

12

Filtering

In discussing Fourier transforms, we developed a number of important properties, among them the convolution property and the modulation property. The convolution property forms the basis for the concept of filtering, which we explore in this lecture. Our objective here is to provide some feeling for what filtering means and in very simple terms how it might be implemented.

The concept of filtering is a direct consequence of the fact that for linear, time-invariant systems the Fourier transform of the output is the Fourier transform of the input multiplied by the frequency response, i.e., the Fourier transform of the impulse response. Because of this, the frequency content of the output is the frequency content of the input shaped by this frequency response. Frequency-selective filters attempt to exactly pass some bands of frequencies and exactly reject others. Frequency-shaping filters more generally attempt to reshape the signal spectrum by multiplying the input spectrum by some specified shaping. Ideal frequency-selective filters, such as lowpass, highpass, and bandpass filters, are useful abstractions mathematically but are not exactly implementable. Furthermore, even if they were implementable, in practical situations they may not be desirable. Often frequency-selective filtering is directed at problems where the spectra of the signals to be retained and those to be rejected overlap slightly; consequently it is more appropriate to design filters with a less severe transition from passband to stopband. Thus, nonideal frequency-selective filters have a passband region, a stopband region, and a transition region between the two. In addition, since they are only realized approximately, a certain tolerance in gain is permitted in the passband and stopband.

A very common example of a simple approximation to a frequency-selective filter is a series RC circuit. With the output taken across the capacitor, the circuit tends to reject or attenuate high frequencies and thus is an approximation to a lowpass filter. With the output across the resistor, the circuit approximates a highpass filter, that is, it attenuates low frequencies and retains high frequencies.

Many simple, commonly used approximations to frequency-selective discrete-time filters also exist. A very common one is the class of moving average filters. These have a finite-length impulse response and consist of moving through the data, averaging together adjacent values. A procedure of this type

is very commonly used with stock market averages to smooth out (i.e., reject) the high-frequency day-to-day fluctuations and retain the lower-frequency behavior representing long-time trends. Cyclical behavior in stock market averages might typically be emphasized by an appropriate discrete-time filter with a bandpass characteristic. In addition to discrete-time moving average filters, recursive discrete-time filters are very often used as frequency-selective filters. In the same way that a simple RC circuit can be used as an approximation to a lowpass or highpass filter, a first-order difference equation is often a simple and convenient way of approximating a discrete-time lowpass or highpass filter.

In this lecture we are able to provide only a very quick glimpse into the topic of filtering. In all its dimensions, it is an extremely rich topic with many detailed issues relating to design, implementation, applications, and so on. In the next and later lectures, the concept of filtering will play a very natural and important role.

Suggested Reading

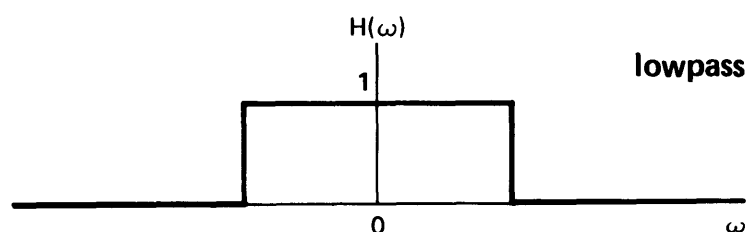
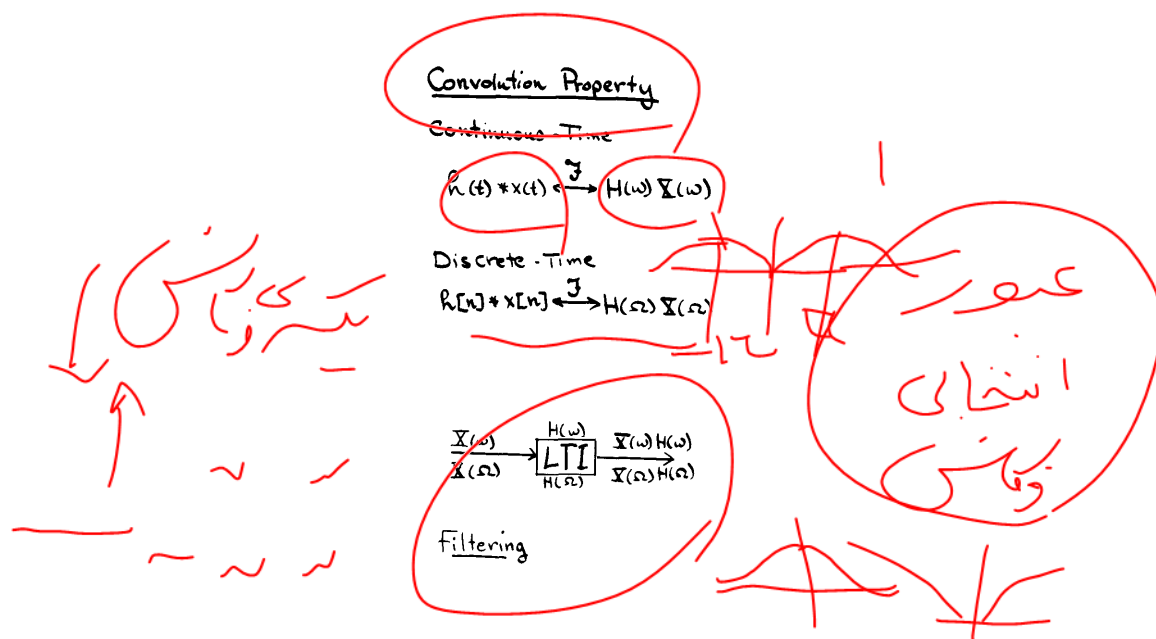
Section 6.1, Ideal Frequency-Selective Filters, pages 401–406

