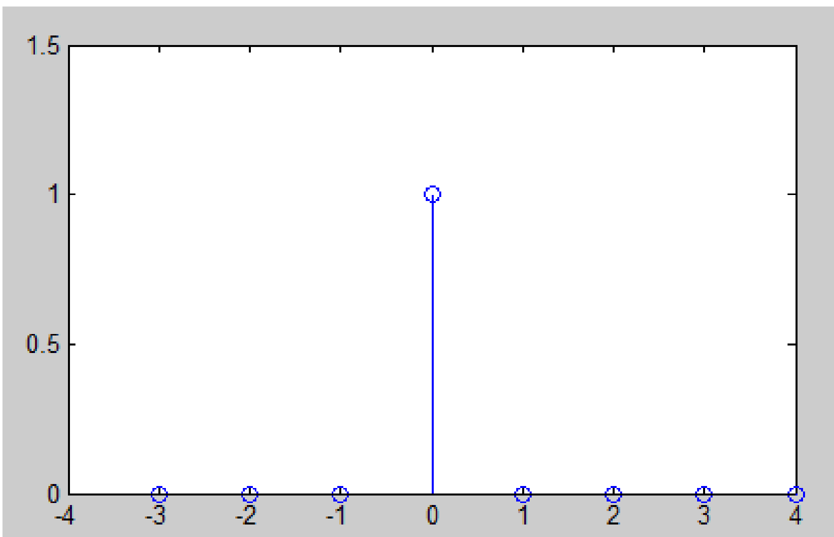
Elementary Discrete-Time signals

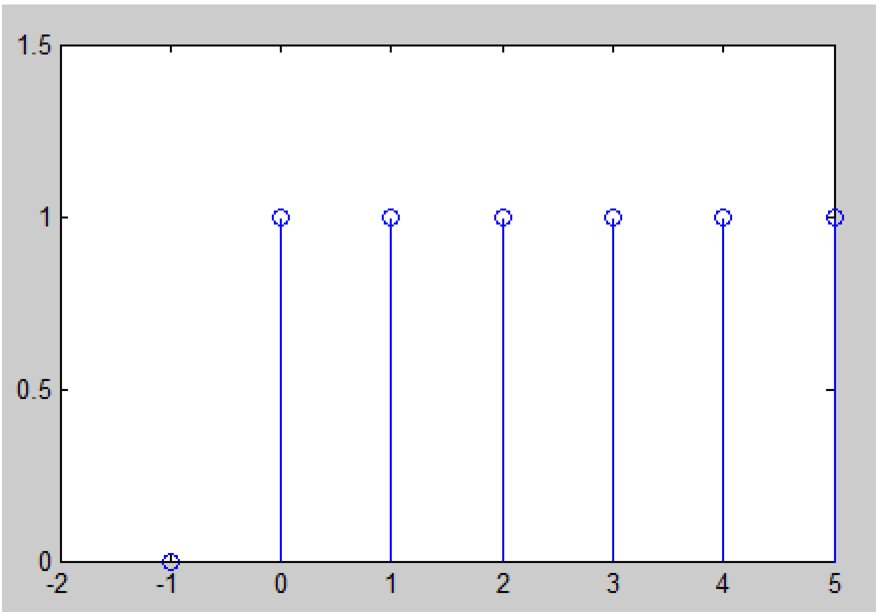
1. Unit Impulse

$$\delta(n) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0 \end{cases}$$



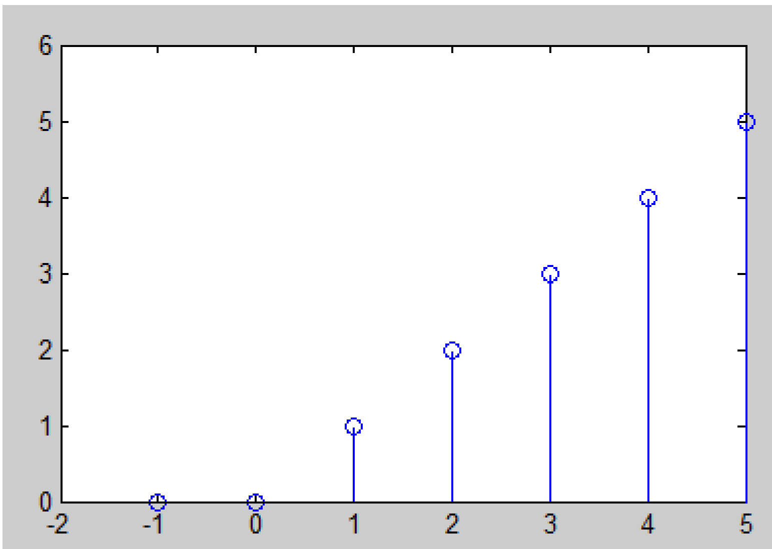
