DIGITAL FILTERS

- Introduction
- Filters defined by linear difference equations
- Examples of digital filters designed by linear constant-coefficient difference equation (LCCDE)
- Response of LCCDE filters to unit sample
- Finite-impulse response (FIR) and infinite-impulse response (IIR) filters
- Linear time-invariant (LTI) systems
- Response of LTI system to arbitrary inputs
- Convolution and correlation
- Determining the impulse response for digital filters described by LCCDEs
- Properties of convolution
- (Convolution example)
- Causality
- Stability

Introduction

- Digital filters are used for separating signals from noise and for frequency analysis, an operation which often reveals important features in the signal
- They typically “pass” or amplify certain frequency components of the signal, while they “stop” or attenuate others

Filters defined by linear difference equations

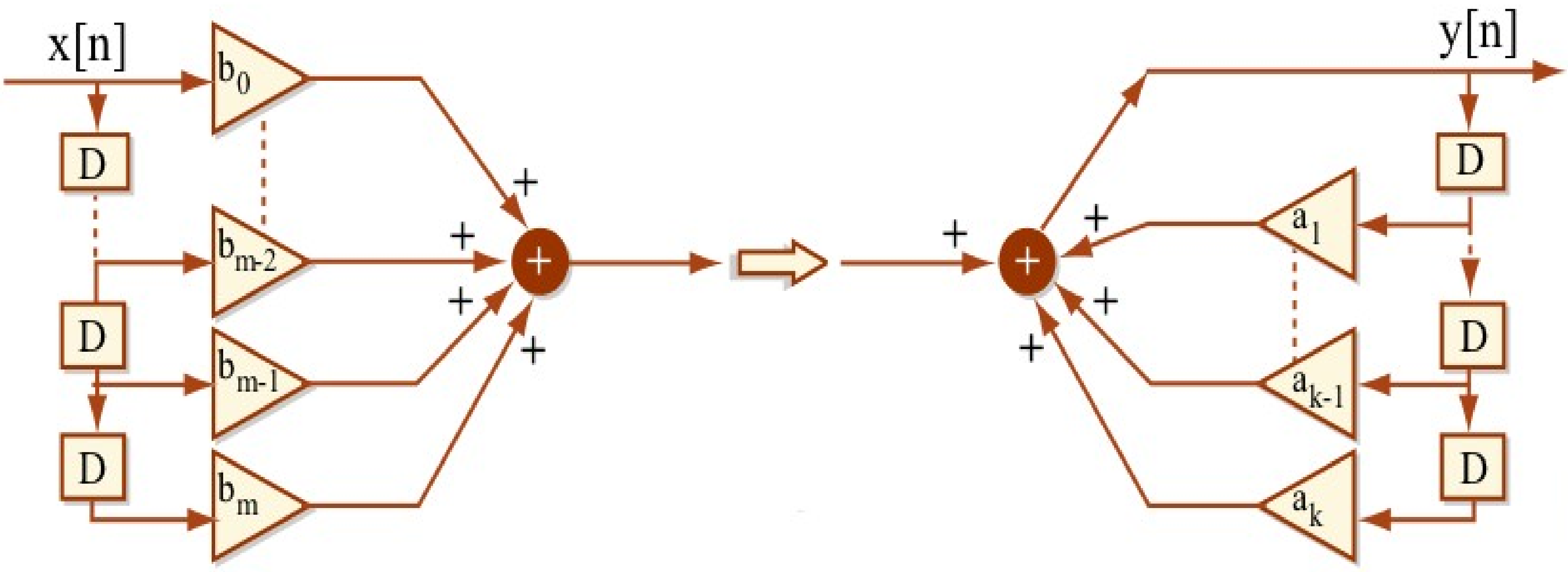
- A *discrete-time system* is any mathematical transformation that maps a discrete-time input signal $x[n]$ into an output signal $y[n]$
- Discrete-time systems defined by a *linear, constant-coefficient difference equation (LCCDE)* constitute an important class of digital filters:

$$y[n] = \sum_{k=1}^{K} a_k y[n-k] + \sum_{m=0}^{M} b_m x[n-m]$$

- The maximum of the numbers M and K is called the order of the filter
- If the input signal is defined for $n >= n0$, then values of both the input and output for a time prior to $n0$ must be known. $y[n]$ must be known for $n0 - K <= n <= n0 - 1$, and $x[n]$ for $n0 - M <= n <= n0$