Section 6.2, Nonideal Frequency-Selective Filters, pages 406–408

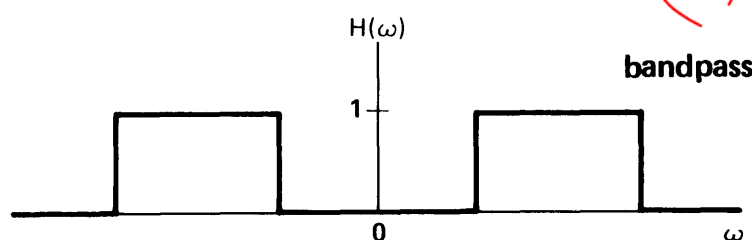
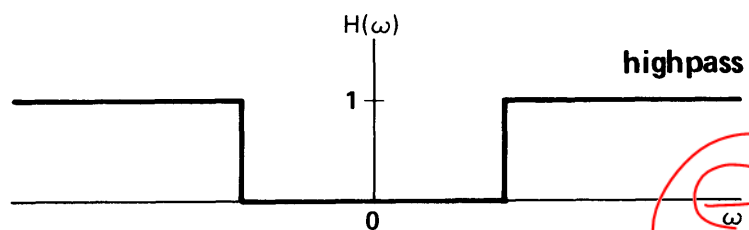
Section 6.3, Examples of Continuous-Time Frequency-Selective Filters Described by Differential Equations, pages 408–413

Section 6.4, Examples of Discrete-Time Frequency-Selective Filters Described by Difference Equations, pages 413–422