Unit step signal

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



Unit ramp signal

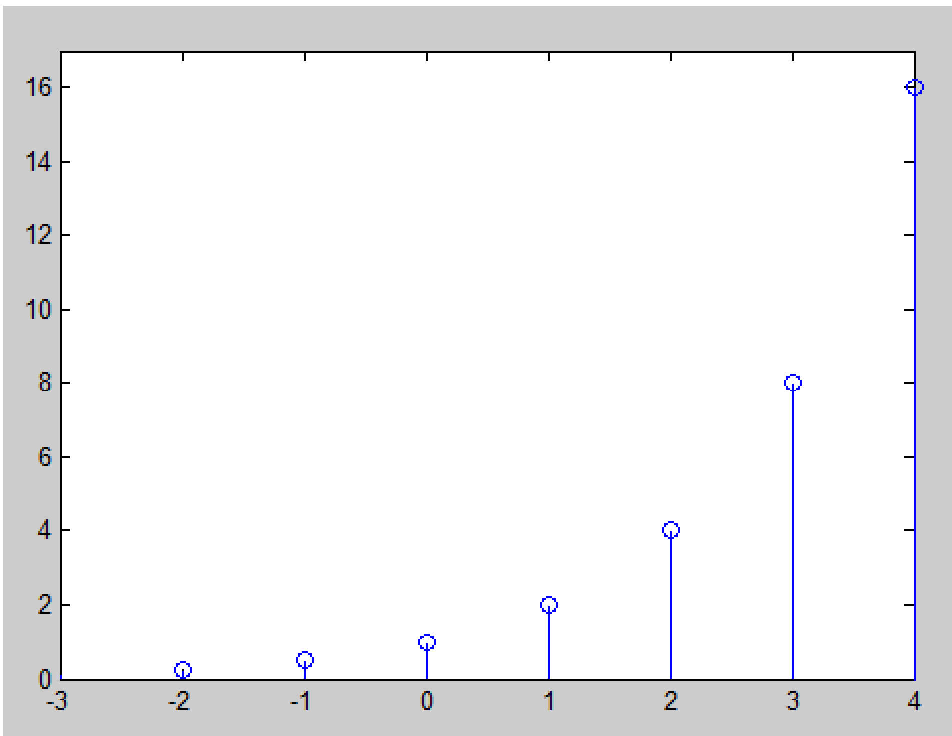
$$u_r[n] = \begin{cases} n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$$