Filters defined by linear difference equations

- Block-diagram representation of general difference equation



(Bertrand Delgutte, MIT OpenCourseWare)

Examples of digital filters designed by linear constant-coefficient difference equation (LCCDE)

1. Simple gain, or amplifier:

$$y[n] = Gx[n]$$

2. Delay of n_0 samples:

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

3. Two-point moving average:

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

4. Euler's formula for approximating the derivative of a continuous-time function:

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

where T_s is the sampling interval.

Examples of digital filters designed by linear constant-coefficient difference equation (LCCDE)

5. Averaging over N consecutive epochs of duration L:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n - kL]$$

6. Trapezoidal integration formula:

$$y[n] = \frac{y[n-1] + (x[n] + x[n-1])T_s}{2}$$

7. Digital "leaky integrator", or first-order lowpass filter:

$$y[n] = ay[n-1] + x[n] \qquad 0 < a < 1$$

8. Digital resonator:

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + bx[n] \qquad a_1^2 + 4a_2 < 0$$

This is the digital equivalent of the harmonic oscillator.

Response of LCCDE filters to unit sample

- The response of an LCCDE filter, *y*[*n*], to an arbitrary signal x[n], can be completely characterized by its response to one particular signal, the unit sample, *δ*[*n*]. The response to *δ*[*n*] is denoted *h*[*n*].

$$\delta[n] \triangleq \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0. \end{cases}$$

- If $a_k = 0$, the response *h*[*n*] to the unit sample, *δ*[*n*], is of finite duration (*FIR filters*, Finite-Impulse Response filters, non-recursive filters)
- If $a_k \neq 0$, the response *h*[*n*] to the unit sample, *δ*[*n*], is of infinite duration (*IIR filters*, Infinite-Impulse Response filters, recursive filters)

$$y[n] = \sum_{k=1}^{K} a_k y[n-k] + \sum_{m=0}^{M} b_m x[n-m]$$

Finite-impulse response (FIR) and infinite-impulse response (IIR) filters

- FIR filters. If all the a_k coefficients are zero, then the output depends only on a finite number of values of the input. Termed also as *all-zero*, or *moving average (MA) filters*. (Examples 1 – 5 above)
- IIR filters. If at least one of the a_k coefficients is nonzero:

  (a) Autoregressive (AR) filters. If all of the b_m coefficients except b_0 are zero, the output depends only on the current value of the input and a finite number of past values of the output. Termed also as *all-pole*, *purely recursive*, or *autoregressive (AR) filters*. The term “autoregressive” means that the output is approximately a sum of its own past values. (Examples 7 and 8 above)

  (b) Autoregressive, moving-average (ARMA) filters. Both a_k and b_m coefficients are nonzero, with $K \geq 1$ and $M > 0$. Also termed as *pole-zero* or *autoregressive, moving average (ARMA) filters*. (Example 6 above)

$$y[n] = \sum_{k=1}^{K} a_k y[n-k] + \sum_{m=0}^{M} b_m x[n-m]$$

Linear time-invariant (LTI) systems

- Linearity

  (a) Superposition. If the response of discrete-time system to $x_1[n]$ is $y_1[n]$, and the response to $x_2[n]$ is $y_2[n]$, then the response to $x_1[n] + x_2[n]$ is $y_1[n] + y_2[n]$.

  (b) Scaling. If the response of a discrete-time system to $x[n]$ is $y[n]$, then the response to $c.x[n]$ is $c.y[n]$, where c is a real or complex constant.
- Time invariance. If the response of a discrete-time system to $x[n]$ is $y[n]$, then the response to $x[n-n_0]$ (input $x[n]$ delayed by n_0 samples) is $y[n-n_0]$ (the original response delayed by n_0 samples).
- Both, FIR and IIR filters defined by a linear, constant-coefficient difference equation are LTI systems
- Median filters are *nonlinear*, but time-invariant
- Adaptive filters are discrete-time systems for which the filter coefficients a_k and b_m vary with time (or n) to meet certain performance criteria. *They are neither linear, nor time-invariant*.