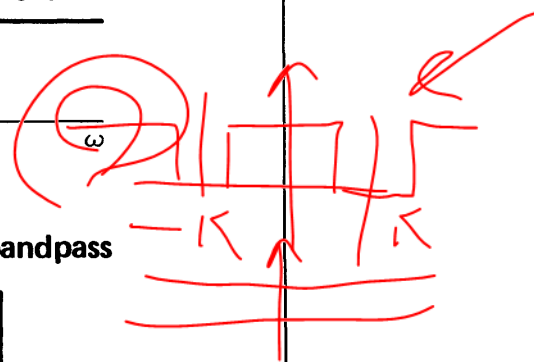
MARKERBOARD
12.1



← Stopband → Passband → Stopband →

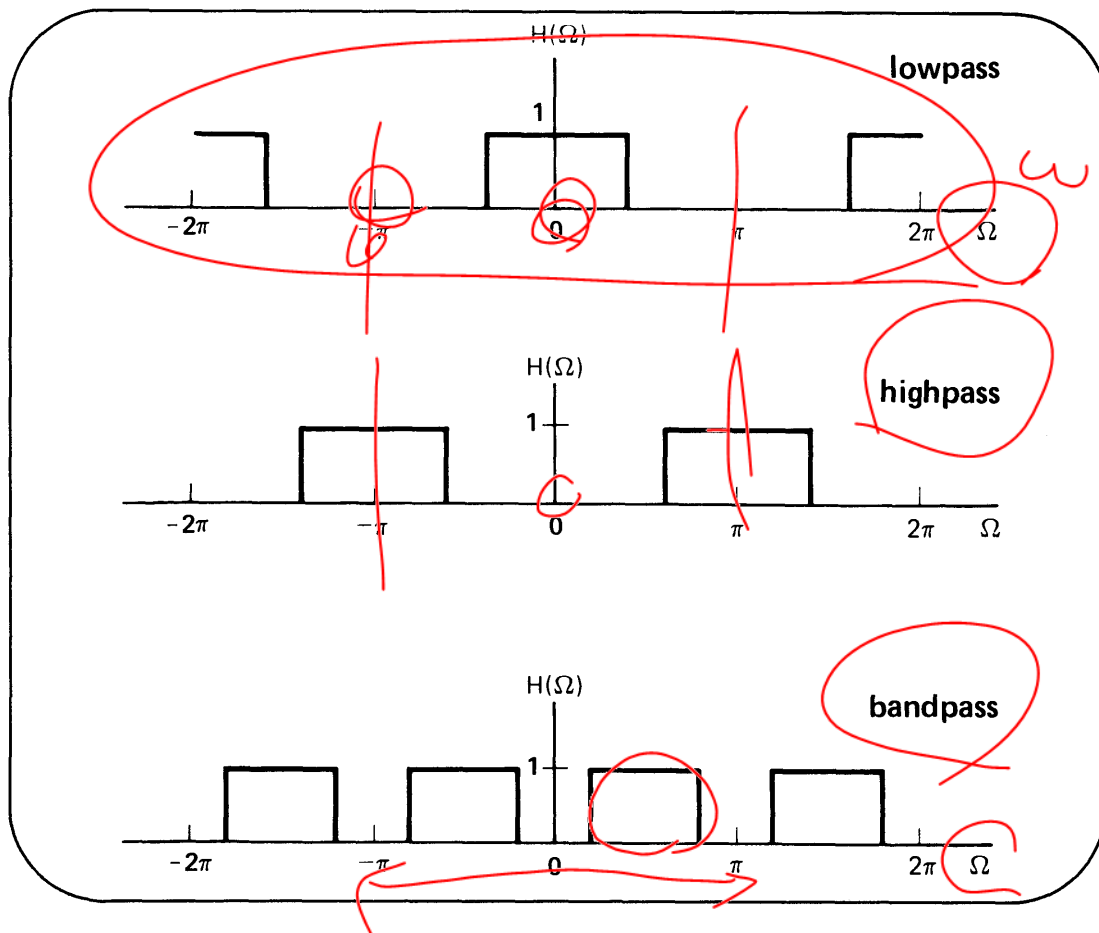


TRANSPARENCY
12.1
Frequency response of ideal lowpass, high-pass, and bandpass continuous-time filters.

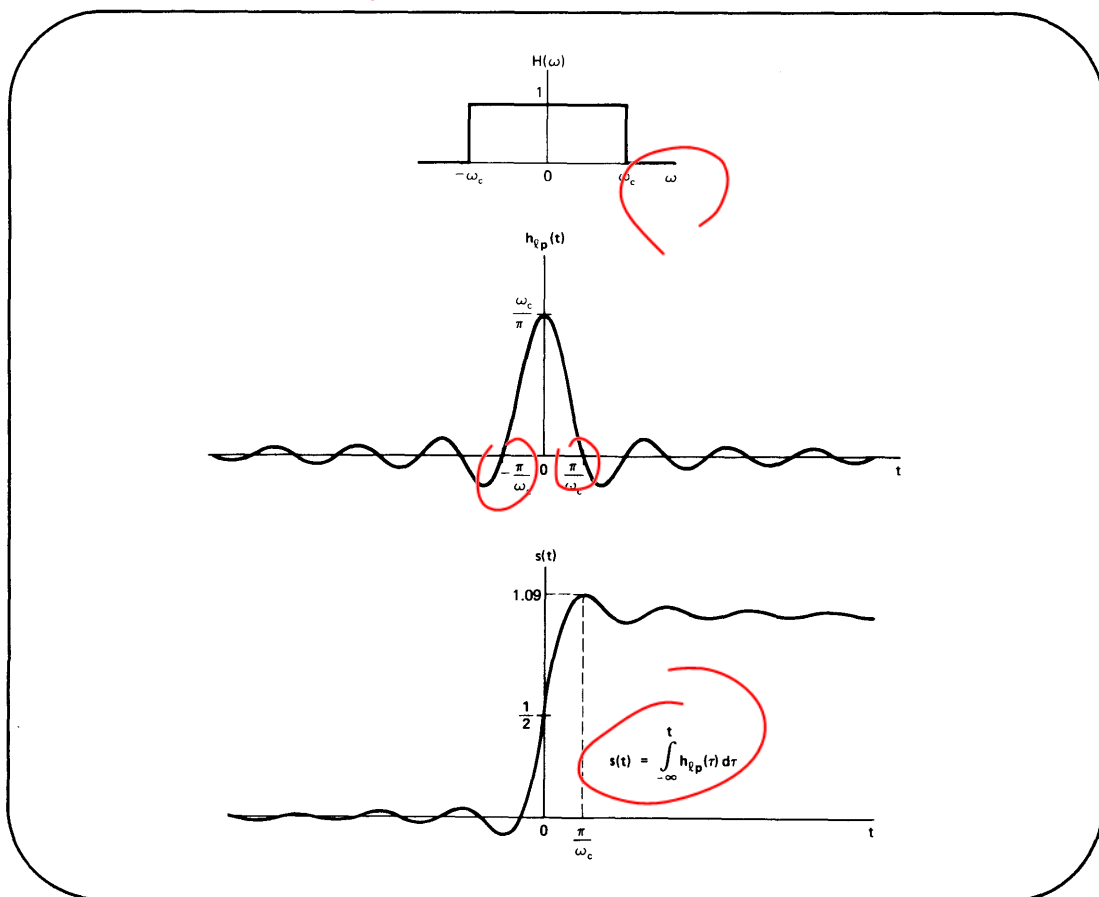


TRANSPARENCY 12.2

Frequency response of ideal lowpass, high-pass, and bandpass discrete-time filters.

