + h[4]z^{-2} + h[6]z^{-3} + h[8]z^{-4} \\ H_1(z) &= h[1] + h[3]z^{-1} + h[5]z^{-2} + h[7]z^{-3} \end{aligned}$$

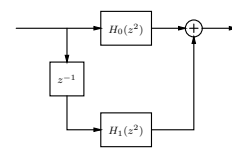


Figure 113: Polyphase realization with two branches in Problem 65(a).

- Four branches

$$\begin{aligned} H(z) &= \sum_{n=0}^8 h[n]z^{-n} \\ &= h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + \\ &\quad h[5]z^{-5} + h[6]z^{-6} + h[7]z^{-7} + h[8]z^{-8} \\ &= (h[0] + h[4]z^{-4} + h[8]z^{-8}) + z^{-1}(h[1] + h[5]z^{-4}) + \\ &\quad z^{-2}(h[2] + h[6]z^{-4}) + z^{-3}(h[3] + h[7]z^{-4}) \\ &= H_0(z^4) + z^{-1}H_1(z^4) + z^{-2}H_2(z^4) + z^{-3}H_3(z^4) \end{aligned}$$

where

$$\begin{aligned}
 H_0(z) &= h[0] + h[4]z^{-1} + h[8]z^{-2} \\
 H_1(z) &= h[1] + h[5]z^{-1} \\
 H_2(z) &= h[2] + h[6]z^{-1} \\
 H_3(z) &= h[3] + h[7]z^{-1}
 \end{aligned}$$

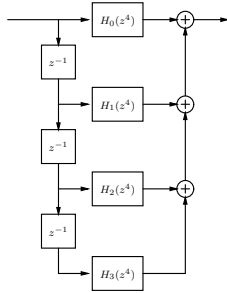


Figure 114: Polyphase realization with four branches in Problem 65(b).

66. **Problem:** Sketch the following specifications of a digital filter on paper. Which of the amplitude responses of the realizations in Figure 115 do fulfill the specifications?

Specifications: Digital lowpass filter, sampling frequency  $f_T$  8000 Hz, passband edge frequency  $f_p$  1000 Hz, transition band 500 Hz (transition band is the band between passband and stopband edge frequencies!), maximum passband attenuation 3 dB, minimum stopband attenuation 40 dB.

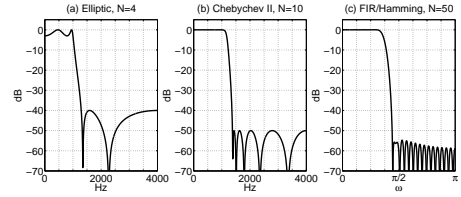


Figure 115: Three realizations in Problem 66: amplitude responses of (a) 4th order elliptic, (b) 10th order Chebyshev II, (c) 50th order FIR using Hamming window.

[L0644] **Solution:** The frequency specifications are in Hertz, radians, and in normalized Matlab frequency in Table 8 and they are drawn in Figure 116 with dashed line.

sampling frequency	$f_T$	8000 Hz	$\omega_T$	$2\pi$ (rad)	$\frac{2}{\pi}$
passband edge	$f_p$	1000 Hz	$\omega_p$	$\pi/4$ (rad)	$\frac{1}{2}$
stopband edge	$f_s$	1500 Hz	$\omega_s$	$3\pi/8$ (rad)	$\frac{3}{4}$
passband ripple	Rp	3 dB			Rp
stopband attenuation	Rs	40 dB			Rs

Table 8: Specifications for the filter in Problem 66.

Now that specifications are written and sketched, the filter order and the filter coefficients are computed using a specific software (e.g. Matlab, `ellipord` and `ellip`, `butterd` and `butter`, etc.). Then the amplitude response  $|H(e^{j\omega})|$  of the calculated filter is plotted in the same picture as the sketch of the specifications (e.g. Matlab, `[...] = freqz(B,A,...)`). If the amplitude response curve fits in the specifications, we have succeeded. In other case, the specifications and the code for the filter are re-checked.

The elliptic IIR filter in Figure 116(a) (via bilinear transform) is of order 4 and it fulfills the specifications exactly.

Chebyshev II filter (Figure 116(b)), which is 10th order IIR, is monotonic in passband and has stopband attenuation of 50 dB instead of 40. The amplitude response fits in the allowed area, and it is already too strict. Probably the order  $N = 8$  would be sufficient.

The third filter (Figure 116(c)) is 50th order FIR, whose transition is narrow enough but at the wrong cut-off frequency. So, this is the only filter, which does not fulfill the specifications. One should check the cut-off frequency so that the amplitude response fits.

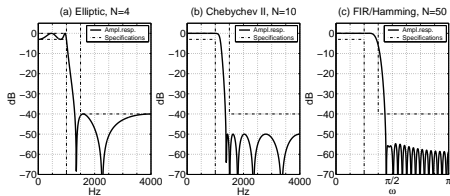


Figure 116: Three realizations in Problem 66: amplitude responses (solid line) with specifications (dashed line) of (a) 4th order elliptic (OK!), (b) 10th order Chebyshev II (OK, too tight realization?), (c) 50th order FIR using Hamming window (bad cut-off frequency).

Approximation	M 2Ed Sec.	M 3Ed Sec.	Response
Butterworth	5.4.2	4.4.2	$ H_a(j\Omega) ^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}}$
Chebyshev I	5.4.3	4.4.3	$ H_a(j\Omega) ^2 = \frac{1}{1+\epsilon^2 T_N^2(\Omega/\Omega_p)}$
Chebyshev II	5.4.3	4.4.3	$ H_a(j\Omega) ^2 = \frac{1}{1+\epsilon^2 [T_N(\Omega_s/\Omega_p)]^2}$
Elliptic	5.4.4	4.4.4	$ H_a(j\Omega) ^2 = \frac{1}{1+\epsilon^2 R_N^2(\Omega/\Omega_p)}$

Table 9: Analog filter approximations in Problem 67.

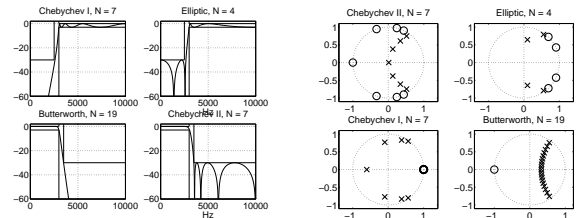


Figure 118: Problem 67, see the titles of each subfigure for filter type and order.

67. **Problem:** Connect first each amplitude response to the corresponding pole-zero plot in Figure 117. Then recognize the following digital IIR filter algorithms: Butterworth, Chebyshev I, Chebyshev II, Elliptic. The conversion from analog to digital form is done using bilinear transform. The sampling frequency in figures is 20 kHz.

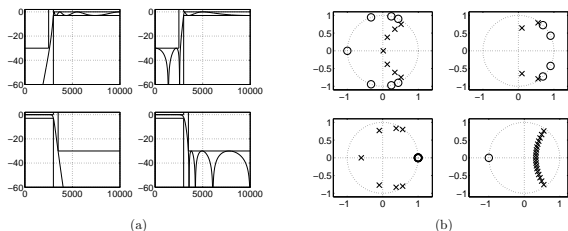


Figure 117: Problem 67. Digital filters from analog approximations through bilinear transform, (a) amplitude responses with specifications,  $f_s = 20000$  Hz (b) pole-zero plots.

[L0643] **Solution:** Analog filter design is represented in (Mitra 2Ed Sec. 5.4 / 3Ed Sec. 4.4). The approximations are given with magnitude-squared responses of Nth order in Table 9.

The response of Butterworth is monotonic. Chebyshev I is equiripple in the passband and monotonic in the stopband whereas Chebyshev II is monotonic in the passband and equiripple in the stopband. Elliptic approximation is equiripple both in the passband and stopband. The filter order can often be obtained by computing the number of local maximum and minimum.

The digital filters are obtained through bilinear transform (Mitra 2Ed Sec. 7.2 / 3Ed Sec. 9.2). Hence, approximations, amplitude responses and pole-zero plots are related to each other according to the Figure 118.

68. **Problem:** Consider the following prototype analog Butterworth-type lowpass filter

$$H_{\text{protoLP}}(s) = \frac{1}{s+1}$$

- Form an analog first-order lowpass filter with cutoff frequency  $\Omega_c$  by substituting  $H(s) = H_{\text{protoLP}}(\frac{s}{\Omega_c})$ . Draw the pole-zero plot in s-plane.
- Implement a discrete first-order lowpass filter  $H_{\text{imp}}(z)$ , whose cutoff frequency (-3 dB) is at  $f_c = 100$  Hz and sampling rate is  $f_s = 1000$  Hz, applying the impulse-invariant method to  $H(s)$ . Draw the pole-zero plot of the filter  $H_{\text{imp}}(z)$ .
- Implement a discrete first-order lowpass filter  $H_{\text{bil}}(z)$  with the same specifications applying the bilinear transform to  $H(s)$ . Prewrap the edge frequency. Draw the pole-zero plot of the filter  $H_{\text{bil}}(z)$ .

[L0613] **Solution:** The solution to the problem starts from the page 129. Two methods for digital IIR design are shown in the lecture slides, impulse invariant method and bilinear transform method.

**Analog Butterworth lowpass filter**

Analog Butterworth filter is discussed in (Mitra 2Ed Sec. 5.4.2 / 3Ed Sec. 4.4.2). The definition of an analog Butterworth filter with cut-off frequency  $\Omega_c$  is  $|H_a(j\Omega)|^2 =$



$1/(1 + (\frac{\Omega}{\Omega_c})^{2N})$  (Mitra 2Ed Eq. 5.31 / 3Ed Eq. 4.33). The first order ( $N = 1$ ) filter is therefore

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^2}$$

$$H_a(s)H_a(-s) = \frac{1}{1 + \left(\frac{s}{j\Omega_c}\right)^2} = \frac{1}{1 - \left(\frac{s}{\Omega_c}\right)^2}$$

$$= \frac{H(s)}{1 + \left(\frac{s}{\Omega_c}\right)} \cdot \frac{H(-s)}{1 + \left(\frac{-s}{\Omega_c}\right)}$$

where  $s = j\Omega$

$$\Rightarrow H_a(s) = \frac{\Omega_c}{s + \Omega_c}$$

The pole in  $s$ -plane is at  $s = -\Omega_c$ .

Here,  $\Omega$  refers to frequency in analog domain ( $H(j\Omega)$ ) and  $\omega$  to frequency in digital domain ( $H(e^{j\omega})$ ).

As said earlier, there are two ways to convert analog filter to digital. The impulse-invariant method is straightforward but it has severe limitations. The bilinear transform is a standard way.

**Impulse-invariant method**, see, e.g. lecture slides:

$$H_a(s) \mapsto h_a(t) \mapsto h[n] = h_a(nT) \mapsto H(z)$$

In the impulse-invariant method the target is to get impulse response of digital filter  $h[n]$  to be the same as the sampled impulse response of analog filter  $h_a(nT)$ . Because IIR filters have normally an impulse response of infinite length, this method brings distortion.

The **bilinear transformation** is acquired when

$$s = k \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$

is inserted into the system function (Mitra 2Ed Eq. 7.21 / 3Ed Eq. 9.15)

$$H(z) = H_a(s) \Big|_{s=k \cdot \frac{1-z^{-1}}{1+z^{-1}}}$$

Note that here  $k$  is a parameter used in the derivation of the bilinear transformation. It is originally  $k = (2/T)$  but can be set  $k = 1$  to simplify the procedure.

The **frequency is warped** before the bilinear transformation (Mitra 2Ed Fig. 7.4, 7.5 / 3Ed Fig. 9.3, 9.4). In the small frequencies the difference is not big, but it is significant in high frequencies. Therefore the discrete-time normalized angular cut-off frequency  $\omega_c$  has to be first **prewarped** into analog-time prewarped cut-off frequency  $\Omega_{pc}$ :

$$\Omega_{pc} = k \cdot \tan\left(\frac{\omega_c}{2}\right)$$

where  $\omega_c = 2\pi f_c / f_T = 2\pi f_c T = \Omega_c T$ , and  $0 < \omega_c < \pi$ , and  $[f_c] = \text{Hz}$ , and  $f_T = 1/T$  is the sampling frequency. For example, if discrete-time  $f_c = 100 \text{ Hz}$  and  $f_s = 1000 \text{ Hz}$ , then  $\Omega_{pc} = 2000 \cdot \tan(0.1\pi)$ , and  $f_{pc} \approx 103.4 \text{ Hz}$ . Analog design has to be done using  $f_{pc}$  instead of  $f_c$  in order to get the cut-off frequency to 100 Hz in the digital filter.