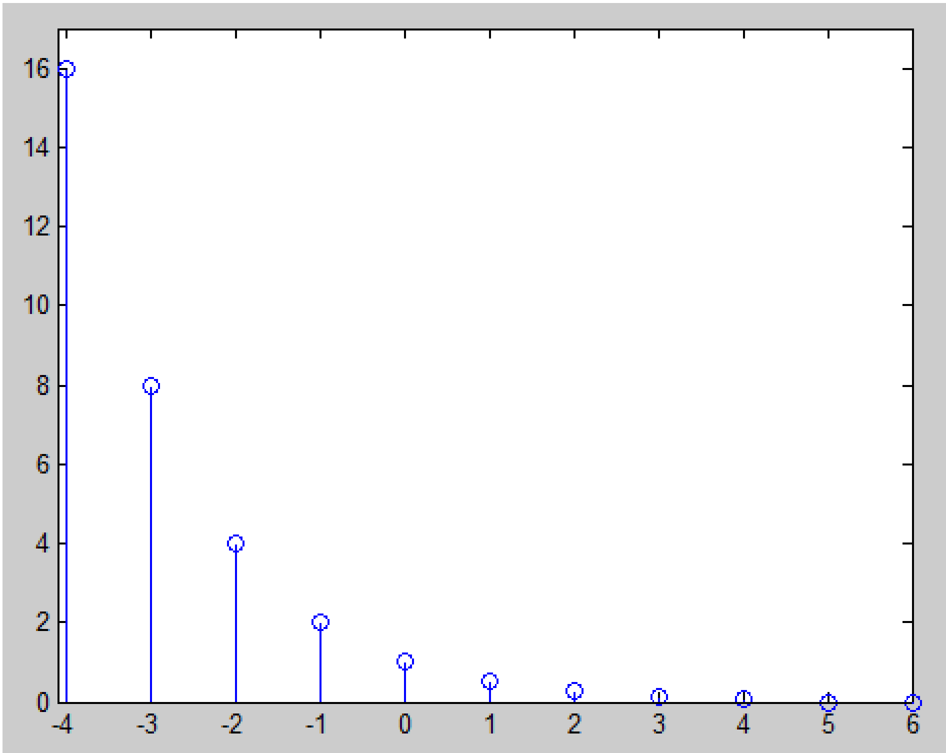
Exponential Signal

$$x[n] = a^n$$

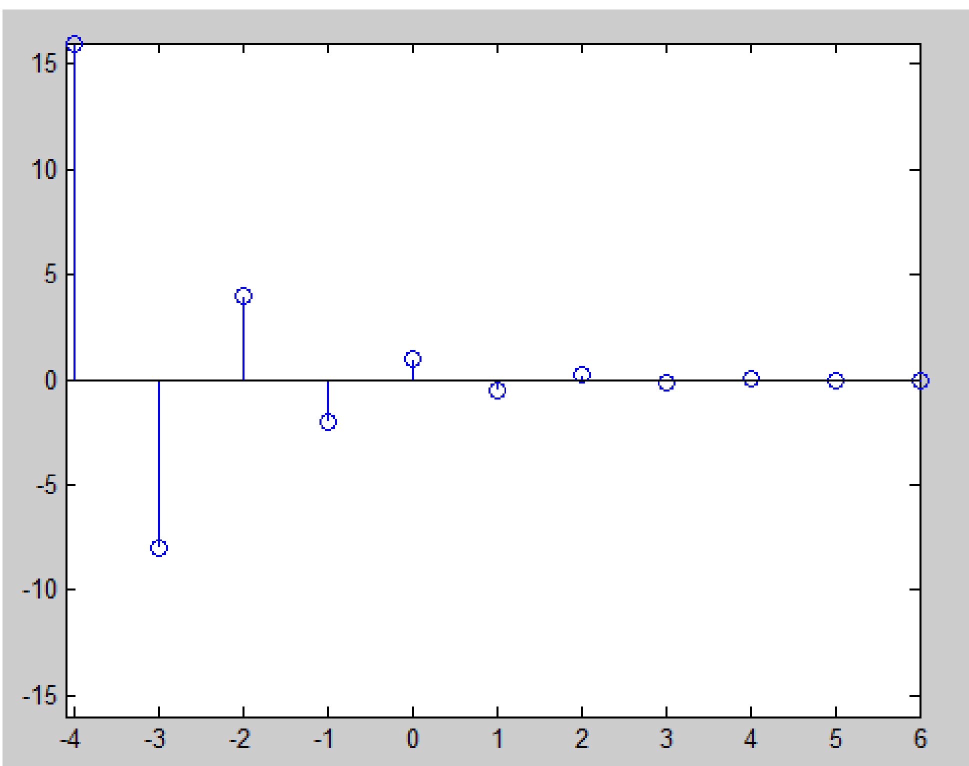
if a is real then $x[n]$ is real



2^n



$\left(\frac{1}{2}\right)^n$



$\left(-\frac{1}{2}\right)^n$