Response of LTI systems to arbitrary inputs

- The response of an LTI system, $y[n]$, to an arbitrary signal $x[n]$, can be completely characterized by its response to one particular signal, the unit sample, $\delta[n]$:

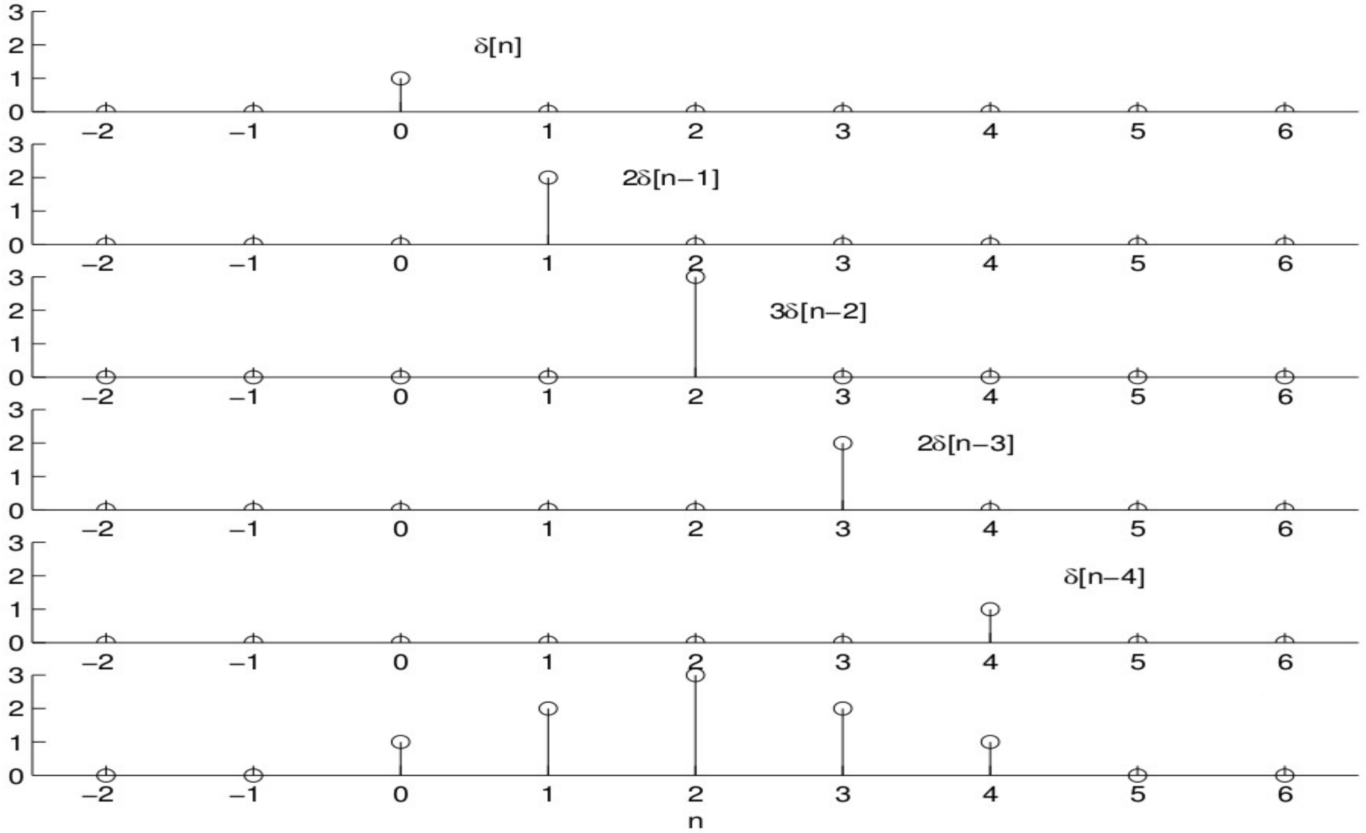
$$\delta[n] \triangleq \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0. \end{cases}$$