**Solution to Problem 68**

a) Substitution gives directly

$$H(s) = H_{\text{protoLP}}(s/\Omega_c) = \frac{\Omega_c}{s + \Omega_c}$$

The pole-zero plot of a lowpass filter in  $s$ -plane is in Figure 119.

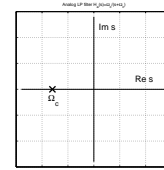


Figure 119: Problem 68(a), LP in  $s$ -plane. The stable pole is at  $s = -\Omega_c$  in the left subspace, the  $y$ -axis is the frequency.

b) Transfer function using the impulse-invariant method.

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c} \mapsto h_a(t) = \Omega_c e^{-\Omega_c t} \mu(t) \mapsto$$

$$h[n] = h_a(nT) = \Omega_c e^{-\Omega_c nT} \mu[n] \mapsto H(z) = \Omega_c \sum_{n=0}^{\infty} e^{-\Omega_c nT} z^{-n} = \frac{\Omega_c}{1 - e^{-\Omega_c T} z^{-1}}$$

The constant  $K$  is introduced in order to scale the maximum of  $|H(e^{j\omega})|$  into unity. Using (Mitra 2Ed Eq. 7.7 / 3Ed Eq. 9.7),  $\omega_c = \Omega_c / f_T = 2\pi f_c / f_T$  and values  $f_T = 1 \text{ kHz}$  (sampling frequency) and  $f_c = 100 \text{ Hz}$  (cut-off frequency),

$$H(z)_{\text{Imp}} = \frac{K}{1 - e^{-\omega_c} z^{-1}} = \frac{K}{1 - e^{-\pi/5} z^{-1}}$$

We also know that the maximum is located at zero frequency, because the frequency response of a Butterworth filter is monotonic. Thus we get

$$\frac{K}{1 - e^{-\pi/5}} = 1 \Leftrightarrow K = 1 - e^{-\pi/5}$$

The transfer function of the filter is therefore

$$H(z)_{\text{Imp}} = \frac{1 - e^{-\pi/5}}{1 - e^{-\pi/5} z^{-1}} = 0.4665 \cdot \frac{1}{1 - 0.5335 z^{-1}}$$

There is a pole at  $z = 0.5335$ , see Figure 120 for the amplitude response in linear scale, in desibels and the pole-zero plot.

c) Transfer function using bilinear transform. Compute the normalized angular discrete-time cut-off frequency  $\omega_c$ ,

$$\omega_c = \frac{2\pi\Omega_c}{\Omega_s} = \frac{2\pi 2\pi f_c}{2\pi f_T} = \frac{2\pi f_c}{f_T} = 0.2\pi$$

and the prewarped cut-off frequency  $\Omega_{pc}$  ( $k = 2/T$ ):

$$\Omega_{pc} = k \cdot \tan\left(\frac{\omega_c}{2}\right) = k \cdot \tan(0.1\pi)$$

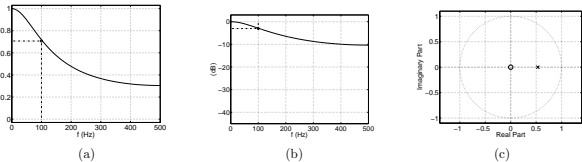


Figure 120: Problem 68, the filter  $H_{\text{Imp}}(z)$  using impulse-invariant method. (a) Amplitude response in linear scale  $|H(e^{j\omega})|$  and (b) in desibels  $10 \cdot \log_{10} |H(e^{j\omega})|^2$ , (c) pole-zero diagram.

The digital filter is obtained through bilinear transform:

$$H(z) = H(s) \Big|_{s=k \cdot \frac{1-z^{-1}}{1+z^{-1}}, \Omega_c=\Omega_{pc}=k \cdot \tan(0.1\pi)}$$

$$= \frac{\Omega_c}{s + \Omega_c} \Big|_{s=k \cdot \frac{1-z^{-1}}{1+z^{-1}}, \Omega_c=\Omega_{pc}=k \cdot \tan(0.1\pi)}$$

$$= \frac{k \cdot \tan(0.1\pi)}{k \cdot \frac{1-z^{-1}}{1+z^{-1}} + k \cdot \tan(0.1\pi)} \Big|_K$$

$$= \frac{\tan(0.1\pi)(1+z^{-1})}{(1+\tan(0.1\pi)) - (1-\tan(0.1\pi))z^{-1}}$$

The last task is to normalize the transfer function. The constant term in denominator polynomial should be scaled to 1, and the maximum value of the amplitude response to 1. While this is a Butterworth lowpass filter, the maximum is reached at  $\omega = 0$ , i.e.,  $z = e^{j\omega} \Big|_{\omega=0} = 1$ .

$$|H(z)_{\text{Bil}}|_{\text{max}} = \left| K \cdot \frac{1+z^{-1}}{1 - \frac{1-\tan(0.1\pi)}{1+\tan(0.1\pi)} z^{-1}} \right|_{z=1} = 1$$

Finally,

$$H_{\text{Bil}}(z) = 0.2452 \cdot \frac{1+z^{-1}}{1 - 0.5095 z^{-1}}$$

There is a zero at  $z = -1$  and a pole at  $z = 0.5095$ . See Figure 121 for the amplitude response in linear scale, in (power) desibels ( $20 \cdot \log_{10}(A) = 10 \cdot \log_{10}(A^2)$ ), and the pole-zero plot. Compare also to the filter obtained through the impulse-invariant method in Figure 120.

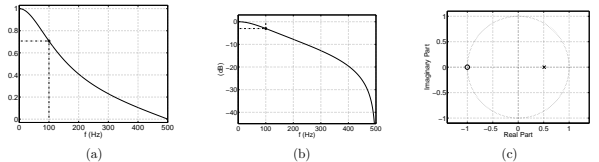


Figure 121: Problem 68, the filter  $H_{\text{Bil}}(z)$  using bilinear transform. (a) Amplitude response in linear scale and (b) in desibels, (c) pole-zero diagram.

**69. Problem:**

- a) Show that the best finite-length approximation to the ideal infinite length impulse response in the mean square error sense is obtained by truncated Fourier series method.
- b) What is the disadvantage of this method and what are the solutions to this problem?

[L0618] **Solution:**

- a) Let the  $H_d(e^{j\omega})$  denote the desired frequency response function. Since  $H_d(e^{j\omega})$  is a periodic function of  $\omega$  with the period  $2\pi$ , it can be expressed as a Fourier series:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n}$$

where the Fourier coefficients  $h_d[n]$  are the corresponding impulse response samples given by

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega, \quad -\infty \leq n \leq \infty$$

For most practical solutions,  $h_d$  is of infinite length and noncausal. Therefore we try to find a finite-duration impulse response sequence  $h_t[n]$  of length  $2M+1$ , whose DFT  $H_t(e^{j\omega})$  approximates the  $H_d(e^{j\omega})$  in some sense. One commonly used approximation criteria is to minimize the integral squared-error

$$\Phi = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_t(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

where

$$H_t(e^{j\omega}) = \sum_{n=-M}^M h_t[n] e^{-j\omega n}$$

Using Parseval's relation

$$\sum_{n=-\infty}^{\infty} g[n] h^*[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(e^{j\omega}) H^*(e^{j\omega}) d\omega,$$

we can rewrite the cost function

$$\Phi = \sum_{n=-\infty}^{\infty} |h_t[n] - h_d[n]|^2 = \sum_{n=-M}^M |h_t[n] - h_d[n]|^2 + \sum_{n=-\infty}^{-M-1} h_d^2[n] + \sum_{n=M+1}^{\infty} h_d^2[n].$$

It obvious from the latter form that the integral-squared error is a minimum when  $h_t[n] = h_d[n]$ , for  $-M \leq n \leq M$ , that is the best finite-length approximation to the ideal infinite-length impulse response in the mean-square error sense is obtained by truncation.

The causality is achieved by delaying the  $h_t[n]$  by  $M$  samples. (Mitra 2Ed Sec. 7.6.1, p. 447 / 3Ed Sec. 10.2.1, p. 527)

- b) Disadvantage is the oscillatory behaviour of  $H_t(e^{j\omega})$  (Gibbs phenomenon). This is caused by simple truncation (window function with abrupt transitions in time domain) and the instability of the ideal filter. Possible solutions are the use of tapered windows (fixed or adjustable) and specification of FIRs with smooth transitions.

- 70. **Problem:** Use windowed Fourier series method and design a FIR-type (causal) lowpass filter with cutoff frequency  $3\pi/4$ . Let the order of the filter be 4.

See Figure 122, in left the amplitude response of the ideal lowpass filter  $H(e^{j\omega})$  with cut-off frequency at  $3\pi/4$ . In right, the corresponding inverse transform of the desired ideal filter  $h_d[n]$ , which is sinc-function according to the transform pair  $\text{rect}(\cdot) \leftrightarrow \text{sinc}(\cdot)$ :

$$h_d[n] = \{ \dots, -0.1592, 0.2251, \underline{0.75}, 0.2251, -0.1592, \dots \}$$

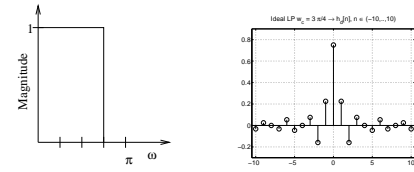


Figure 122: Problem 70: (a) The amplitude response of the ideal lowpass filter, and (b) the corresponding impulse response  $h_d[n]$  values. The cut-off frequency is at  $\omega_c = 3\pi/4$ .

- a) Use the rectangular window of length 5, see Figure 123(a). The window function is  $w_r[n] = 1, -M \leq n \leq M, M = 2$
- b) Use the Hamming window of length 5, see Figure 123(b). The window function is

$$w_h[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{2M}\right), \quad -M \leq n \leq M, M = 2$$

which results to  $w_h[n] = \{0.08, 0.54, \underline{1}, 0.54, 0.08\}$

- c) Compare how the amplitude responses of the filters designed in (a) and (b) differ assuming that the window size is high enough (e.g.  $M = 50$ ).

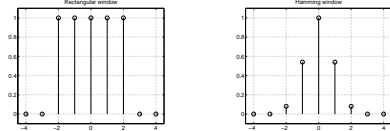


Figure 123: Problem 70: (a) rectangular window  $w_r[n]$  of length 5, and (b) Hamming window  $w_h[n]$  of length 5.

[L0614] **Solution:** Digital FIR filter design with windowed (truncated) Fourier series method. The idea is to find infinite-length impulse response of the ideal filter and truncate it so that a realizable finite-length filter is obtained.

$$h_t[n] = h_d[n] \cdot w[n] \quad \leftrightarrow \quad H_t(z) = H_d(z) \otimes W(z)$$

Now, when cut-off frequency (-3 dB) is at  $\omega_c = 3\pi/4$ , the infinite-length impulse response of the ideal filter is:

$$h_d[n] = \sin\left(\frac{3\pi}{4}n\right) / (\pi n) = (3/4) \text{sinc}\left(\frac{3}{4}n\right)$$

When computing values,  $\sin(x)/x \rightarrow 1$ , when  $x \rightarrow 0$ , or  $\text{sinc}(x) \rightarrow 1$ , when  $x \rightarrow 0$ . So, we get  $h_d[n] = \{ \dots, -0.1592, 0.2251, \underline{0.75}, 0.2251, -0.1592, \dots \}$ .

- a) Now we are using rectangular window  $w_r[n]$  of length 5 (4th order),

$$w_r[n] = \begin{cases} 1, & -2 \leq n \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

Hence,

$$h_t[n] = h_d[n] \cdot w_r[n] = \{-0.1592, 0.2251, \underline{0.75}, 0.2251, -0.1592\}$$

If causal filter is needed, then the shift by two is needed  $h_c[n] = h_t[n - 2] = \{-0.1592, 0.2251, 0.75, 0.2251, -0.1592\}$ .

In Figure 124 time-domain view:

- (a)  $h_d[n]$  (IIR), (b)  $w_r[n]$ , and (c)  $h_t[n] = h_d[n] \cdot w_r[n]$  (FIR).