$x[n] = a^n$ if a is complex

$a = re^{j\theta}$

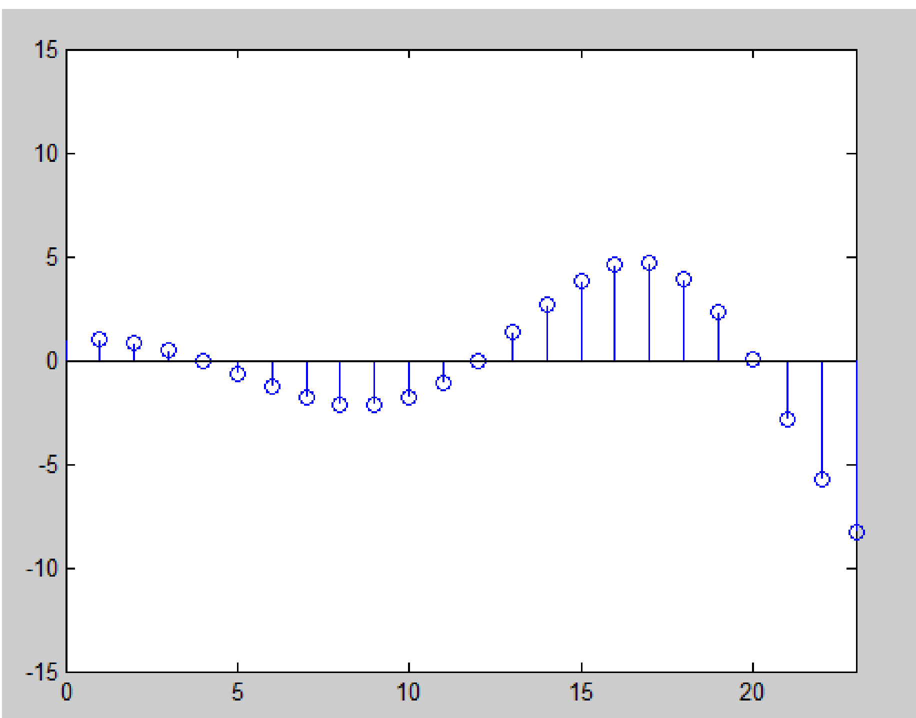
$x[n] = r^n e^{jn\theta}$

$= r^n [\cos n\theta + j \sin n\theta]$

$= r^n \cos n\theta + j r^n \sin n\theta$

$x_R[n] = r^n \cos n\theta$