**TRANSPARENCY 12.3**

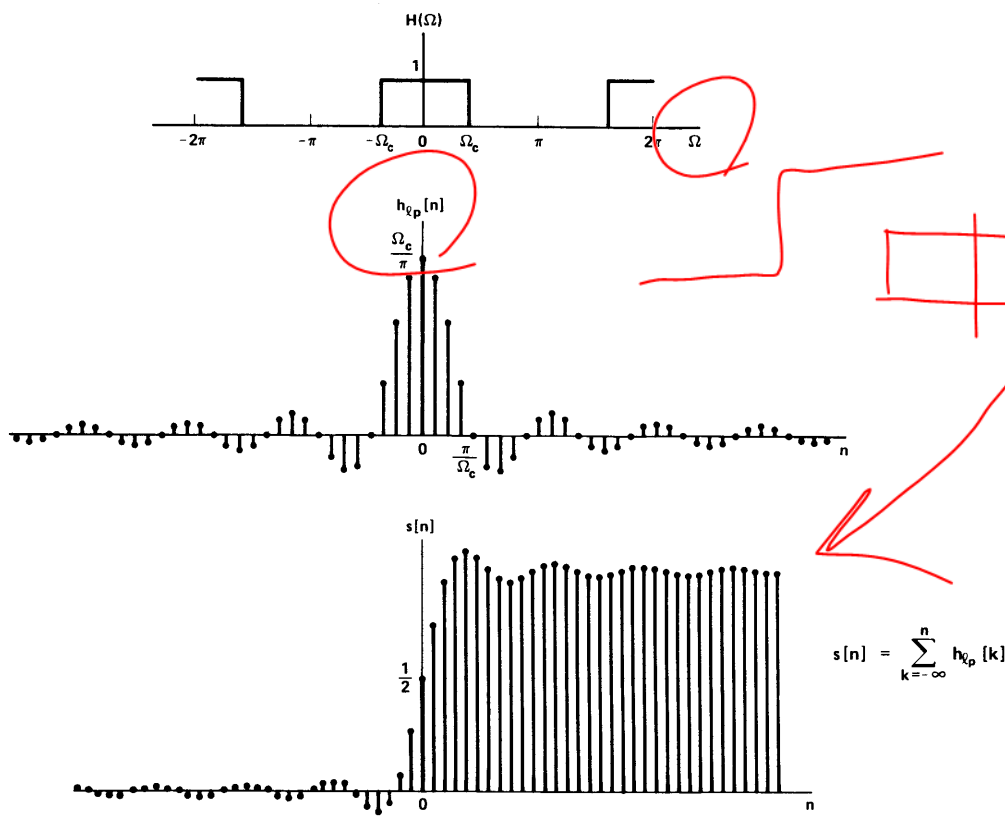
The impulse response and step response of an ideal continuous-time lowpass filter.



TRANSPARENCY

12.4

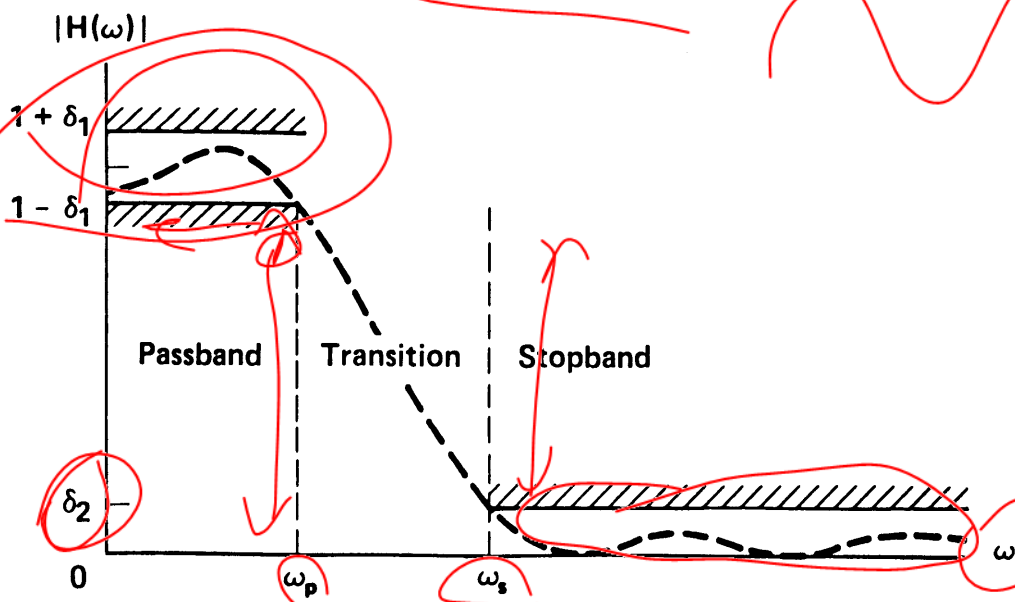
The impulse response and step response of an ideal discrete-time lowpass filter.