In Figure 125 the corresponding frequency-domain view:

- (a)  $H_d(e^{j\omega})$  (ideal, desired), (b)  $W_r(e^{j\omega})$ , and (c)  $H_t(e^{j\omega}) = H_d(e^{j\omega}) \otimes W_r(e^{j\omega})$  (realisable).

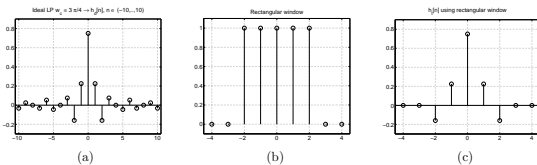


Figure 124: Problem 70(a): time domain view, (a)  $h_d[n]$ , (b)  $w_r[n]$ , (c)  $h_t[n]$ .

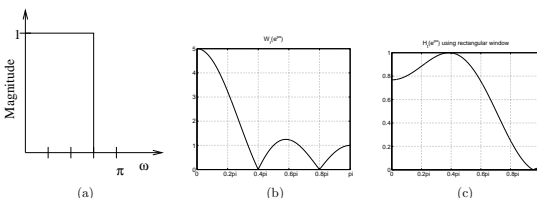


Figure 125: Problem 70(a): frequency domain ( $0 \dots \pi$ ), (a)  $H_d(e^{j\omega})$ , (b)  $W_r(e^{j\omega})$ , (c)  $H_t(e^{j\omega})$ .

- b) Now we are using Hamming window<sup>4</sup>  $w_h[n]$  of length 5,

$$w_h[n] = \begin{cases} 0.54 + 0.46 \cos(2\pi n/4), & -2 \leq n \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

Hence,

$$\begin{aligned} h_t[n] &= h_d[n] \cdot w_h[n] = h_d[n] \cdot (0.54 + 0.46 \cos(2\pi n/(2M))) \\ &= \{0.08 \cdot (-0.1592), 0.54 \cdot 0.2251, \underline{0.75}, 0.54 \cdot 0.2251, 0.08 \cdot (-0.1592)\} \\ &= \{-0.0127, 0.1215, \underline{0.75}, 0.1215, -0.0127\} \end{aligned}$$

If causal filter is needed, then

$$h_c[n] = h_t[n - 2] = \{-0.0127, 0.1215, 0.75, 0.1215, -0.0127\}$$

In Figure 126 time-domain view:

- (a)  $h_d[n]$ , (b)  $w_h[n]$ , and (c)  $h_t[n] = h_d[n] \cdot w_h[n]$ .

In Figure 127 the corresponding frequency-domain view:

- (a)  $H_D(e^{j\omega})$ , (b)  $W_h(e^{j\omega})$ , and (c)  $H_t(e^{j\omega}) = H_D(e^{j\omega}) \otimes W_h(e^{j\omega})$ .

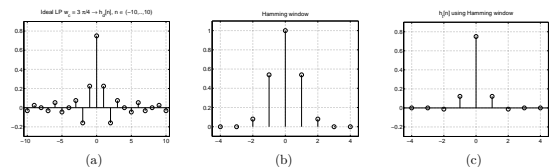


Figure 126: Problem 70(b): time domain view, (a)  $h_d[n]$ , (b)  $w_h[n]$ , (c)  $h_t[n]$ .

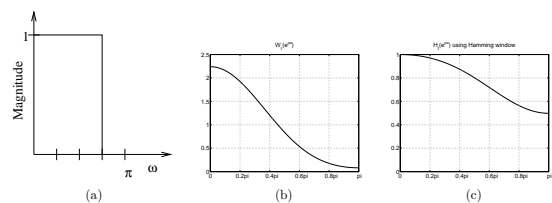


Figure 127: Problem 70(b): frequency domain ( $0 \dots \pi$ ), (a)  $H_D(e^{j\omega})$ , (b)  $W_h(e^{j\omega})$ , (c)  $H_t(e^{j\omega})$ .

- c) **Some examples of window functions:**

- i) Rectangular  $N=11$ , Figure 128
- ii) Rectangular  $N=65$ , Figure 129
- iii) Hamming  $N=65$ , Figure 130

<sup>4</sup>The expression is slightly different from that given in (Mitra 2Ed Eq. 7.75, p. 452 / 3Ed Eq. 10.31, p. 533) but the same as in Matlab.

There are three figures for each item. Top left figure is the window function in time domain  $w[n]$ . The causal version can be obtained by shifting. Bottom left figure is the window function in frequency domain  $W(e^{j\omega})$ . The third figure in right is the amplitude frequency of actual filter which is obtained via window function method. The desired lowpass filter  $H_d(e^{j\omega})$  is drawn in dashed line, the implemented filter  $H_1(e^{j\omega}) = H_d(e^{j\omega}) \otimes W(e^{j\omega})$  is solid line. The cut-off frequency is at 100 Hz, and the sampling frequency is 1000 Hz.

Notice that

- i) Rectangular  $N=11$  gives insufficient result.
- ii) Rectangular  $N=65$  gives sharp transition band but oscillates (Gibbs phenomenon).
- iii) Hamming  $N=65$  is flat both in passband and stopband but the transition band is not as tight as in (ii).

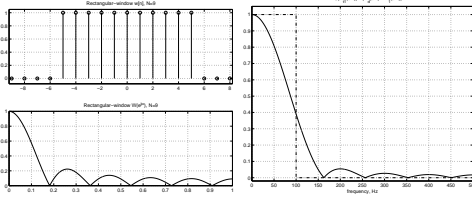


Figure 128: Rectangular window  $N = 11$ , see the text in Problem 70(c).

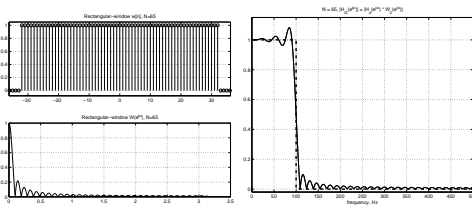


Figure 129: Rectangular window  $N = 65$ , see the text in Problem 70(c).

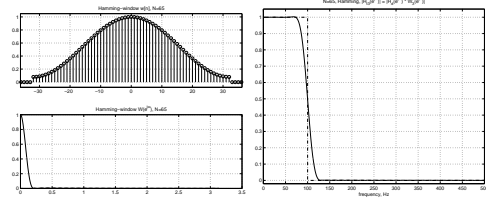


Figure 130: Hamming window  $N = 65$ , see the text in Problem 70(c).

71. **Problem:** The following transfer functions  $H_1(z)$  and  $H_2(z)$  representing two different filters meet (almost) identical amplitude response specifications

$$H_1(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

where  $b_0 = 0.1022$ ,  $b_1 = -0.1549$ ,  $b_2 = 0.1022$ ,  $a_1 = -1.7616$ , and  $a_2 = 0.8314$ , and

$$H_2(z) = \sum_{k=0}^{12} h[k] z^{-k}$$

where  $h[0] = h[12] = -0.0068$ ,  $h[1] = h[11] = 0.0730$ ,  $h[2] = h[10] = 0.0676$ ,  $h[3] = h[9] = 0.0864$ ,  $h[4] = h[8] = 0.1040$ ,  $h[5] = h[7] = 0.1158$ ,  $h[6] = 0.1201$ .

For each filter,

- a) state if it is a FIR or IIR filter, and what is the order
- b) draw a block diagram and write down the difference equation
- c) determine and comment on the computational and storage requirements
- d) determine first values of  $h_1[n]$

[L0730] **Solution:** The transfer functions  $H_1(z)$  and  $H_2(z)$  have been designed using the same amplitude specifications, see Figure 131.

- a)  $H_1(z)$  is IIR. There is a denominator polynomial.  
 $H_2(z)$  is FIR. There is only the nominator polynomial.
- b)  $H_1(z)$  is an IIR filter. In order to show the feedback in time domain one has to use inverse  $z$ -transform:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$Y(z)(1 + a_1 z^{-1} + a_2 z^{-2}) = X(z)(b_0 + b_1 z^{-1} + b_2 z^{-2}) \quad | Z^{-1}\{.\}$$

$$y[n] + a_1 y[n-1] + a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

From the difference equation the block diagram can be drawn (Figure 132). Note that the same coefficients can be found also in the form of  $H_1(z)$ .

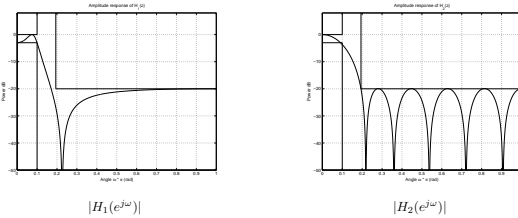


Figure 131: Amplitude responses of  $H_1(z)$  and  $H_2(z)$  in Problem 71.

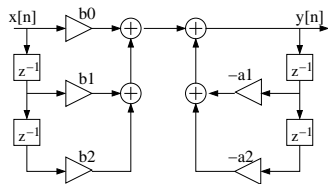


Figure 132:  $H_1(z)$  as a block diagram in Problem 71.

The impulse response  $h_1[n]$  of FIR filter  $H_2(z)$  is directly seen and its length is 13 (finite impulse response). The block diagram consists only of multipliers and delays (Figure 133).

- c) From examination of the two difference equations the computational and storage requirements for both filters are summarized in Table 10.

	FIR	IIR
Number of multiplications	13	5
Number of additions	12	4
Storage locations (coefficients and data)	26	10

Table 10: Computational and storage requirements of  $H_1(z)$  and  $H_2(z)$ .

It is evident that the IIR filter is more economical in both computational and storage requirements than the FIR filter. However, there are some tricks to improve FIR filter structure, see e.g. (Mitra 2Ed Sec. 6.3.3, 6.3.4 / 3Ed Sec. 8.3.3, 8.3.4)

- d) A simple way to determine the impulse response is to insert an impulse  $x[n] = \delta[n]$  into input and compute recursively with difference equation what comes out in  $y[n]$ . The registers are assumed to be zero in the initial moment. Another way to solve first values of  $h_1[n]$  is to apply long division. Unfortunately, both cases are heavy by hands. Inverse  $z$ -transform can be used in order to receive exact  $h_1[n]$ . Using Matlab,

$$h_1[n] = \{0.1022, 0.0251, 0.0615, 0.0875, 0.1029, 0.1086, \dots\}$$

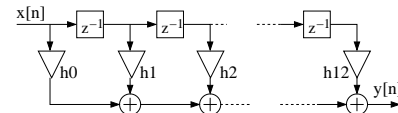


Figure 133:  $H_2(z)$  as a block diagram in Problem 71.

72. **Problem:** See the digital filter structure in Figure 134. Write down all equations for  $w_i[n]$  and  $y[n]$ . Create an equivalent matrix representation  $\mathbf{y}[n] = \mathbf{F}\mathbf{y}[n] + \mathbf{G}\mathbf{y}[n-1] + \mathbf{x}[n]$ , where  $\mathbf{y}[n] = [w_1[n] \ w_2[n] \ w_3[n] \ w_4[n] \ y[n]]^T$ . Verify the computability condition by examining the matrix  $\mathbf{F}$ . Develop a computable set of time-domain equations. Develop the precedence graph (Mitra 2Ed Sec. 8.1, p. 515 / 3Ed Sec. 11.1, p. 589).

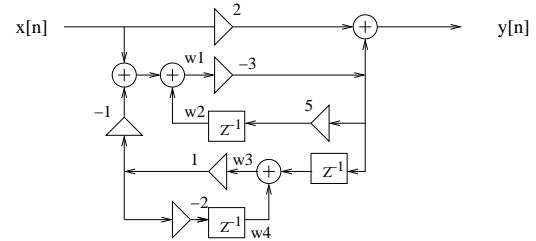


Figure 134: Problem 72: Digital filter structure.

[L0701] **Solution:** In this problem issues of computable set of time-domain equations are considered (Mitra 2Ed Sec. 8.1, p. 515 / 3Ed Sec. 11.1, p. 589). See the digital filter structure in Figure 134.

The difference equations for  $w_i[n]$  and  $y[n]$  can be written as

$$w_1[n] = x[n] - w_3[n] + w_2[n]$$

$$w_2[n] = -15w_1[n-1]$$

$$w_3[n] = -3w_1[n-1] + w_4[n]$$

$$w_4[n] = -2w_3[n-1]$$

$$y[n] = 2x[n] - 3w_1[n]$$

Note that you cannot compute this ordered set of time-domain equations in this order, i.e., the set is noncomputable. For instance, in order to get the value of  $w_1[n]$  one has to compute  $w_2[n]$  and  $w_3[n]$  first. It is not directly seen either, if the structure contains delay-free loops (like  $w_n[n] = ax[n] + \dots + bw_n[n]$ ).

