Solved examples on topics presented in Lecture I

Basic signal manipulation (independent variable):

Schaum - example 1.7a-c/20

Basic signal manipulation (dependent variable, multiplication of signals): see also corresponding `Matlab script`

Implement in Matlab audio-effect Fadeout (gradual attenuation of a signal). Load audio file `samba_short.wav` (command `audioread`). Loudness is the amplitude of the audio signal. By multiplication of the signal via monotonically decreasing positive function, the loudness will gradually diminish.

Properties of LTI systems:

Linearity: Schaum - example 1.12a-b/25
Shift invariance: Schaum - example 1.14a-b/27
Causality: Schaum - example 1.20a-c/32
Stability: Schaum - example 1.21a-c/32

Convolution using composition of shifted impulse responses:

Response of LTI system to unit impulse $\delta[n]$ impulse response $h[n]$.
Response of LTI system to amplified unit impulse $c \cdot \delta[n]$ is $c \cdot h[n]$ (due to homogeneity).
Response of LTI system to shifted unit impulse $\delta[n - n_0]$ is $h[n - n_0]$ (due to shift invariance).
General signal $x[n]$ can be expressed as a sum of shifted and amplified pulses.
Response of LTI system to such a signal is thus a sum of responses to each of the particular pulses (due to additivity). This operation is actually the convolution.

Let us compute the convolution of signal $x[n]$ with impulse response $h[n]$ (i.e., $y[n] = h[n] * x[n]$), when given

$$\begin{aligned} x[n] &= 2\delta[n+2] + \delta[n+1] + \delta[n-1] + 3\delta[n-2] \\ h[n] &= \delta[n-1] - \delta[n-2] + 2\delta[n-3]. \end{aligned}$$

The responses can be, for the sake of clarity, written in the form of a table. Here are examples of responses (of the system given by $h[n]$) to amplified and shifted impulses (samples with unspecified value are equal to zero).

n	-2	-1	0	1	2	3	4	5
$T(\delta[n]) = h[n]$				1	-1	2		
$T(2\delta[n]) = 2h[n]$				2	-2	4		
$T(\delta[n+1]) = h[n+1]$			1	-1	2			
$T(\delta[n-1]) = h[n-1]$					1	-1	2	

Similarly, the responses to samples of the signal $x[n]$ can be written by

n	-2	-1	0	1	2	3	4	5
$T(2\delta[n+2]) = 2h[n+2]$		2	-2	4				
$T(\delta[n+1]) = h[n+1]$			1	-1	2			
$T(\delta[n-1]) = h[n-1]$					1	-1	2	
$T(3\delta[n-2]) = 3h[n-2]$						3	-3	6
$y[n] = h[n] * x[n]$		2	-1	3	3	2	-1	6

where the last row is a sum of the partial responses, i.e.,

$$y[n] = h[n] * x[n] = 2\delta[n+1] - \delta[n] + 3\delta[n-1] + 3\delta[n-2] + 2\delta[n-3] - \delta[n-4] + 6\delta[n-5].$$

The length of the convolution output sequence (duration of $y[n]$) is always $L_x + L_h - 1$, where L_x duration of sequence $x[n]$ and L_h duration of sequence $h[n]$. In our example it is thus $L_x + L_h - 1 = 5 + 3 - 1 = 7$.

The first non-zero sample of $y[n]$ has index equal to sum of indices of the first non-zero sample of $x[n]$ and the first non-zero sample of $h[n]$. In our example $(-2) + 1 = -1$.

The last non-zero sample of $y[n]$ has index equal to sum of indices of the last non-zero sample of $x[n]$ and the last non-zero sample of $h[n]$. In our example $2 + 3 = 5$.

Try: to demonstrate the commutativity by swapping the roles of $x[n]$ and $h[n]$ in the table.
Check: the result of the previous calculation using Matlab function (`conv([2 1 0 1 3],[1 -1 2])`).

Convolution in Matlab: see also corresponding `Matlab script`

The response of an LTI system given by impulse response to an input signal is computed using convolution. Alternative terminology: Convolution realizes the filtration of a given signal by a filter given by an impulse response. In Matlab, this is achieved using command `conv`.

Example: Load the samba_short.wav audio file and filter it using two filters given by impulse responses stored in DP.mat and HP.mat. DP is a "low-pass filter"(we will define it later), leaving mainly the bass line. HP is a "high-pass filter", leaving only the highest frequency (part of the cymbal sound).