TRANSPARENCY

12.5

Approximation to a continuous-time lowpass filter.

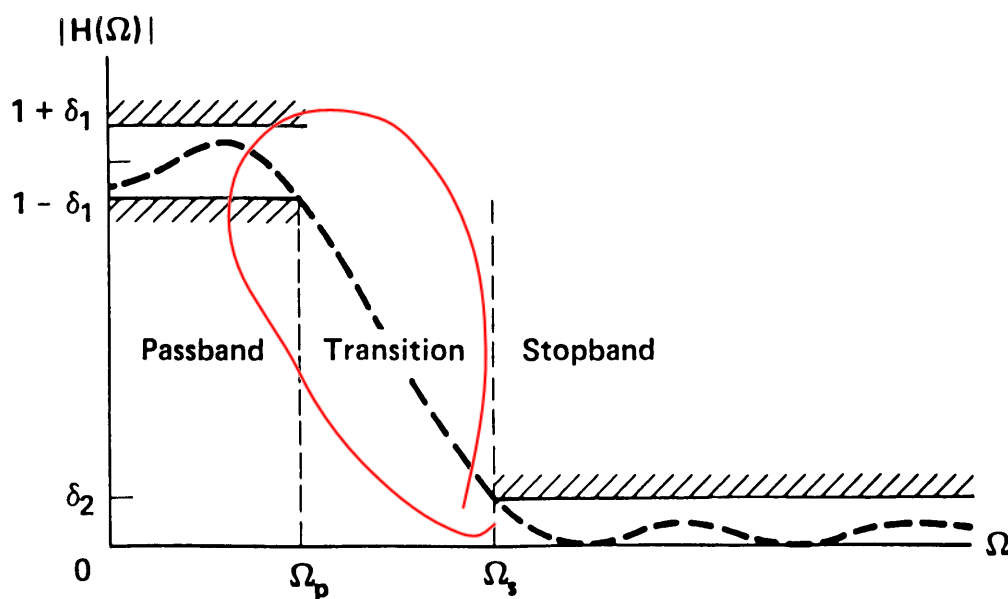


Ripple

TRANSPARENCY

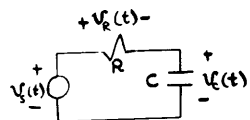
12.6

Approximation to a discrete-time lowpass filter.



MARKERBOARD

12.2



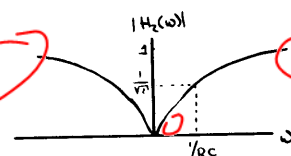
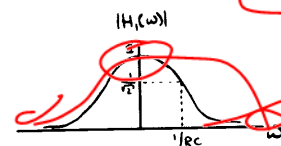
$$RC \frac{dV_c(t)}{dt} + V_c(t) = V_s(t)$$

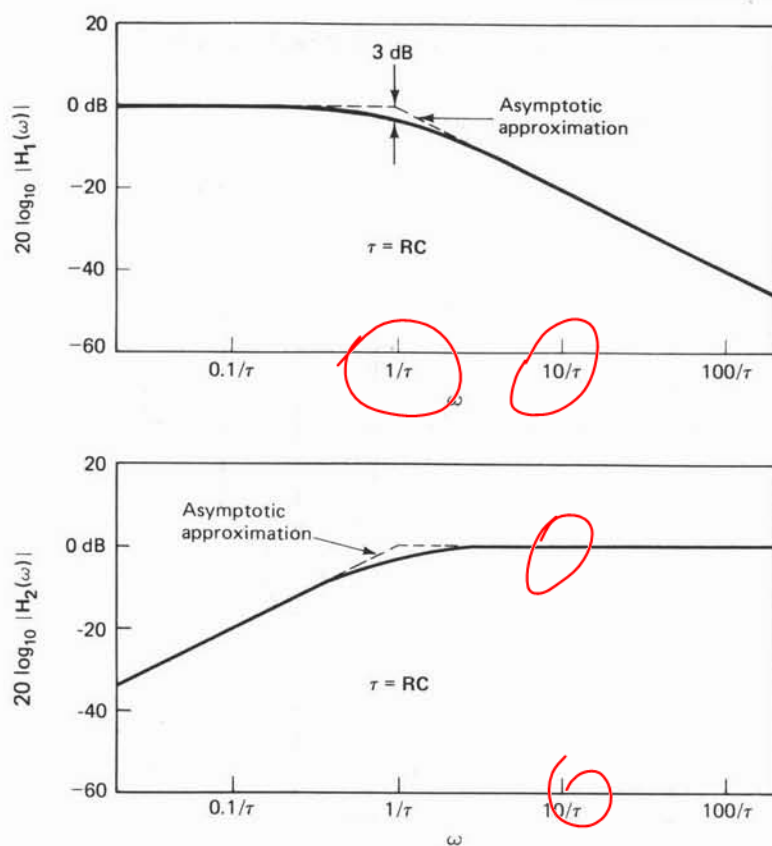
$$RC j\omega V_c(\omega) + V_c(\omega) = V_s(\omega)$$