We start from forming a matrix representation for the above set of equations using

$$\mathbf{y}[n] = \mathbf{F}\mathbf{y}[n] + \mathbf{G}\mathbf{y}[n-1] + \mathbf{x}[n]$$

where  $\mathbf{y}[n] = [w_1[n] \ w_2[n] \ w_3[n] \ w_4[n] \ y[n]]^T$ .  $\mathbf{F}$  contains coefficients at the time moment  $n$ , and  $\mathbf{G}$  coefficients at the previous time  $n - 1$ . The matrix representation is

$$\begin{bmatrix} w_1[n] \\ w_2[n] \\ w_3[n] \\ w_4[n] \\ y[n] \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1[n-1] \\ w_2[n-1] \\ w_3[n-1] \\ w_4[n-1] \\ y[n-1] \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -15 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1[n-1] \\ w_2[n-1] \\ w_3[n-1] \\ w_4[n-1] \\ y[n-1] \end{bmatrix} + \begin{bmatrix} x[n] \\ 0 \\ 0 \\ 0 \\ 2x[n] \end{bmatrix}$$

See the matrix  $\mathbf{F}$  closer. If the diagonal element in  $\mathbf{F}$  is nonzero, then the computation of the present value  $w_i[n]$  requires knowledge of itself (delay-free loop), which makes the structure totally noncomputable.

Any nonzero element in the top triangular of  $\mathbf{F}$  makes the ordered set of equation non-computable. The task is to re-order the equations so that this triangular becomes zero.

A signal flow-graph representation of the filter structure is created in Figure 135. The dependent and independent signal variables  $\mathbf{y}[n]$  are represented as nodes. Note that here all different coefficients have been replaced by a single constant  $C = 1$  (omitted) because we are not interested in exact values of variables.

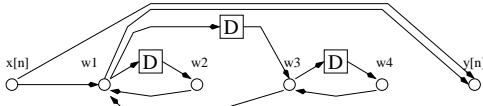


Figure 135: Problem 72: Signal flow-graph representation of the digital filter structure of Figure ?? . All coefficients have been omitted. Delay registers are marked with  $D$ .

First, remove all delay branches and branches going out from the input node, see the reduced signal-flow chart in Figure 136(a). Label all those nodes which have only outgoing branches into a set  $\{N_1\}$ . Second, remove all outgoing branches from nodes  $\{N_1\}$ , see Figure 136(b). Label all nodes which have only outgoing branches into a set  $\{N_2\}$ . Repeat until there is no nodes left. If the algorithm stops before, there is a delay-free loop and the whole system is noncomputable. Here we get

$$\begin{aligned} \{N_1\} &= \{w_2, w_4\} \\ \{N_2\} &= \{w_3\} \\ \{N_3\} &= \{w_1\} \\ \{N_4\} &= \{y\} \end{aligned}$$

The graph with branches and nodes shown in Figure 137 is called precedence graph (Mitra 2Ed Sec. 8.1.2, p. 518 / 3Ed Sec. 11.1.2, p. 592). The computational order of the variables inside the same set  $\{N_i\}$  can be chosen arbitrary.

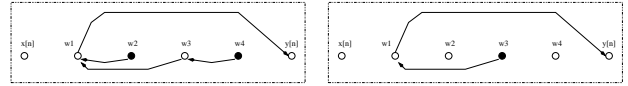


Figure 136: Problem 72: (a) The reduced signal flow-graph obtained by removing outgoing branches from the input and all delay branches. All nodes with only outgoing branches,  $w_2$  and  $w_4$ , belong to the set  $\{N_1\}$ . (b) All outgoing branches from nodes in the set  $\{N_1\}$  have been removed. All nodes with only outgoing branches,  $w_3$ , belong to the set  $\{N_2\}$ .

The computable ordered set of equations is

$$\begin{aligned} w_2[n] &= -15w_1[n-1] \\ w_4[n] &= -2w_3[n-1] \\ w_3[n] &= -3w_1[n-1] + w_4[n] \\ w_1[n] &= x[n] - w_3[n] + w_2[n] \\ y[n] &= 2x[n] - 3w_1[n] \end{aligned}$$

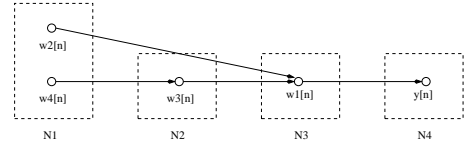


Figure 137: Problem 72: Precedence graph with node sets  $N_i$ . Coefficients have been omitted.

73. **Problem:** Suppose that the calculation of FFT for a one second long sequence, sampled with 44100 Hz, takes 0.1 seconds. Estimate the time needed to compute (a) DFT of a one second long sequence, (b) FFT of a 3-minute sequence, (c) DFT of a 3-minute sequence. The complexities of DFT and FFT can be approximated with  $\mathcal{O}(N^2)$  and  $\mathcal{O}(N \log_2 N)$ , respectively.

[L0700] **Solution:** Fast Fourier Transform (FFT) is a computationally effective algorithm for calculating the Discrete Fourier Transform (DFT) of a sequence (Mitra 2Ed Sec. 8.3.2 / 3Ed Sec. 11.3.2).

The computational complexity of FFT is  $\mathcal{O}(N \log N)$  where  $N$  is the length of the sequence. The complexity of the basic algorithm for DFT is quadratic to the input length i.e.  $\mathcal{O}(N^2)$ .

Here, it is supposed that the calculation of FFT for a one second long sequence, sampled with 44100 Hz, takes 0.1 seconds. Thus, the length of the sequence is  $N = 1 \text{ s} \times 44100 \text{ Hz} = 44100$  samples and we can approximate the number of operations needed for the calculation as  $N \log_2 N$  (using the base-2 logarithm). Since performing these operations takes 0.1 seconds, we get the (average) execution time for a single operation:

$$t = \frac{0.1 \text{ s}}{44100 \log_2(44100)} \approx 147 \text{ ns}$$

a) The time needed to compute DFT of a one second long sequence is estimated as the number of operations needed times the execution time for a single operation:

$$N^2 t = 44100^2 \times 147 \text{ ns} \approx 300 \text{ s} \approx 5 \text{ min}$$

b) A 3-minute sequence, sampled with 44100 Hz, consists of  $N' = 180 \text{ s} \times 44100 \text{ Hz} = 7938000$  samples. Calculating FFT for  $N'$  takes approximately:

$$N' \log_2(N') t = 7938000 \log_2(7938000) \times 147 \text{ ns} \approx 30 \text{ s}$$

c) Calculating DFT for  $N'$  takes approximately:

$$(N')^2 t = 7938000^2 \times 147 \text{ ns} \approx 9 \cdot 10^6 \text{ s} \approx 100 \text{ d}$$

It should be noted that these are only very crude approximations of the actual time it takes to calculate the FFT and DFT algorithms with different sizes of input sequences. The  $\mathcal{O}(\cdot)$  notation omits all additive constants and constant coefficients of the complexity and concerns only the asymptotic behavior of complexity when  $N$  grows without limit. In addition, the length of  $N$  is assumed to be a power of 2 in FFT algorithms.

74. **Problem:** Using radix-2 DIT FFT algorithm with modified butterfly computational module compute discrete Fourier transform for the sequence  $x[n] = \{2, 3, 5, -1\}$  (Mitra 2Ed Sec. 8.3.2, p. 538 / 3Ed Sec. 11.3.2, p. 610). The equation pair on  $r$ th level (Mitra 2Ed Eq. 8.42a, 8.42c, p. 543 / 3Ed Eq. 11.45a, 11.45c, p. 614)

$$\begin{aligned} \Psi_{r+1}[\alpha] &= \Psi_r[\alpha] + W_N^k \Psi_r[\beta] \\ \Psi_{r+1}[\beta] &= \Psi_r[\alpha] - W_N^k \Psi_r[\beta] \end{aligned}$$

[L0766] **Solution:** Discrete Fourier transform (DFT)

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad k = 0 \dots N - 1$$

where  $W_N = e^{-j2\pi/N}$ , can be computed efficiently using fast Fourier transform (FFT) algorithms. Algorithms are based on “divide and rule” – decomposing the  $N$ -point DFT computation into smaller ones, and taking advantage of the periodicity and symmetry properties of  $W_N^{nk}$ . The computational complexity of DFT is quadratic  $\mathcal{O}(N^2)$  whereas that of FFT is  $\mathcal{O}(N \log_2 N)$ . The difference is remarkable with large  $N$ .  $N$  is required to be power of two  $2^\mu = N$ . In addition, the temporary results during the algorithm can be saved in the same registers (in-place computation), which is desirable for memory management. See the literature for more details about deriving FFT algorithms (Mitra 2Ed Sec. 8.3.2, p. 540 / 3Ed Sec. 11.3.2, p. 610).

Here we apply radix-2 DIT FFT algorithm to compute DFT. DIT stands for decimation-in-time and radix-2 means that the decimation factor is 2 at each step. Modified butterfly module is depicted in Figure 138 and with equations (Mitra 2Ed Eq. 8.42a, 8.42c, p. 543 / 3Ed Eq. 11.45a, 11.45c, p. 614)

$$\begin{aligned} \Psi_{r+1}[\alpha] &= \Psi_r[\alpha] + W_N^k \Psi_r[\beta] \\ \Psi_{r+1}[\beta] &= \Psi_r[\alpha] - W_N^k \Psi_r[\beta] \end{aligned}$$

where  $r$  is the level of computation  $r = 1, \dots, \mu$ , constant  $\mu = \log_2 N_{x[n]}$ , and at each level  $r$  the coefficients  $W_N^k$ :  $N = N(r) = 2^r = 2^l, \dots, 2^\mu$ , and  $l = 0, \dots, 2^r - 1 - 1$ . The number of  $\Psi_r[m]$  is  $N_{x[n]}$ ,  $m = 0, \dots, N_{x[n]} - 1$ .

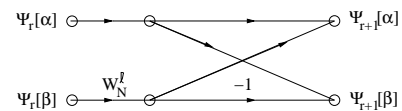


Figure 138: Problem 74: Modified butterfly module.

Here  $\{\Psi_r[m]\}$  is the input sequence in the bit-reversed order to be transformed, that is, a sample  $x[(b_\mu \dots b_2 b_1 b_0)_2]$  appears in the location  $m = (b_0 b_1 b_2 \dots b_\mu)_2$  as  $\Psi_r[(b_0 b_1 b_2 \dots b_\mu)_2]$  (bits  $b_i \in \{0, 1\}$ ).

The algorithm is efficient also in the sense that only values of one level  $r$  (number of  $N_{x[n]}$ ) has to be kept in the memory, this is called *in-place computation*.