$x_I[n] = r^n \sin n\theta$



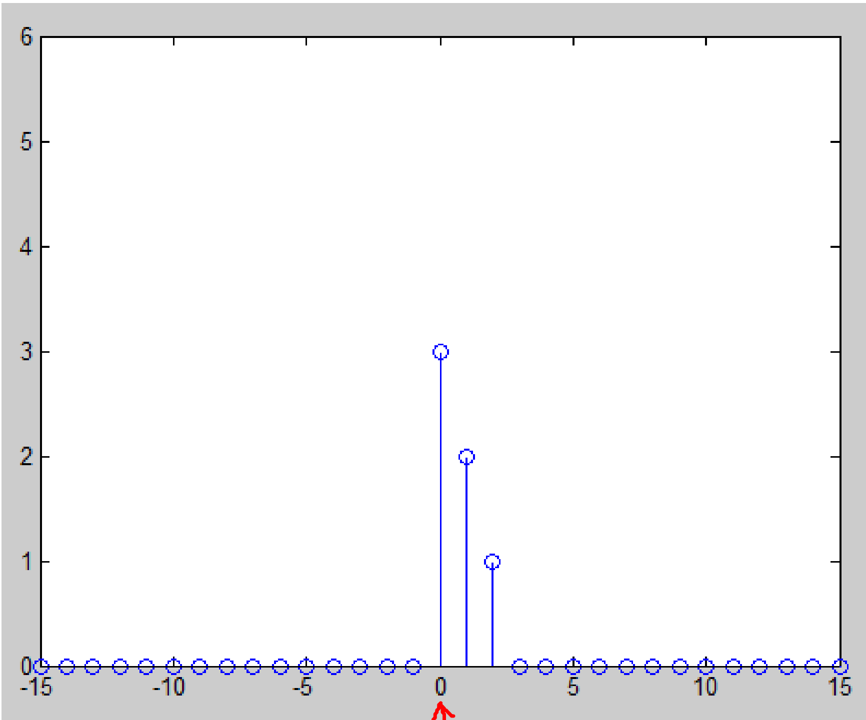
$x[n] = a^n$

$a = 1.1 \angle \frac{\pi}{8}$

$x_R[n]$

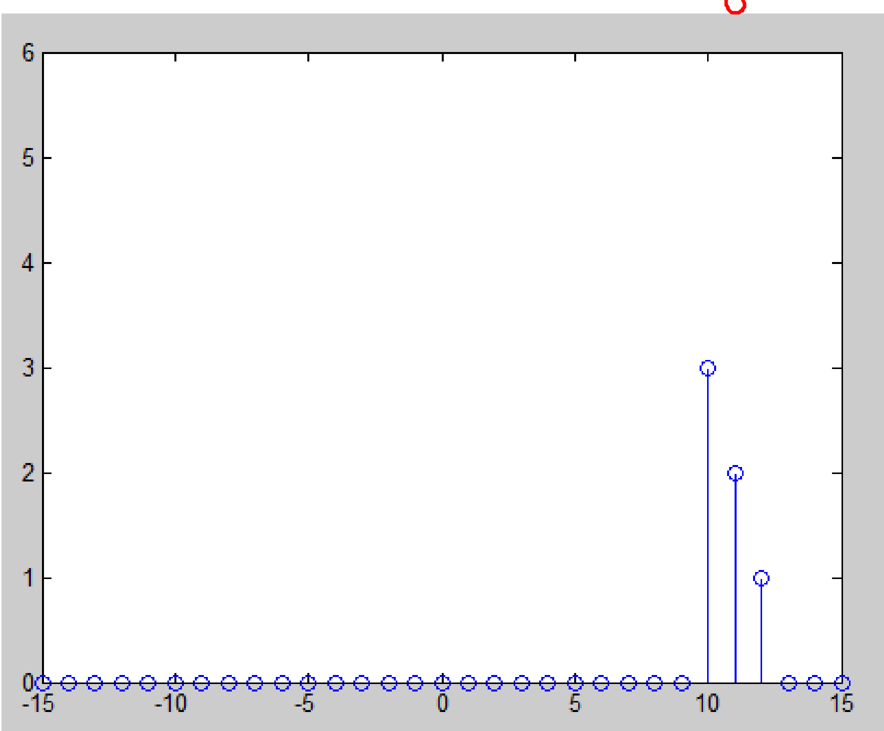
Time delay $\quad x[n-n_d] \quad n_d > 0$