$$\frac{V_c(\omega)}{V_s(\omega)} = H_1(\omega) = \frac{1}{1 + RCj\omega}$$

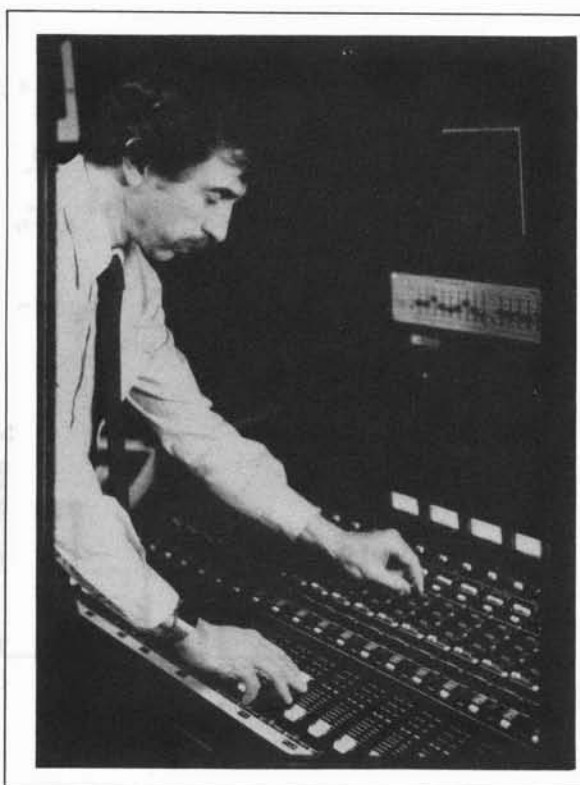
$$\frac{V_R(\omega)}{V_s(\omega)} = H_2(\omega) = 1 - H_1(\omega)$$

$$H_2(\omega) = \frac{RCj\omega}{1 + RCj\omega}$$



**TRANSPARENCY
12.7**

Bode plots for a first-order RC circuit approximation to a lowpass filter and a highpass filter.

**DEMONSTRATION
12.1**

Filtering using an audio control panel.

TRANSPARENCY

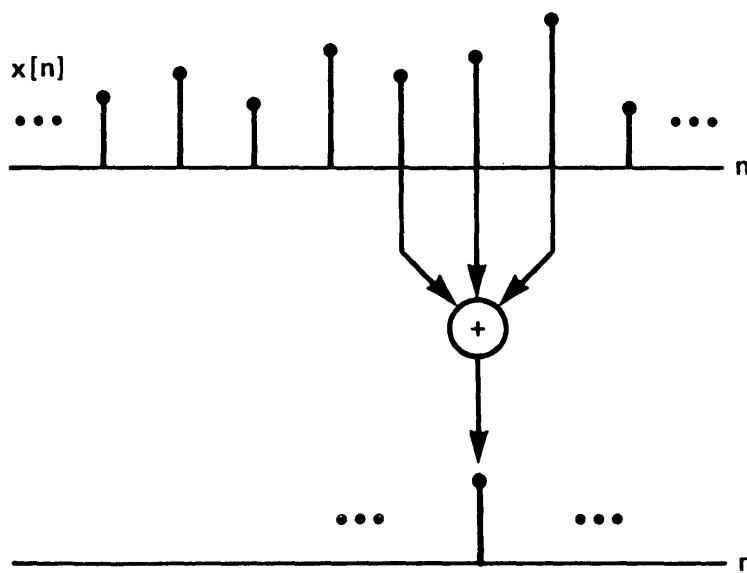
12.8

A three-point moving average discrete-time filter.

NON-RECURSIVE (MOVING AVERAGE) FILTERS

Three-point moving average:

$$y[n] = \frac{1}{3} \{x[n-1] + x[n] + x[n+1]\}$$



TRANSPARENCY

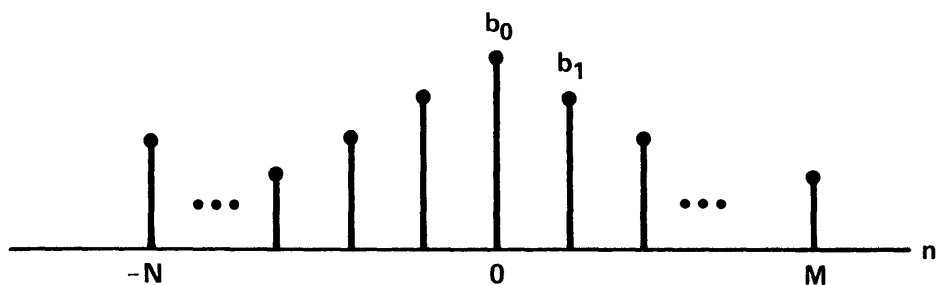
12.9

A general discrete-time moving average filter.