Hence, here we apply radix-2 DIT FFT algorithm to compute DFT of the sequence  $x[n] = \{2, 3, 5, -1\}$ , where  $N_{x[n]} = 4$  and  $\mu = \log_2 4 = 2$ . Bit-reversed order gives the starting point

- (index  $(0)_{10} = (00)_2$ )  $\Psi_1[0] = x[0] = 2$  (bit-reversed  $(00)_2 = (0)_{10}$ )
- (index  $(1)_{10} = (01)_2$ )  $\Psi_1[1] = x[2] = 5$  (bit-reversed  $(10)_2 = (2)_{10}$ )
- (index  $(2)_{10} = (10)_2$ )  $\Psi_1[2] = x[1] = 3$  (bit-reversed  $(01)_2 = (1)_{10}$ )
- (index  $(3)_{10} = (11)_2$ )  $\Psi_1[3] = x[3] = -1$  (bit-reversed  $(11)_2 = (3)_{10}$ )

The flow-graph for the algorithm is depicted in Figure 139. Computing the layer  $r = 1$  where  $W_N^j = \{W_2^0\} = \{1\}$

$$\begin{aligned} \Psi_2[0] &= \Psi_1[0] + W_2^0 \Psi_1[1] = 2 + 5 = 7 \\ \Psi_2[1] &= \Psi_1[0] - W_2^0 \Psi_1[1] = 2 - 5 = -3 \\ \Psi_2[2] &= \Psi_1[2] + W_2^0 \Psi_1[3] = 3 - 1 = 2 \\ \Psi_2[3] &= \Psi_1[2] - W_2^0 \Psi_1[3] = 3 + 1 = 4 \end{aligned}$$

and the layer  $r = 2$  with  $W_N^j = \{W_4^0, W_4^1\} = \{1, -j\}$

$$\begin{aligned} \Psi_3[0] &= \Psi_2[0] + W_4^0 \Psi_2[2] = 7 + 2 = 9 \\ \Psi_3[2] &= \Psi_2[0] - W_4^0 \Psi_2[2] = 7 - 2 = 5 \\ \Psi_3[1] &= \Psi_2[1] + W_4^1 \Psi_2[3] = -3 - 4j \\ \Psi_3[3] &= \Psi_2[1] - W_4^1 \Psi_2[3] = -3 + 4j \end{aligned}$$

which gives the final result (compare to Problem 13 and Problem 50)

$$\begin{aligned} X[0] &= \Psi_3[0] = 9 \\ X[1] &= \Psi_3[1] = -3 - 4j \\ X[2] &= \Psi_3[2] = 5 \\ X[3] &= \Psi_3[3] = -3 + 4j \end{aligned}$$

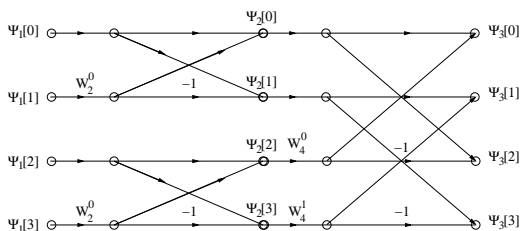


Figure 139: Problem 74: Flow-graph of the radix-2 DIT FFT algorithm with modified butterfly module.

**Remark.** Radix-2 DIT FFT for the sequence of length  $N = 8$  is just adding one layer more as can be seen in Figure 140. In practice  $N$  depends on the application, e.g., in speech processing  $N$  corresponding to 20 milliseconds with  $f_T = 44100$  Hz results to  $0.02 \cdot 44100 \rightarrow$  either  $N = 512$  or  $N = 1024$ .

No overflow occurred this time.

- c) The difference  $0_{\Delta}10101 - 0_{\Delta}01111$  can be carried out as an addition of positive binary fraction  $0_{\Delta}10101$  and the two's-complement representation of  $-0_{\Delta}01111$ , which is given by  $1_{\Delta}10001$ .

$$\begin{array}{r} 1 \quad \quad \quad 1 \quad \text{carry} \\ 0 \ \Delta \ 1 \ 0 \ 1 \ 0 \ 1 \\ + 1 \ \Delta \ 1 \ 0 \ 0 \ 0 \ 1 \\ \hline 10 \ \Delta \ 0 \ 0 \ 1 \ 1 \ 0 \end{array}$$

The extra 1 bit on the left of the sign bit is dropped resulting in  $0_{\Delta}00110$ , which is the correct difference.

- d)  $0_{\Delta}01011 - 0_{\Delta}10001$ . The complement of  $-0_{\Delta}10001$  is given by  $1_{\Delta}01111$ .

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \ \text{carry} \\ 0 \ \Delta \ 0 \ 1 \ 0 \ 1 \ 1 \\ + 1 \ \Delta \ 0 \ 1 \ 1 \ 1 \ 1 \\ \hline 1 \ \Delta \ 1 \ 1 \ 0 \ 1 \ 0 \end{array}$$

which can be changed back to a positive fraction by inverting bits and adding one:  $-0_{\Delta}00110$ .

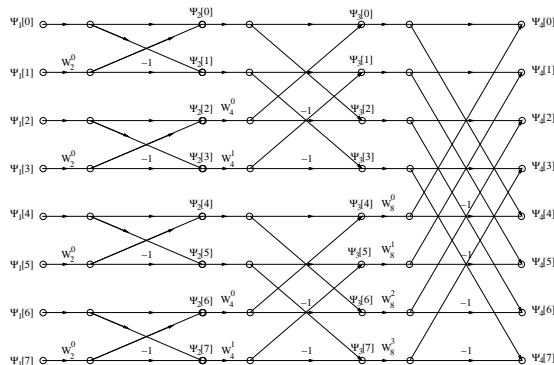


Figure 140: Problem 74: The same algorithm with  $N = 8$ , i.e., bit-reversed inputs  $\Psi_1[0], \dots, \Psi_1[7]$ . In the right-most layer the multipliers are  $W_8^0, W_8^1, W_8^2$ , and  $W_8^3$ , and the outputs are  $\Psi_4[0], \dots, \Psi_4[7]$ .

75. **Problem:** Perform the following binary additions

- a)  $0_{\Delta}10101 + 0_{\Delta}01111$
- b)  $0_{\Delta}01011 + 0_{\Delta}10001$

and compute the differences by performing binary additions of a positive fraction and a negative number represented in two's complement form:

- a)  $0_{\Delta}10101 - 0_{\Delta}01111$
- b)  $0_{\Delta}01011 - 0_{\Delta}10001$

[L0768] **Solution:** Negative fractions are often represented in two's complement form, which is received from a positive fraction by inverting all bits and adding one.

- a)  $0_{\Delta}10101 + 0_{\Delta}01111$

$$\begin{array}{r} 1 \quad 1 \ 1 \ 1 \ 1 \ \text{carry} \\ 0 \ \Delta \ 1 \ 0 \ 1 \ 0 \ 1 \\ + 0 \ \Delta \ 0 \ 1 \ 1 \ 1 \ 1 \\ \hline 1 \ \Delta \ 0 \ 0 \ 1 \ 0 \ 0 \end{array}$$

As the sign bit is 1, there has been overflow and the sum is not correct.

- b)  $0_{\Delta}01011 + 0_{\Delta}10001$

$$\begin{array}{r} 1 \ 1 \ \text{carry} \\ 0 \ \Delta \ 0 \ 1 \ 0 \ 1 \ 1 \\ + 0 \ \Delta \ 1 \ 0 \ 0 \ 0 \ 1 \\ \hline 0 \ \Delta \ 1 \ 1 \ 1 \ 0 \ 0 \end{array}$$

76. **Problem:** Express the decimal number  $-0.3125$  as a binary number using sign bit and four bits for the fraction in the format of (a) sign-magnitude, (b) ones' complement, (c) two's complement. What would be the value after truncation, if only three bits are saved.

[L0764] **Solution:** The binary number representation is discussed in (Mitra 2Ed Sec. 8.4 / 3Ed Sec. 11.8). Now,  $-0.3125 = -5/16$ . We can express it in fixed-point representation using a sign bit  $s$  and four bits for the fraction.

There are three different forms for negative numbers, for which all the sign bit is 0 for a positive number and 1 for a negative number.

- a) Sign-magnitude format:  $1_{\Delta}0101$ .  
 $b$ -bit fraction is always  $\sum_{i=1}^b a_{-i} 2^{-i}$ . For a negative number  $s = 1$ :  
 $S = -(0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4}) = -0.3125$ .
- b) Ones' complement:  $1_{\Delta}1010$ .  
 Decimal number  $S = -s(1 - 2^{-b}) + \sum_{i=1}^b a_{-i} 2^{-i}$ . The negative number can also be achieved by complementing all bits of the corresponding positive value ( $+0.3125 \hat{=} 0_{\Delta}0101 \rightarrow 1_{\Delta}1010 \hat{=} -0.3125$ ).  
 $S = -1(1 - 2^{-4}) + (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4})$   
 $= -0.9375 + 0.625 = -0.3125$
- c) Two's complement:  $1_{\Delta}0111$ .  
 Decimal number  $S = -s + \sum_{i=1}^b a_{-i} 2^{-i}$ . It can also be achieved by complementing all bits and adding 1 to the least-significant bit (LSB) ( $+0.3125 \hat{=} 0_{\Delta}0101 \rightarrow 1_{\Delta}1010 + 1 = 1_{\Delta}1011 \hat{=} -0.3125$ ).  
 $S = -1 + (1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4})$   
 $= -1 + 0.6875 = -0.3125$

After truncation

- a)  $1_{\Delta}0101 \rightarrow 1_{\Delta}01 \hat{=} -0.25$
- b)  $1_{\Delta}1010 \rightarrow 1_{\Delta}10 \hat{=} -0.25$
- c)  $1_{\Delta}1011 \rightarrow 1_{\Delta}10 \hat{=} -0.5$

it can be seen that in this case truncation of (a) and (b) produced a bigger number, but (c) a smaller. The analysis of quantization (truncation) process (Mitra 2Ed Sec. 9.1 / 3Ed Sec. 12.1) results to quantization errors depicted in Problem 78.

77. **Problem:** In the following Figure 141, some error probability density functions of the quantization error are depicted.

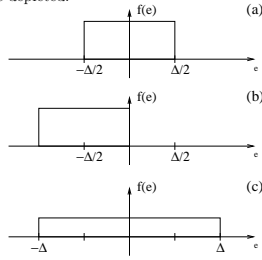


Figure 141: Problem 77: Error density functions.

- (a) Rounding
- (b) Two's complement truncation
- (c) Magnitude (one's complement) truncation

is used to truncate the intermediate results. Calculate the expectation value of the quantization error  $m_e$  and the variance  $\sigma_e^2$  in each case.

$$E[E] = \int_{-\infty}^{\infty} f(e) e \, de, \quad \text{Var}[E] = E[(E - E[E])^2] = E[E^2] - (E[E])^2$$

[L0752] **Solution:** In this problem we are analysing different types of quantization methods.  $\Delta$  here means the quantization step,  $\Delta = 2^{-B}$ . For example, if we are using  $(B + 1) = (4 + 1)$  bits and fixed-point numbers with two's complement representation, possible  $2^{B+1} = 32$  quantized values are  $\{-1, -15/16, -14/16, \dots, 14/16, 15/16\}$ .

The area (integral) of the probability density function  $f(e)$  is always one. All the distributions are uniform. Hence,  $f(e)$  (height of the box) of each pdf is easily computed. We first compute  $E[E] = m_e$  and  $\text{Var}[E] = E[(E - E[E])^2] = \sigma_e^2$  for general uniform distribution (see Figure 142).

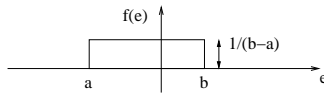


Figure 142: Computing the mean and variance of general uniform distribution in Problem 77.

$$f(e) = \begin{cases} \frac{1}{b-a} & a \leq e \leq b \\ 0 & e < a \vee e > b \end{cases}$$

$$\begin{aligned} m_e &= \int_{-\infty}^{\infty} e f(e) \, de = \int_a^b e \frac{1}{b-a} \, de = \frac{1}{b-a} \int_a^b e \, de \\ &= \frac{1}{b-a} \frac{1}{2} (b^2 - a^2) = \frac{1}{2} \frac{1}{b-a} (b-a)(b+a) = \frac{1}{2} (b+a) \\ \sigma_e^2 &= \int_{-\infty}^{\infty} (e - m_e)^2 f(e) \, de = \int_a^b \left[ e - \frac{1}{2}(a+b) \right]^2 \frac{1}{b-a} \, de \\ &= \frac{1}{b-a} \int_a^b \frac{1}{3} \left[ e - \frac{1}{2}(a+b) \right]^3 \, de \\ &= \frac{1}{3} \frac{1}{b-a} \left\{ \left[ b - \frac{1}{2}(a+b) \right]^3 - \left[ a - \frac{1}{2}(a+b) \right]^3 \right\} \\ &= \frac{1}{3} \frac{1}{b-a} \left\{ \left[ \frac{1}{2}b - \frac{1}{2}a \right]^3 - \left[ \frac{1}{2}a - \frac{1}{2}b \right]^3 \right\} \\ &= \frac{1}{12} \frac{1}{b-a} (b-a)^3 = \frac{1}{12} (b-a)^2 \end{aligned}$$

Computation of mean and variance for each tree cases in the exercise paper, (a) rounding, (b) two's complement truncation, and (c) magnitude truncation.