$x[n]$



advance $\longleftarrow$ now $\longrightarrow$ delay

$x[n-10]$



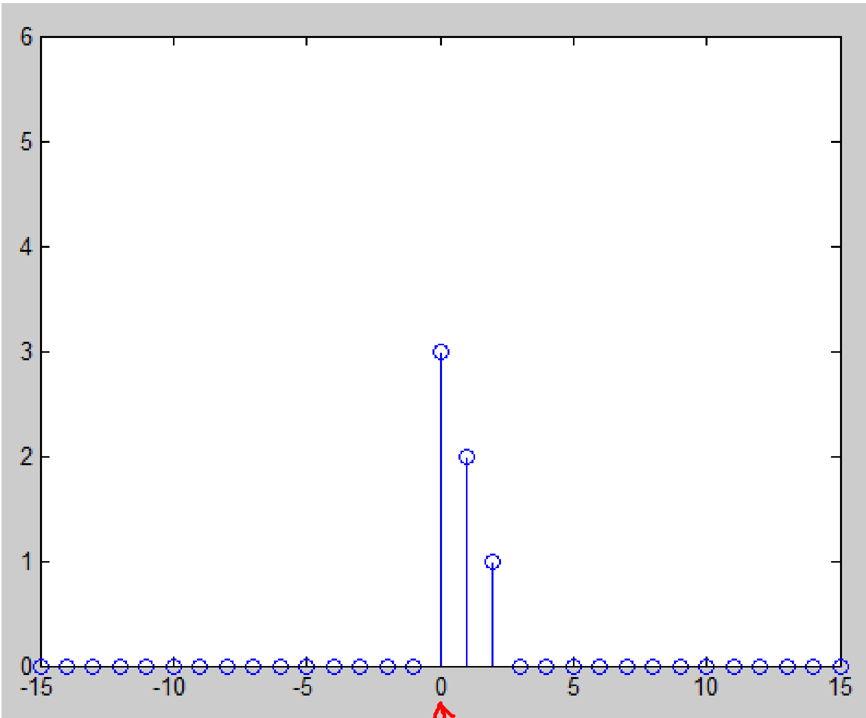
$x[n-n_0]$

delay if $n_0 > 0$

Delay

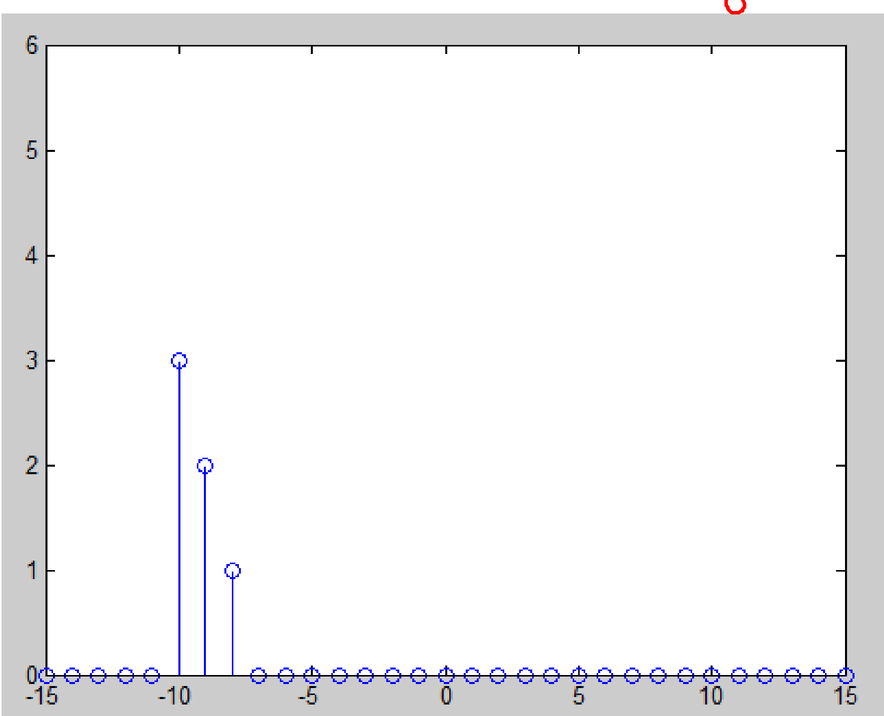
Time advance $x[n+n_d]$ $n_d > 0$

$x[n]$



advance ← now → delay

$x[n+10]$



$x[n+n_0]$

advance if $n_0 > 0$

$\underbrace{\hspace{4em}}_{\text{advance}}$

input output

$$x[n] \longrightarrow \boxed{\text{Discrete System}} \longrightarrow y[n]$$

Discrete systems are described by difference equations

In this course we will deal with discrete systems that are described by linear difference equations with constant coefficients