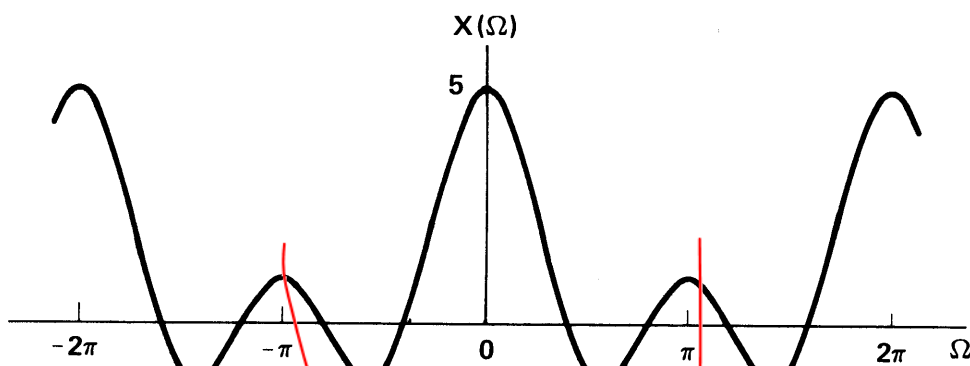
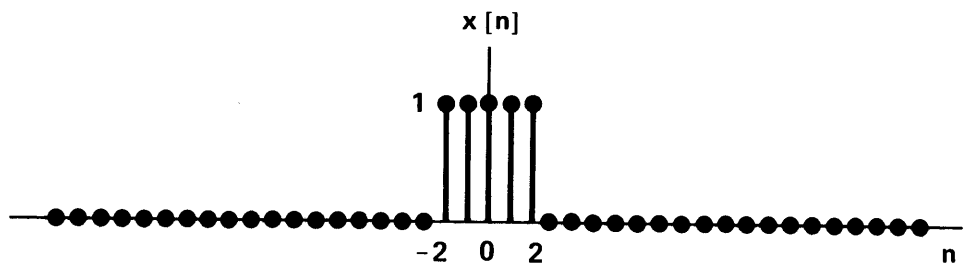
$$y[n] = \frac{1}{3} \{x[n-1] + x[n] + x[n+1]\}$$

$$y[n] = \frac{1}{N+M+1} \sum_{k=-N}^M x[n-k]$$

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

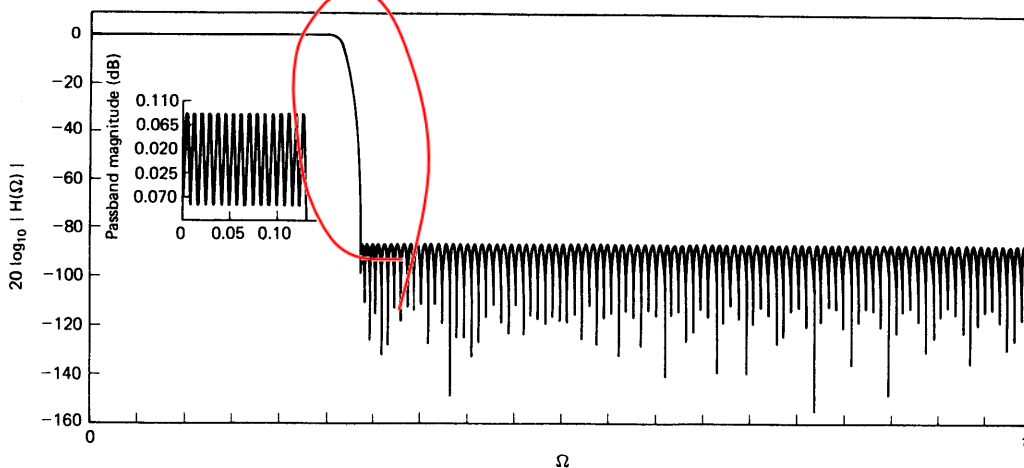


Example 5.7:

**TRANSPARENCY**

12.10

Impulse response and frequency response for a five-point moving average lowpass filter with equal weights. [Example 5.7 from the text.]

**TRANSPARENCY**

12.11

Frequency response of an optimally designed moving average filter with 256 weights.

TRANSPARENCY

12.12

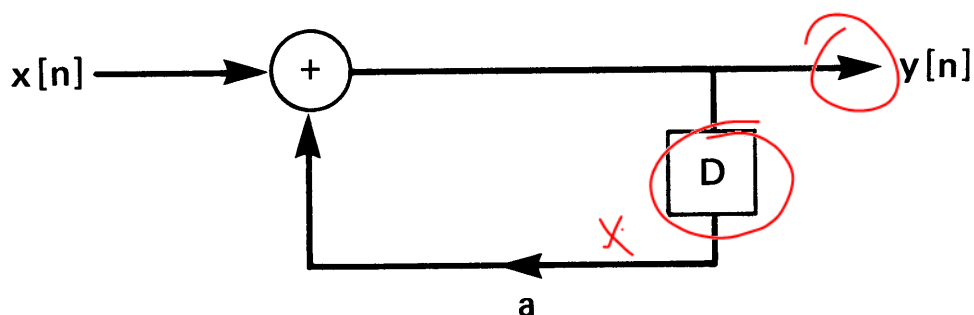
Difference equation and block diagram for a recursive discrete-time filter.