- a) Rounding:  $a = -\frac{\Delta}{2}, b = \frac{\Delta}{2}$

$$m_e = \frac{1}{2} \left( -\frac{\Delta}{2} + \frac{\Delta}{2} \right) = 0$$

$$\sigma_e^2 = \frac{1}{12} \left[ \frac{\Delta}{2} - \left( -\frac{\Delta}{2} \right) \right]^2 = \frac{\Delta^2}{12}$$

- b) Two's complement truncation:  $a = -\Delta, b = 0$

$$m_e = \frac{1}{2} (-\Delta + 0) = -\frac{\Delta}{2}$$

$$\sigma_e^2 = \frac{1}{12} [0 - (-\Delta)]^2 = \frac{\Delta^2}{12}$$

- c) Magnitude truncation:  $a = -\Delta, b = \Delta$

$$m_e = \frac{1}{2} (-\Delta + \Delta) = 0$$

$$\sigma_e^2 = \frac{1}{12} [\Delta - (-\Delta)]^2 = \frac{\Delta^2}{3}$$

78. **Problem:** In this problem we study the roundoff noise in direct form FIR filters. Consider an FIR filter of length  $N$  having the transfer function

$$H(z) = \sum_{k=0}^{N-1} h[k] z^{-k}$$

Sketch the direct form realization of the transfer function.

- a) Derive a formula for the roundoff noise variance when quantization is done before summations.
- b) Repeat (a) for the case where quantization is done after summations, i.e. a double precision accumulator is used.

[L0753] **Solution:** Direct form realization of the filter. Quantization blocks are marked by  $Q$  in Figure 143.

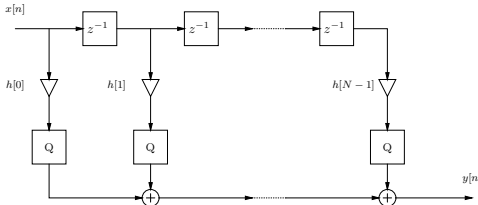


Figure 143: Filter with finite wordlength in Problem 78.

- a) The roundoff noise model ( $e_i[n]$ 's are error sources), when quantization is done before summations, is depicted in Figure 144.

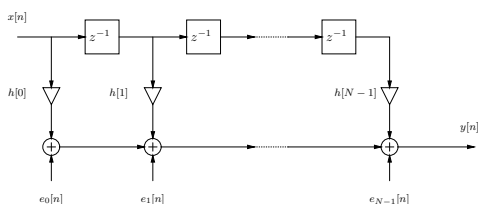


Figure 144: Roundoff noise model with  $N$  quantization points in Problem 78.

It is assumed that the quantization is done using rounding.  $B + 1$  bits are used in the coefficient quantization ( $\Delta = 2^{-B}$ ):

$$\Rightarrow \sigma_e^2 = \frac{2^{-2B}}{12}, \quad m_e = 0 \text{ for all } e_i[n], \quad i = 0, \dots, N-1.$$

Transfer functions from noise sources to the output are equal to unity. Total output noise is thus

$$e[n] = \sum_{i=0}^{N-1} e_i[n].$$

The expectation of the total noise is  $E[e[n]] = 0$  because rounding was applied. The variance of the noise is

$$\begin{aligned} \sigma_{e,tot}^2 &= E[e^2[n]] - \underbrace{E[e[n]]^2}_{=0 \text{ (rounding)}} \\ &= E \left[ \left( \sum_{i=0}^{N-1} e_i[n] \right)^2 \right] \quad [E[e_i[n]e_j[n]] = 0, i \neq j] \\ &= \sum_{i=0}^{N-1} E[e_i^2[n]] = \sum_{i=0}^{N-1} \sigma_e^2 = N\sigma_e^2 = N \frac{2^{-2B}}{12} \end{aligned}$$

- b) The model, when quantization is done after summations, is drawn in Figure 145. Now there is only one quantization point, i.e., there is only one noise source,  $e[n]$ . The expectation of the total noise is again  $E[e[n]] = 0$ . However, variance is smaller. This is achieved by using more bits in temporary computations.

$$\Rightarrow \sigma_{e,tot}^2 = \sigma_e^2 = \frac{2^{-2B}}{12}.$$

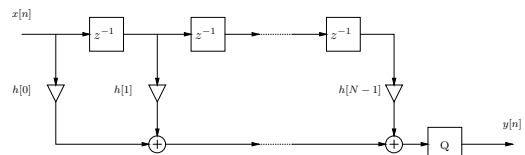


Figure 145: Filter with only one quantization point in Problem 78.

**Remark.** Consider a receipt with several products. In (a) you would round each purchase before summing whereas in (b) you sum all purchases and round them in the end. While this is done for a set of receipts, expectation and variance can be computed.

79. **Problem:** The effect of the additive quantization noise  $e[n]$  on the input signal  $x[n]$  can be evaluated by computing the *signal-to-quantization noise ratio* ( $SNR_{A/D}$ ) in dB defined by

$$SNR_{A/D} = 10 \log_{10} \left( \frac{\sigma_x^2}{\sigma_e^2} \right) \text{ dB},$$

where  $\sigma_x^2$  and  $\sigma_e^2$  are the input signal and noise variances, respectively.

For a bipolar  $(b+1)$ -bit A/D converter, the quantization noise power can be expressed as

$$\sigma_e^2 = \frac{2^{-2b}(R_{FS})^2}{48},$$

where  $R_{FS} = K\sigma_x$  is the full-scale range of the converter.

Derive the formula for  $SNR_{A/D}$  in the case of a bipolar  $(b+1)$ -bit A/D converter and calculate the  $SNR_{A/D}$  for 8, 12, and 16 bit converters with  $K = 6$ .

[L0770] **Solution:** Signal-to-quantization noise ratio in the case of a bipolar  $(b+1)$ -bit A/D converter:

$$\begin{aligned} SNR_{A/D} &= 10 \log_{10} \left( \frac{\sigma_x^2}{\sigma_e^2} \right) \\ &= 10 \log_{10} \left( \frac{48\sigma_x^2}{2^{-2b}(R_{FS})^2} \right) \\ &= -10 \log_{10} 2^{-2b} + 10 \log_{10} 48 + 10 \log_{10} \left( \frac{\sigma_x^2}{(R_{FS})^2} \right) \\ &= 6.02b + 16.81 - 20 \log_{10} \left( \frac{R_{FS}}{\sigma_x} \right) \text{ dB} \end{aligned}$$

Now,  $R_{FS} = K\sigma_x = 6\sigma_x$  and  $b$  is the used bitrate minus one. Therefore, we get:

$$\begin{aligned} b = 7 & : SNR_{A/D} = 6.02 \cdot 7 + 16.81 - 20 \log_{10}(6) = 43.39 \text{ dB} \\ b = 11 & : SNR_{A/D} = 6.02 \cdot 11 + 16.81 - 20 \log_{10}(6) = 67.47 \text{ dB} \\ b = 15 & : SNR_{A/D} = 6.02 \cdot 15 + 16.81 - 20 \log_{10}(6) = 91.56 \text{ dB} \end{aligned}$$

80. **Problem:** The quantization errors occurring in the digital systems may be compensated by error-shaping filters (Mitra 2Ed Sec. 9.10 / 3Ed Sec. 12.10). The error components are extracted from the system and processed e.g. using simple digital filters. In this way part of the noise at the output of the system can be moved to a band of no interest.

Consider a lowpass DSP system with a second-order noise reduction system in Figure 146.

- What is the transfer function of the system if infinite wordlength is used?
- Derive an expression for the transform of the quantized output,  $Y(z)$ , in terms of the input transform,  $X(z)$ , and the quantization error,  $E(z)$ , and hence show that the error feedback network has no adverse effect on the input signal.
- Deduce the expression for the error feedback function.
- What values  $k_1$  and  $k_2$  should have in order to work as an error-shaping system?

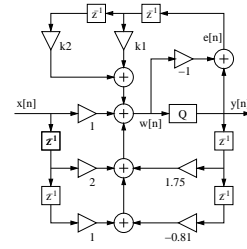


Figure 146: Second-order system with second-order noise reduction in Problem 80.

[L0765] **Solution:** The quantization errors produced in digital systems may be compensated by error-shaping filters. First-order and second-order feedback structures are introduced in (Mitra 2Ed Sec. 9.10.1, 9.10.2 / 3Ed Sec. 12.10.1, 12.10.2). The error components are extracted from the system and processed e.g. using simple digital filters. This way the noise at the output of the system can be reduced.

Consider first the block diagram shown in Figure 147(a) and its round-off noise model in Figure 147(b).

- If infinite precision is used, the quantization is not needed and  $e[n] \equiv 0$  (see Figure 147(b) with  $e[n] = 0$ ). In that case, the system function is

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 1.75z^{-1} + 0.81z^{-2}}$$

Computing zeros and poles we get a pole-zero diagram from which it can be derived that the filter is lowpass (Figure 148).

- From Figure 147(a) it can be obtained the following difference equations:

$$\begin{aligned} e[n] &= y[n] - w[n] \\ w[n] &= (x[n] + 2x[n-1] + x[n-2]) \\ &\quad + (1.75y[n-1] - 0.81y[n-2]) \\ &\quad + (k_1e[n-1] + k_2e[n-2]) \end{aligned}$$

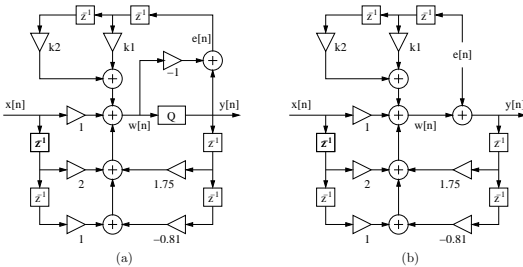


Figure 147: (a) Second-order direct form I system with second-order noise reduction, (b) and its noise model in Problem 80.

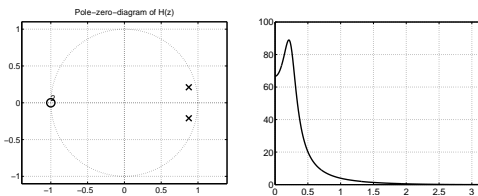


Figure 148: Problem 80: (a) The pole-zero plot and (b) the magnitude response of  $H(z) = (1 + 2z^{-1} + z^{-2}) / (1 - 1.75z^{-1} + 0.81z^{-2})$ .

After z-transform,

$$\begin{aligned} Y(z) &= \left[ \frac{1 + 2z^{-1} + z^{-2}}{1 - 1.75z^{-1} + 0.81z^{-2}} \right] X(z) + \left[ \frac{1 + k_1z^{-1} + k_2z^{-2}}{1 - 1.75z^{-1} + 0.81z^{-2}} \right] E(z) \\ &= H(z)X(z) + H_e(z)E(z) \end{aligned}$$

It can be observed that the noise transfer function  $H_e(z)$  modifies only the quantization error.

- The noise transfer function is

$$H_e(z) = \frac{1 + k_1z^{-1} + k_2z^{-2}}{1 - 1.75z^{-1} + 0.81z^{-2}} = H_{eu}(z)H_{es}(z)$$

Notice that without error-shaping feedback structure, i.e.,  $k_1 = 0$  and  $k_2 = 0$ , the noise transfer function is ( $u =$  unshaped)

$$H_{eu}(z) = \frac{1}{1 - 1.75z^{-1} + 0.81z^{-2}}$$

So, the error-feedback circuit is actually shaping the error spectrum by ( $s =$  shaping)

$$H_{es}(z) = 1 + k_1z^{-1} + k_2z^{-2}$$

- Without error-shaping the quantized output spectrum is

$$Y_u(z) = H(z)X(z) + H_{eu}(z)E(z)$$

Error-shaping filter  $H_{es}(z)$  should efficiently discard the effects of the poles of  $H_{eu}(z)$ . Error-feedback coefficients are chosen to be simple integers or fractions ( $k_i = 0, \pm 0.5, \pm 1, \pm 2$ ), so that the multiplication can be performed using a binary shift operation and it will not introduce an additional quantization error. Choosing  $k_1 = -2, k_2 = 1, H_{es}(z) = 1 - 2z^{-1} + z^{-2}$  is a highpass filter with two zeros at  $z = 1$ . Finally, the compensated output spectrum  $Y_s(z)$  with random white round-off noise  $E(z) = 1$ , can be written as (see Figure 149)

$$\begin{aligned} Y_s(z) &= H(z)X(z) + H_{eu}(z)H_{es}(z)E(z) \\ Y_s(z) &= H(z)X(z) + \frac{1 - 2z^{-1} + z^{-2}}{1 - 1.75z^{-1} + 0.81z^{-2}} \cdot 1 \end{aligned}$$

The error shaping structure lowers the noise in the passband by pushing it into the stopband of the filter (Mitra 2Ed Fig. 9.45 / 3Ed Fig. 12.46).