- The key to prove this property is to write the signal $x[n]$ as a weighted sum of delayed unit samples:

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

Response of LTI systems to arbitrary inputs

- The decomposition of a triangular signal, $x[n]$, into a sum of unit samples



(Bertrand Delgutte, MIT OpenCourseWare)

Response of LTI systems to arbitrary inputs

- Let the $h[n]$ be the response of an LTI system to the unit sample $\delta[n]$
  (a) By the time-invariance property: the response to $\delta[n-m]$ must be $h[n-m]$
  (b) By the scaling property: the response to $x[m].\delta[n-m]$ is $x[m].h[n-m]$.
  Note that $x[m]$ is considered to be a constant weighting factor for the delayed unit sample $\delta[n-m]$ because it does not depend on the index n
  (c) By the superposition principle: the response of an LTI system, $y[n]$, to $x[n]$ can be written as a weighted sum of the $h[n-m]$:

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \triangleq x[n] * h[n]$$

- This expression is by definition the *discrete convolution of* $x[n]$ *with* $h[n]$ ($x[n] * h[n]$)
- If we know the response of an LTI system (denoted $h[n]$) to a unit sample $\delta[n]$, then we can determine the response of that system, $y[n]$, to any arbitrary input $x[n]$
  (This does not hold for nonlinear and time-varying systems)

Convolution and correlation

Convolution

Origin f w rotated 180°

0 0 0 1 0 0 0 0   8 2 3 2 1

        0 0 0 1 0 0 0 0
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
  8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
        8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
                      8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0

Cropped convolution result

0 1 2 3 2 8 0 0

Correlation

Origin f w

0 0 0 1 0 0 0 0   1 2 3 2 8

        0 0 0 1 0 0 0 0
1 2 3 2 8

Starting position alignment

Zero padding

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
  1 2 3 2 8

Position after one shift

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
        1 2 3 2 8

Position after four shifts

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
                      1 2 3 2 8

Final position

Full correlation result

0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result

0 8 2 3 2 1 0 0

(Gonzales, Woods)

Determining the impulse response for digital filters described by LCCDEs

- The response of an LTI system, $y[n]$, to any signal $x[n]$ can be computed if the system's unit-sample response, or impulse response, $h[n]$ is known
- For FIR filters, the unit-sample response can be found by inspection from the b_m coefficients:

$$h[m] = \begin{cases} b_m & \text{if } 0 \leq m \leq M \\ 0 & \text{otherwise} \end{cases}$$

Determining the impulse response for digital filters described by LCCDEs

- The unit-sample responses of the previous FIR filter examples

1. Gain:

$$h[n] = G\delta[n]$$

2. Delay:

$$h[n] = \delta[n - n_0]$$

3. Two-point moving average:

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

4. Euler's approximation to the derivative:

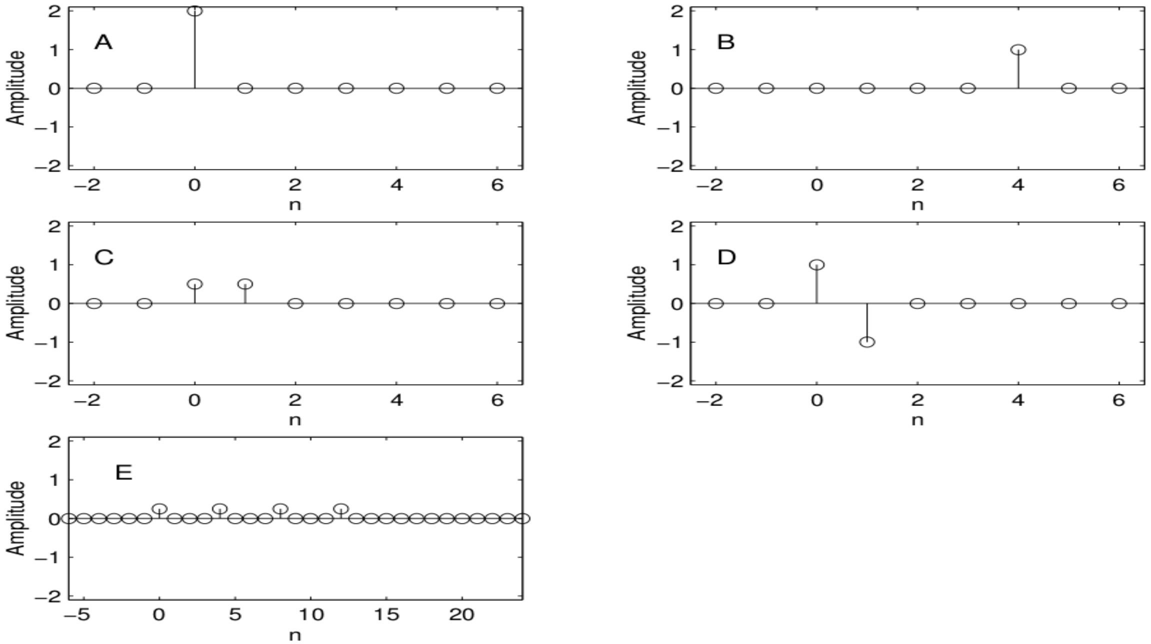
$$h[n] = (\delta[n] - \delta[n-1])/T_s$$