RECURSIVE DISCRETE-TIME FILTERS

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

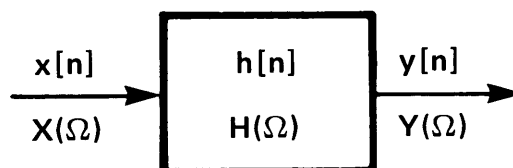
First order

$$y[n] - ay[n-1] = x[n]$$

**TRANSPARENCY**

12.13

Determination of the frequency response of a first-order system using the properties of the Fourier transform. [Example 5.5 from the text.]



$$y[n] - ay[n-1] = x[n]$$

$$Y(\Omega) - a e^{-j\Omega} Y(\Omega) = X(\Omega)$$

$$Y(\Omega) = \frac{1}{1 - a e^{-j\Omega}} X(\Omega)$$

$$H(\Omega) = \frac{1}{1 - a e^{-j\Omega}}$$

$$\underbrace{a^n u[n]}_{h[n]} \xrightarrow{\mathcal{F}} \underbrace{\frac{1}{1 - a e^{-j\Omega}}}_{H(\Omega)}$$

(Example 5.5)

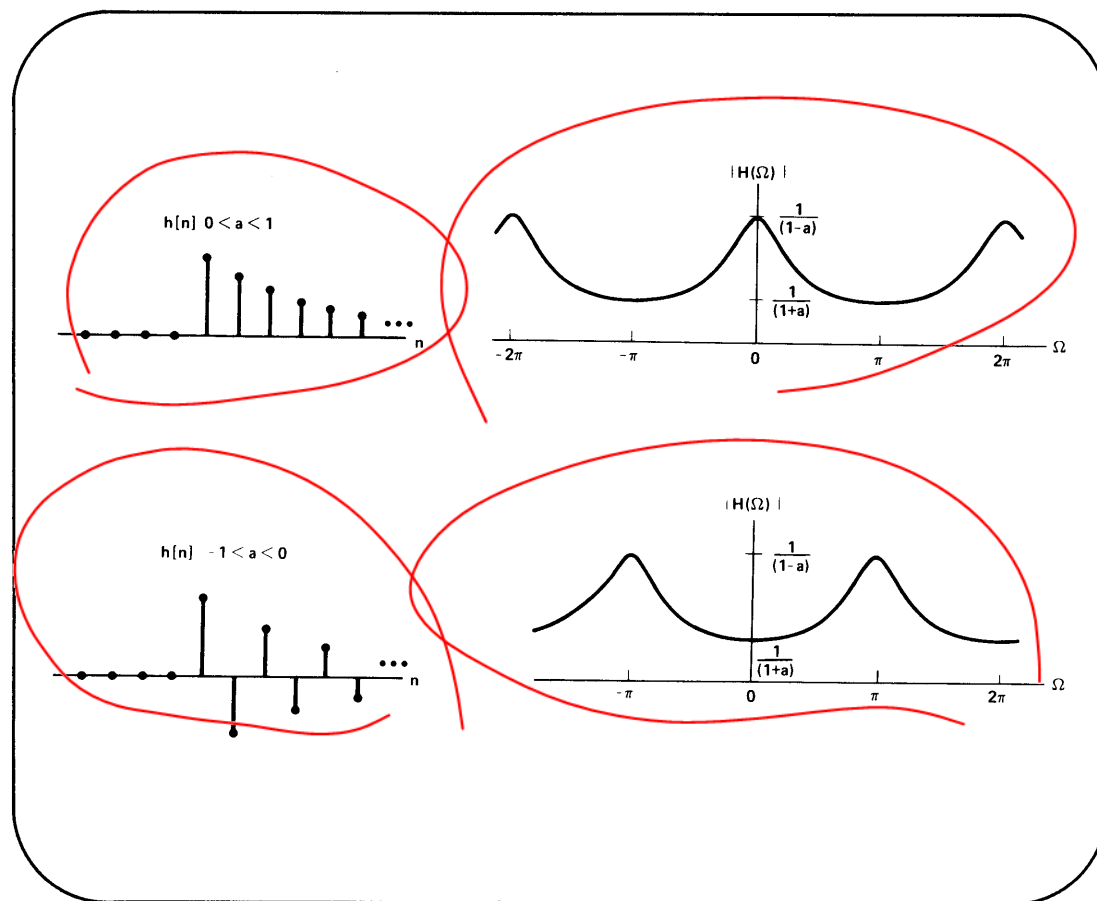
**TRANSPARENCY****12.14**

Illustration of the impulse response and frequency response for a first-order system.

MIT OpenCourseWare
<http://ocw.mit.edu>

Resource: Signals and Systems
Professor Alan V. Oppenheim

The following may not correspond to a particular course on MIT OpenCourseWare, but has been provided by the author as an individual learning resource.

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.