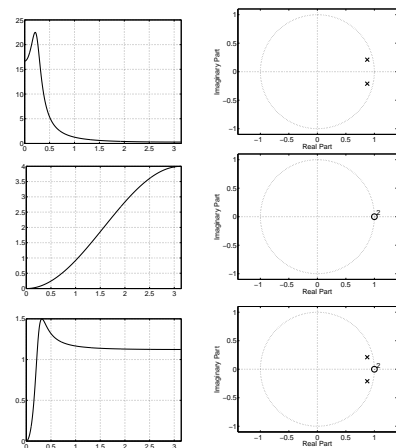


Figure 149: Problem 80: First row,  $H_{eu}(z)$ , error without feedback. Second row,  $H_{es}(z)$ , compensation by FIR-2 error feedback. Third row,  $H_e(z)$ , compensation by FIR-2 error feedback.

81. **Problem:** Consider a cosine sequence  $x[n] = \cos(2\pi(f/f_s)n)$  where  $f = 10$  Hz and  $f_s = 100$  Hz as depicted in the top left in Figure 150. While it is a pure cosine, its spectrum is a peak at the frequency  $f = 10$  Hz (top middle) or at  $\omega = 2\pi f/f_s = 0.2\pi$  (top right).

- a) Sketch the output sequence  $x_u[n]$  with circles using up-sampler with up-sampling factor  $L = 2$ , and draw its spectra into second row. Original sequence values of  $x[n]$  are marked with crosses. The spectrum in middle column is 0..200 Hz and in right 0.. $2\pi$ , i.e.,  $0..f_s$ .

$$x_u[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \dots \\ 0, & \text{otherwise} \end{cases} \quad X_u(e^{j\omega}) = X(e^{j\omega L})$$

- b) Sketch the output sequence  $x_d[n]$  with circles using down-sampler with down-sampling factor  $M = 2$ , and draw its spectra into bottom row.

$$x_d[n] = x[nM] \quad X_d(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\omega-2\pi k)/M})$$

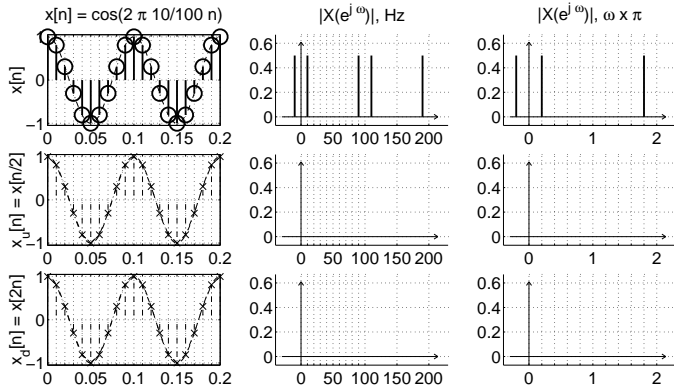


Figure 150: Empty figures for Problem 81. The up-sampling factor  $L = 2$ , and the down-sampling factor  $M = 2$ . **Left column:** sequence  $x[n]$  with circles, fill in the sequences  $x_u[n]$  and  $x_d[n]$ . X-axis: time (0...0.2 s). **Middle column:** Spectrum  $X(e^{j\omega})$  (10 Hz component, 100 Hz sampling frequency), fill in the spectra  $X_u(e^{j\omega})$  and  $X_d(e^{j\omega})$ . X-axis: frequency (0...200 Hz). **Right column:** Spectrum  $X(e^{j\omega})$  ( $2\pi \cdot (10/100) = 0.2\pi$ ), fill in the spectra  $X_u(e^{j\omega})$  and  $X_d(e^{j\omega})$ . X-axis: angular frequency (0... $2\pi$ ).

[L0910] **Solution:** Sometimes it is necessary or useful to change the sampling frequency  $f_s$ . Consider music formats DAT (48 kHz) and CD (44.1 kHz).

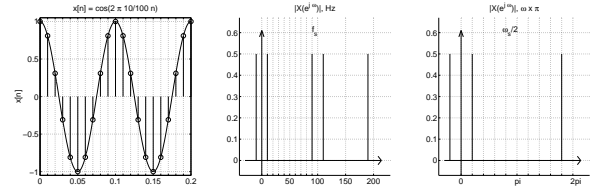


Figure 151: Problem 81(a). The original sequence of a cosine of  $f = 10$  Hz and its spectrum. The angular frequency  $\omega = 2\pi(f/f_s) = 2\pi(10/100) = 0.2\pi$ .

- a) Up-sampling with factor  $L = 2$ . In the time domain there will be  $L-1$  zeros between the original samples, see Figure 152(a).

$$x_u[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \dots \\ 0, & \text{otherwise} \end{cases} \\ = \begin{cases} x[n/2], & n = 0, \pm 2, \pm 4, \dots \\ 0, & \text{otherwise} \end{cases}$$

In the frequency domain the sampling frequency is multiplied by  $L$ , hence, the new sampling frequency is 200 Hz.  $L - 1$  images from the original spectrum are emerged equivalently between 0 and  $f_{s,new}$ .

$$X_u(e^{j\omega}) = X(e^{j\omega L}) = X(e^{j2\omega})$$

Each cosine is a peak pair ( $\pm f$ ) in the spectrum. The original peaks are at  $f = 10$  and  $f = 200 - 10 = 190$  Hz, and after up-sampling new images at  $f = 90$  and  $f = 110$  Hz, as shown in Figure 152(b). The same with angular frequencies is shown in Figure 152(c).

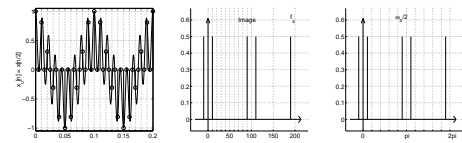


Figure 152: Problem 81(a). Up-sampled signal  $x_u[n]$ , factor  $L = 2$ . The sampling frequency is increased to 200 Hz, and there is an image spectrum.

Notice that if you ideally convert the sequence  $x_u[n]$  into continuous-time  $x_u(t)$  you will find also a high frequency component, an image component. Normally images are filtered out using a lowpass filter (see anti-imaging and anti-aliasing filters). See Figure 153.

- b) Down-sampling with factor  $M = 2$  means taking only every second sample.

$$x_d[n] = x[nM] = x[2n]$$

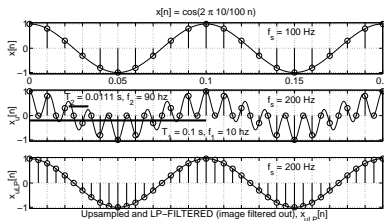


Figure 153: A closer look at up-sampling. Top, original sequence. Middle  $L = 2$ ,  $L - 1 = 1$  zeros added between the original samples. Bottom, using (ideal) LP-filter to remove the image, i.e., 90 Hz component. The continuous curve is plotted only for better visual view. See the text in Problem 81(a).

A possible effect is losing information. However, in this case, this does not occur because  $f = 10$  Hz  $<$   $f_{s,new}/2 = 25$  Hz. See Figure 154(a).

In the frequency domain the sampling frequency is decreased to 50 Hz. See Figures 154(b)-(c).

$$X_d(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\omega-2\pi k)/M})$$

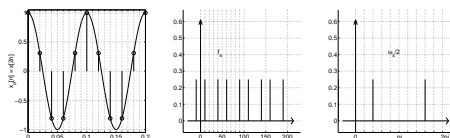


Figure 154: Problem 81(b). Down-sampled signal  $x_d[n]$ , factor  $M = 2$ . The sampling frequency is decreased to 50 Hz.

82. **Problem:** Express the output  $y[n]$  of the system shown in Figure 155 as a function of the input  $x[n]$ .

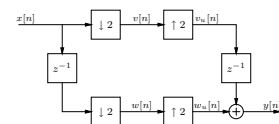


Figure 155: Multirate system of Problem 82.

[L0909] **Solution:** Consider an input signal  $x[n]$  with the corresponding  $z$ -transform  $X(z)$ . After factor-of- $L$  up-sampling, the  $z$ -transform of the signal  $x_u[n]$  is

$$X_u(z) = X(z^L)$$

and after factor-of- $M$  down-sampling, the  $z$ -transform of the signal  $x_d[n]$  is

$$X_d(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M} W_M^{-k})$$

where  $W_M = e^{-j2\pi/M}$ . See (Mitra 2Ed Sec. 10.1.2 / 3Ed Sec. 13.1.2) for the derivation of these equations.

Using these equations, let us derive the  $z$ -transforms of the intermediate signals  $v[n]$ ,  $v_u[n]$ ,  $w[n]$ , and  $w_u[n]$  and finally the  $z$ -transform of the output  $y[n]$ . Let us denote the delayed version of the input as  $X'(z) = z^{-1}X(z)$ . Furthermore, note that  $W_2^{-1} = e^{j2\pi/2} = -1$ .

$$V(z) = \frac{1}{2} \sum_{k=0}^1 X(z^{1/2} W_2^{-k}) = \frac{1}{2} X(z^{1/2}) + \frac{1}{2} X(-z^{1/2})$$

$$W(z) = \frac{1}{2} \sum_{k=0}^1 X'(z^{1/2} W_2^{-k}) = \frac{1}{2} z^{-1/2} X(z^{1/2}) - \frac{1}{2} z^{-1/2} X(-z^{1/2})$$

$$V_u(z) = V(z^2) = \frac{1}{2} X(z) + \frac{1}{2} X(-z)$$

$$W_u(z) = W(z^2) = \frac{1}{2} z^{-1} X(z) - \frac{1}{2} z^{-1} X(-z)$$

$$Y(z) = z^{-1} V_u(z) + W_u(z) = z^{-1} X(z)$$

or  $y[n] = x[n - 1]$  in time-domain (derive the same in time-domain!).



83. **Problem:** Show that the factor-of- $L$  up-sampler  $x_u[n]$  and the factor-of- $M$  down-sampler  $x_d[n]$  defined as in Problem 81 are linear systems.

[L0911] **Solution:** First, consider the up-sampler. Let  $x_1[n]$  and  $x_2[n]$  be two arbitrary inputs with  $y_1[n]$  and  $y_2[n]$  as the corresponding outputs. Now,

$$y_1[n] = \begin{cases} x_1[n/L] & : n = 0, \pm L, \pm 2L, \dots \\ 0 & : \text{otherwise} \end{cases}$$

$$y_2[n] = \begin{cases} x_2[n/L] & : n = 0, \pm L, \pm 2L, \dots \\ 0 & : \text{otherwise} \end{cases}$$

Let us now apply the input  $x_3[n] = \alpha x_1[n] + \beta x_2[n]$  with the corresponding output  $y_3[n]$  as

$$y_3[n] = \begin{cases} \alpha x_1[n/L] + \beta x_2[n/L] & : n = 0, \pm L, \pm 2L, \dots \\ 0 & : \text{otherwise} \end{cases}$$

$$= \begin{cases} \alpha x_1[n/L] & + \\ 0 & \end{cases} + \begin{cases} \beta x_2[n/L] & : n = 0, \pm L, \pm 2L, \dots \\ 0 & : \text{otherwise} \end{cases}$$

$$= \alpha y_1[n] + \beta y_2[n]$$

Thus, the up-sampler is a linear system.

Now, consider the down-sampler with the inputs  $x_1[n]$  and  $x_2[n]$  and the corresponding outputs  $y_1[n]$  and  $y_2[n]$ . Now,  $y_1[n] = x_1[nM]$  and  $y_2[n] = x_2[nM]$ . By applying the input  $x_3[n] = \alpha x_1[n] + \beta x_2[n]$  we get the corresponding output  $y_3[n] = x_3[nM] = \alpha x_1[nM] + \beta x_2[nM]$ . Hence, the down-sampler is also a linear system.

It should also be noted, that both the up-sampler and the down-sampler are time-varying, i.e. not LTI systems.

84. **Problem:** Consider the multirate system shown in Figure 156 where  $H_0(z)$ ,  $H_1(z)$ , and  $H_2(z)$  are ideal lowpass, bandpass, and highpass filters, respectively, with frequency responses shown in Figure 157(a)-(c). Sketch the Fourier transforms of the outputs  $y_0[n]$ ,  $y_1[n]$ , and  $y_2[n]$  if the Fourier transform of the input is as shown in Figure 157(d).

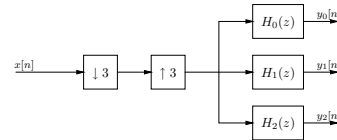


Figure 156: Multirate system of Problem 84.