e.g. $y[n] = a\,y[n-1] + x[n]$ ← linear difference equation

constant coefficient

order

$$y[n] = -\sum_{k=1}^{N} a_k\, y[n-k] + \sum_{k=0}^{M} b_k\, x[n-k]$$

constant coefficients

$$\sum_{k=0}^{N} a_k\, y[n-k] = \sum_{k=0}^{M} b_k\, x[n-k] \qquad a_0 = 1$$

a_k, b_k define the system

output is to be determined

input is known

$$y[n] = -\sum_{k=1}^{N} a_k\, y[n-k] + \sum_{k=0}^{M} b_k\, x[n-k]$$

output at time n

depends on

output at times $n-1, n-2, \cdots$

input at times $n-1, n-2, \cdots$

output depends on past values of output

→ memory, state

Example

unknown known known

$$y[n] = a\,y[n-1] + x[n]$$

solve recursively for $n \geq 0$

recursion

$$y[0] = a\,y[-1] + x[0]$$
$$y[1] = a\,y[0] + x[1]$$
$$y[2] = a\,y[1] + x[2]$$

Besides a and $x[n]$ we also need $y[-1]$ to get a solution

$y[-1]$ is known as the initial condition

$$y[1] = a\{a\,y[-1] + x[0]\} + x[1] = a^2 y[-1] + a\,x[0] + x[1]$$

$$y[2] = a\{a^2 y[-1] + a\,x[0] + x[1]\} + x[2]$$
$$= a^3 y[-1] + a^2 x[0] + a\,x[1] + x[2]$$

$$y[n] = a^{n+1} y[-1] + a^{n} x[0] + a^{n-1} x[1] + \cdots + a\,x[n-1] + x[n]$$

Example $\quad y[n] = 2x[n] + x[n-1] + 0.5x[n-2]$

known

$$y[0] = 2x[0] + x[-1] + 0.5x[-2]$$
$$y[1] = 2x[1] + x[0] + 0.5x[-1]$$
$$y[2] = 2x[2] + x[1] + 0.5x[0]$$
$$\vdots$$

no recursion

no initial condition needed to solve for y

$$y[n] = \underbrace{-\sum_{k=1}^{N} a_k\, y[n-k]}_{\text{recursive}} + \sum_{k=0}^{M} b_k\, x[n-k]$$

requires initial conditions

$y[-1], y[-2], \cdots y[-N]$

State of the system at $n = -1, -2 \cdots -N$

Nth order difference equation requires N initial conditions $y[-1], y[-2] \ldots y[-N]$

$$y[n] = \underbrace{-\sum_{k=1}^{N} a_k \, y[n-k]}_{\text{recursive}} + \sum_{k=0}^{M} b_k \, x[n-k]$$

If all the initial conditions are equal to zero

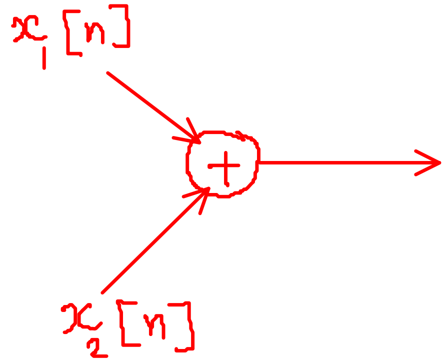
$$y[-1] = y[-2] = \cdots = y[-N] = 0$$

then we say that the system is initially relaxed.

State of an initially relaxed system is equal to zero

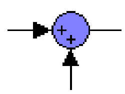
Block Diagrams

Adder



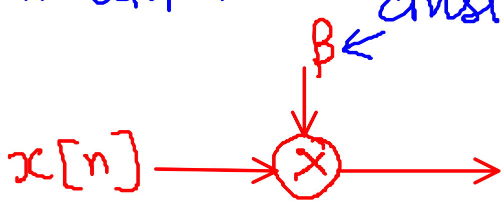
$y[n] = x_1[n] + x_2[n]$

simulink



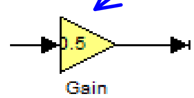
constant multiplier

constant



$y[n] = \beta x[n]$

simulink



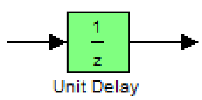
unit delay element

$$x[n] \longrightarrow \boxed{\text{Unit Delay}} \longrightarrow y[n] = x[n-1]$$

z^{-1}

$$x[n] \longrightarrow \boxed{\tfrac{1}{z}} \longrightarrow y[n] = x[n-1]$$

simulink



Memory can be physically realized by FlipFlops (registers)