5. Averager:

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1}\delta[n - kL]$$

Determining the impulse response for digital filters described by LCCDEs

- The unit-sample responses $h[n]$ of the previous FIR filter examples



Unit sample responses of simple FIR filters (A) Gain with $G = 2$. (B) Delay with $n_0 = 4$. (C) Two-point moving average. (D) Euler's approximation to the derivative with $T_s = 1$. (E) Averager with $N = 4$ and $L = 6$.

(Bertrand Delgutte, MIT OpenCourseWare)

Properties of convolution

- Convolution is a commutative operation

$$x[n] * h[n] = h[n] * x[n]$$

- Convolution is an associative operation

$$x[n] * (h1\,[n] * h2\,[n]) = (x[n] * h1\,[n]) * h2\,[n]$$

- Convolution is distributive over addition

$$(x[n] * h1\,[n]) + (x[n] * h2\,[n]) = x[n] * (h1\,[n] + h2\,[n])$$

(Convolution example)

- We will consider the response of the first-order low-pass filter to a rectangular pulse of duration *N*:

$$x[n] = u[n] - u[n-N] = \begin{cases} 1 & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

- Where *u*[*n*] is the unit step, defined by:

$$u[n] \triangleq \begin{cases} 0 & \text{if } n < 0 \\ 1 & \text{if } n \geq 0 \end{cases}$$

- The unit-sample response of the filter (exponential unit-sample response):

$$h[n] = a^n u[n]$$

- Convolution:

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} a^m u[m] x[n-m] = \sum_{m=0}^{\infty} a^m x[n-m]$$

(Convolution example)

- Three regions must be distinguished:
  1. For *n* < *0*, *x*[*n* − *m*] is equal to zero for *m* ≥ *0*, so that *y*[*n*] = *0*. This is generally true if both *x*[*n*] and *h*[*n*] are zero for negative times.
  2. For *0* ≤ *n* ≤ *N* − *1*, the sum is from *m* = *0* to *n* because *x*[*n* − *m*] is zero for *m* > *n*. Therefore:

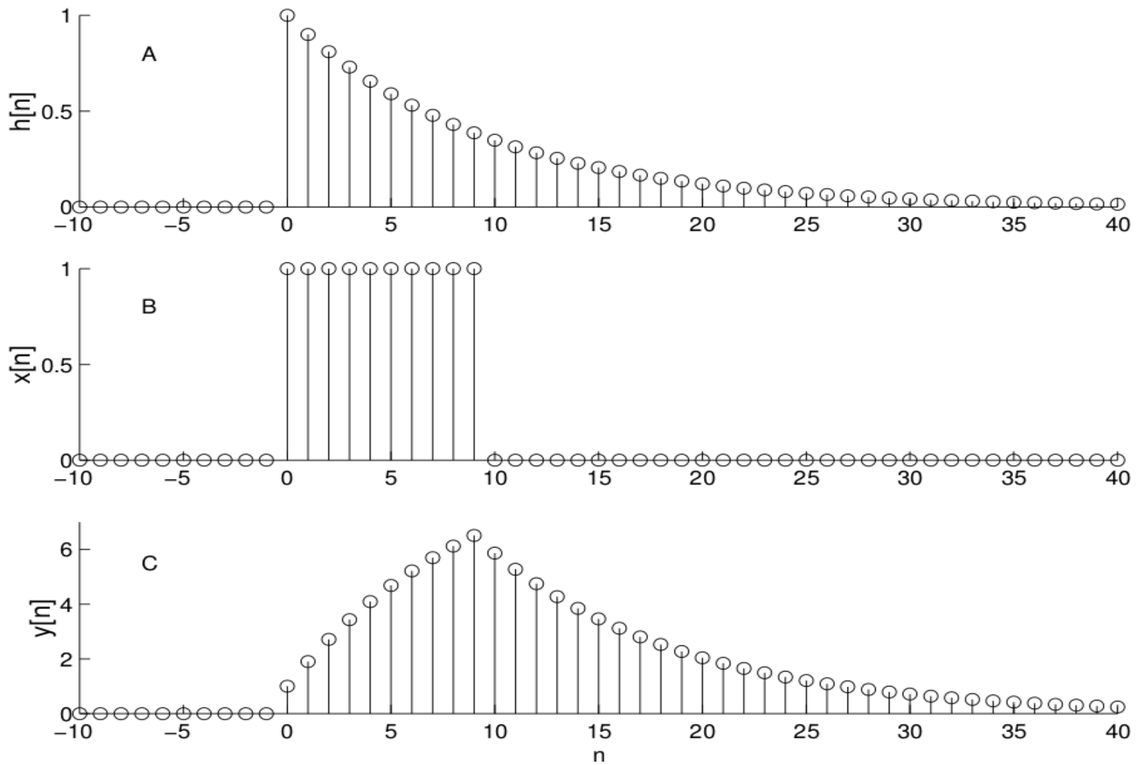
     $$y[n] = \sum_{m=0}^{n} a^m = \frac{1 - a^{n+1}}{1 - a}$$

     In this range, *y*[*n*] exponentially approaches the asymptote *1 / (1 – a)*
  3. For *n* ≥ *N*, *x*[*n* − *m*] is zero outside of the interval *n* − *N* + *1* ≤ *m* ≤ *n* :

$$y[n] = \sum_{m=n-N+1}^{n} a^m = a^n - N + 1 \frac{1 - a^N}{1 - a} = \frac{1 - a^- N}{1 - a^- 1} a^n$$

The *y*[*n*] exponentially decays to zero.

(Convolution example)



Convolution example: (A) Unit-sample response of the first-order low-pass filter, $h[n] = a^n u[n]$, with $a = 0.9$. (B) Input signal, $x[n] = u[n] - u[n-N]$, with $N = 10$. (C) Output signal.

(Bertrand Delgutte, MIT OpenCourseWare)

Causality

- A discrete-time system is said to be causal if its response at time *n0* depends only on the input for times $n \leq n0$
- Causality is necessary for processing signals in real time (control applications)
- When the signal has been stored prior to processing, the notions of "past" and "future" become largely a matter of convention, and it is possible to use non-causal filters
- Causality is of little relevance for signals where the independent variable is not time, such as digital images
- An example of non-causal FIR filter: $$y[n] = \sum_{m=-M}^{M} b_m x[n-m]$$
- FIR filters of this form whose unit-sample response is symmetric around the origin, i.e., $b_m = b_{(-m)}$, are referred to as zero-phase, since they introduce no delay in the processing
- A zero-phase FIR filter can be changed into a causal filter by shifting its unit-sample response by half the duration of the unit-sample response (linear phase, delay)

Stability

- A system is said to be stable if a bounded input gives a bounded output
- For an LTI system to be stable, it is necessary and sufficient that its unit-sample response be absolutely summable:

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

- FIR filters are always stable
- IIR filters are not necessarily stable; for example, the first-order low/pass filter, $y[n] = a\,y[n-1] + x[n]$, is unstable if $|a| \geq 1$ because its unit-sample response, $h[n]$:

$$a^n u[n]$$

is not absolutely summable.