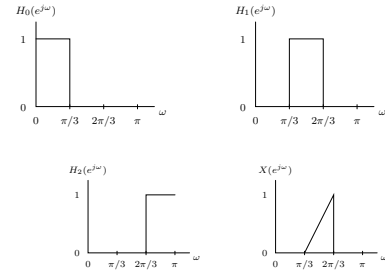


Figure 157: (a)-(c) Ideal filters  $H_0(z)$ ,  $H_1(z)$ ,  $H_2(z)$ , (d) Fourier transform of the input of Problem 84.

[L0912] **Solution:** First, let us denote the down-sampled signal as  $x_d[n]$  and the again up-sampled signal as  $x_u[n]$ , shown in Figure 158.

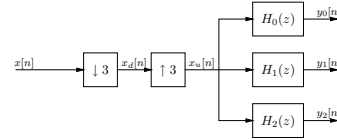


Figure 158: The multirate system in Problem 84.

The corresponding Fourier transforms (spectra)  $X_d(z)$  and  $X_u(z)$  are as follows (notice the reduced amplitude) in Figure 159.

Now, the Fourier transforms of the outputs  $Y_0(z)$ ,  $Y_1(z)$ , and  $Y_2(z)$ , are obtained by (ideally) filtering  $X_u(z)$ . The output spectra are in Figure 160.

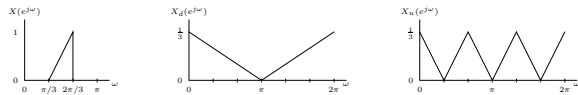


Figure 159: Original, upsampled and downsampled spectrum in Problem 84.

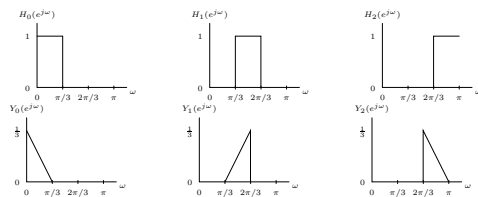


Figure 160: Bandpass filters in top row, and corresponding Output spectra in bottom row in Problem 84.

85. **Problem:** Consider a FIR filter, whose specifications are (i) lowpass, (ii) passband ends at  $\omega_p = 0.15\pi$ , (iii) stopband starts from  $\omega_s = 0.2\pi$ , (iv) passband maximum attenuation is 1 dB, (v) stopband minimum attenuation is 50 dB. The filter is to be implemented using truncated Fourier series method (window method) with Hamming window.

- Sketch the specifications on paper.
- The filter order  $N$  can be estimated using (*Mitra 2Ed Table 7.2 / 3Ed Table 10.2*): the transition bandwidth is  $\Delta\omega = |\omega_p - \omega_s|$ , and for Hamming window  $\Delta\omega = 3.32\pi/M$ , where the window  $w[n]$  is in range  $-M \leq n \leq +M$ . What is the minimum order  $N$  which fulfills the specifications?
- The cut-off frequency of the filter in the window method is defined to be  $\omega_c = 0.5 \cdot (\omega_p + \omega_s)$ . Derive an expression for  $h_{FIR}[n]$  when using (a) and (b). What is the value of  $h_{FIR}[n]$  at  $n = 0$ ?
- Consider now another way to implement a FIR filter with the same specifications. In the interpolated FIR filter (IFIR) (*Mitra 2Ed Sec. 10.3, p. 680 / 3Ed Sec. 10.6.2, p. 568*) the filter is a cascade of two FIR filters  $H_{IFIR}(z) = G(z^L) \cdot F(z)$ . Using the factor  $L = 4$  compute the order of  $H_{IFIR}(z) = G(z^L) \cdot F(z)$  and compare to the original filter  $H_{FIR}(z)$ .

[L0913] **Solution:** Specifications for a FIR filter were the following: (i) lowpass, (ii) passband ends at  $\omega_p = 0.15\pi$ , (iii) stopband starts from  $\omega_s = 0.2\pi$ , (iv) passband maximum attenuation is 1 dB, (v) stopband minimum attenuation is 50 dB. Implementation using truncated Fourier series method (window method) with Hamming window.

- Specifications are drawn in Figure 161(a).

b) Hamming window is defined as

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{2M}\right) \quad -M \leq n \leq +M$$

The window length is  $2M + 1$  and the window/filter order  $N = 2M$ . When having the specifications  $\omega_p = 0.15\pi$  and  $\omega_s = 0.2\pi$  we get  $\Delta\omega = 0.05\pi$ . The connection of  $M$  and transition band  $\Delta\omega$  with Hamming window is

$$M = \lceil \frac{3.32\pi}{\Delta\omega} \rceil$$

where  $\lceil \cdot \rceil$  is rounding up to the next integer. Minimum order with Hamming window is

$$N = 2M = 2 \cdot \lceil \frac{3.32\pi}{0.05} \rceil = 2 \cdot \lceil \frac{3.32}{0.05} \rceil = 134$$

c) The cut-off frequency of the filter in the window method is defined to be  $\omega_c = 0.5 \cdot (\omega_p + \omega_s)$ . The filter of order  $N = 2M$  is computed by

$$h_{FIR}[n] = h_{ideal}[n] \cdot w[n], \quad -M \leq n \leq M$$

where  $h_{FIR}[n]$  is the filter constructed from the ideal filter with cut-off at  $\omega_c = 0.175\pi$  multiplied by a Hamming window with  $M = 67$

$$h_{ideal}[n] = \frac{\sin(\omega_c n)}{\pi n} = \frac{\omega_c}{\pi} \text{sinc}\left(\frac{\omega_c n}{\pi}\right) = 0.175 \text{sinc}(0.175n) \quad -\infty < n < \infty$$

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{134}\right) \quad -67 \leq n \leq 67$$

$$h_{FIR}[n] = 0.175 \text{sinc}(0.175n) \cdot \left(0.54 + 0.46 \cos\left(\frac{2\pi n}{134}\right)\right) \quad -67 \leq n \leq 67$$

In the origo  $w[0] = 1$  and  $h_{FIR}[0] = h_{ideal}[0] = 0.175$ . The magnitude response of the filter is in Figure 161(b) with thick line  $H_{FIR}(z)$ . It can be seen that the filter fulfills given specifications.

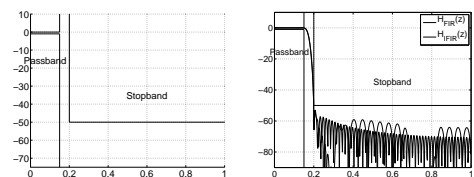


Figure 161: Problem 85: (a) Specifications for the filter. The scale in x-axis  $[0 \dots 1]$  corresponds  $\omega = [0 \dots \pi]$ . (b) Magnitude responses of  $|H_{FIR}(e^j\omega)|$  (thick) and  $|H_{IFIR}(e^j\omega)|$  (thin). Interpolated FIR filter is  $H_{IFIR}(z) = G(z^L) \cdot F(z)$ .

- Consider now another way to implement a FIR filter with the same specifications. In the interpolated FIR filter (IFIR) (*Mitra 2Ed Sec. 10.3, p. 680 / 3Ed Sec. 10.6.2, p. 568*) the filter is a cascade of two FIR filters  $H_{IFIR}(z) = G(z^L) \cdot F(z)$ .  $G(z^L)$  is derived from  $G(z)$  by replacing all delays by  $L$ -multiple delays, i.e., all  $z$  are replaced by  $z^L$  (upsampling).





normalized angular frequency (normalisoitu kulmataajuus), **12**, **23**  
 notch filter (lovisuodin), **25**  
 number representation (numeron esittäminen tietokoneessa), **15**  
 numeron esittäminen tietokoneessa (number representation), **15**  
 näytteenottotaajuus (quantization level), **33**  
 näytteenottotaajuus (sampling frequency), **23**, **33**, **76**  
 odd function (pariton funktio), **7**  
 one-sided spectrum (yksipuoleinen spektri), **26**  
 order (asteluku), **16**  
 osamurtohajotelma (partial fraction decomposition), **18**  
 osamurtokehitemä (partial fraction expansion), **18**  
 osoitin (phasor), **7**, **10**  
 parillinen funktio (even function), **7**  
 pariton funktio (odd function), **7**  
 partial fraction decomposition (osamurtohajotelma), **18**  
 partial fraction expansion (osamurtokehitemä), **18**  
 periodicity of signal (signaalin jaksollisuus), **34**  
 perusjakso (fundamental period), **34**  
 phasor (osoitin), **7**, **10**  
 phasor.m, **10**  
 pienin yhteinen jaettava (least common multiple), **36**  
 poles – roots of denominator polynomial (navat – nimittäjäpolynomin juuret), **17**  
 polynomin juuret (roots of a polynomial), **16**  
 Welch, **26**  
 quantization level (näytteenottotaajuus), **33**  
 quasi-periodic (kvasijaksollinen), **36**  
 radiaanit (radians), **6**  
 radians (radiaanit), **6**  
 rajataajuus (cut-off frequency), **24**  
 rajavärähtely (border effect), **37**, **39**  
 reaali maailman signaalit (real-life signals), **36**  
 real-life signals (reaali maailman signaalit), **36**  
 remainder (jakojäännös), **15**  
 residue (jäännös), **15**  
 residuez, **18**  
 ristikorrelaatio (cross-correlation), **64**  
 roots, **17**  
 roots of a polynomial (polynomin juuret), **16**  
 sampling frequency (näytteenottotaajuus), **23**, **33**, **76**  
 sekoitevaiheinen suodin (mixed-phase filter), **113**  
 sequence (lukujono), **23**, **32**  
 Shannon sampling theorem (Shannonin näytteenottoteoreema), **76**  
 Shannonin näytteenottoteoreema (Shannon sampling theorem), **76**  
 short-time Fourier transform (STFT) (lyhytaika Fourier-muunnos (STFT)), **27**  
 signaalin jaksollisuus (periodicity of signal), **34**  
 siirtofunktio (transfer function), **17**, **31**  
 sinc (sinc), **14**, **21**  
 spectrogram (spektrogrammi), **27**  
 spectrum estimation (spektrin estimointi), **26**  
 spektrin estimointi (spectrum estimation), **26**  
 spektrogrammi (spectrogram), **27**  
 stabiilisuusehto (stability condition), **48**, **73**  
 stability condition (stabiilisuusehto), **48**, **73**  
 stationaarinen signaali (stationary signal), **27**  
 stationary signal (stationaarinen signaali), **27**  
 step response (askelvaste), **51**  
 suodin, suodatint (filter), **24**  
 suora muoto (direct form), **120**  
 suorakulmainen koordinaatisto (Cartesian coordinate system), **6**  
 suotimen asteluku (filter order), **96**  
 superpositio (superposition), **44**, **47**  
 superpositio (superpositio), **44**, **47**  
 suurin yhteinen jakaja (greatest common divisor), **36**  
 taajuus (frequency), **23**  
 taajuustaso (frequency-domain), **31**  
 taajuusvaste (frequency response), **31**, **38**, **39**  
 tehokas laskenta (FFT) (in-place computation (FFT)), **144**  
 time-domain (aikataso), **31**  
 transfer function (siirtofunktio), **17**, **31**  
 transform-domain (muunnostaso), **31**  
 two-point moving average (kahden pisteen keskiarvoistava suodin), **37**  
 two-sided spectrum (kaksipuolinen spektri), **26**  
 unit impulse function (yksikköimpulssifunktio), **32**  
 unit step function (yksikköaskelfunktio), **32**  
 vahvistaa (amplify), **24**  
 vaimentaa (attenuate), **24**  
 viivaspektri (line spectrum), **26**  
 virtauskaavio (flow diagram), **38**, **39**

wavread, **33**  
 yksikköaskelfunktio (unit step function), **32**  
 yksikköimpulssifunktio (unit impulse function), **32**  
 yksipuoleinen spektri (one-sided spectrum), **26**  
 ympyräpuskuri (circular buffer), **15**  
 ympyräsiirto (circular shift), **15**, **91**  
 zero-phase filter (nollavaiheinen suodin), **107**  
 zeros – roots of numerator polynomial (nollat – osoitinpolynomin juuret), **17**  
 äärellisen pitkä impulssivaste, FIR (finite impulse response, FIR), **38**  
 äärettömän pitkä impulssivaste, IIR (infinite impulse response, IIR), **39**