Discrete - Time Signals and Systems

FIR Filters-I

Yogananda Isukapalli

Discrete-Time System

Something that can transform an input sequence into an output sequence of numbers

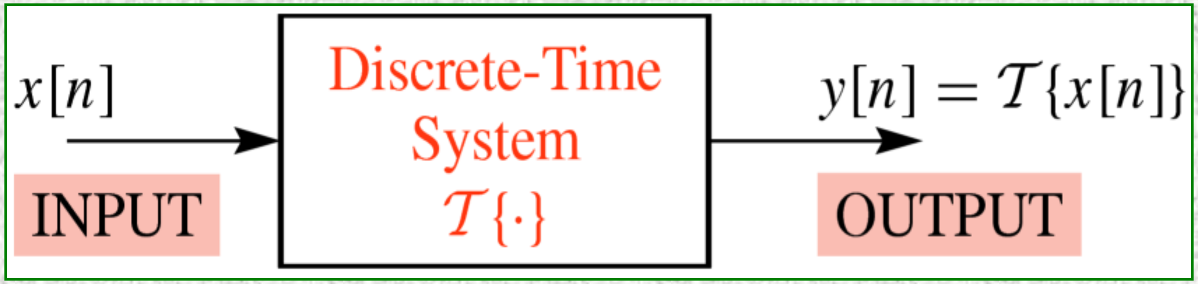


Fig.8.1

Since a discrete-time signal is a sequence of numbers, the operator τ can be described by a mathematical formula. It is just a computational process

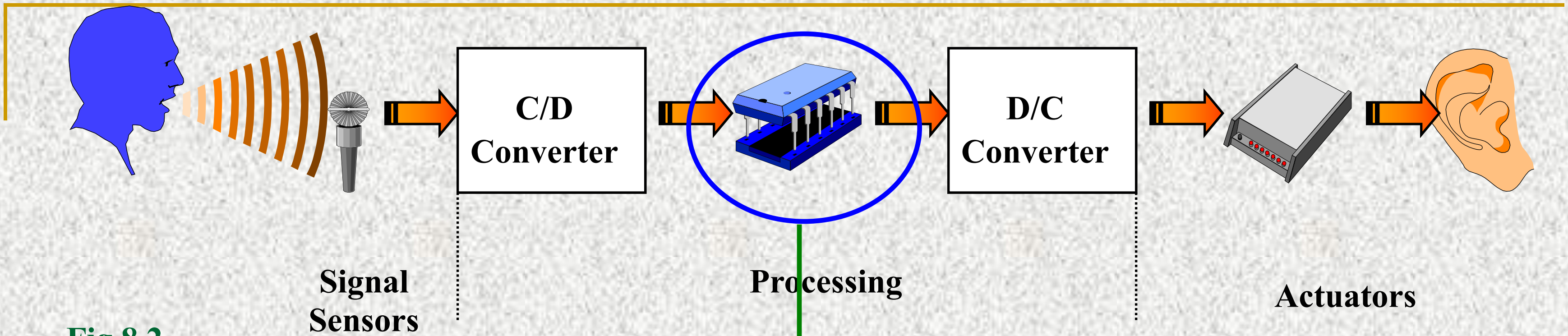


Fig.8.2

Discrete-Time System

An Evolution of computing machines

For the price of a small house, you could have one of these

Model 4136 PROGRAMMABLE DIGITAL FILTER

ROCKLAND PROGRAMMABLE DIGITAL FILTER · MODEL 4136

Variable-Order Digital Filter for Realizing All Classical Designs

The Rockland Model 4136 Programmable Digital Filter consists of a second-order digital filter section which is multiplexed four ways to achieve eighth-order filtering. Each of the four sections has fully-programmable coefficients which are stored internally in a read-write memory.

Filter input and output words are in 16-bit parallel form at a maximum sampling rate of 80 KHz while internal computations are made with 24-bit ac-

TRANSFER FUNCTION

The transfer function from filter input to filter output in z-transform notation is given by

$$H_N(z) = \prod_{n=1}^{N} \frac{K_n(1+z^{-1}A1_n+z^{-2}A2_n)}{1-z^{-1}B1_n-z^{-2}B2_n} \quad (1)$$

where N=0,1,2,3,4 is one-half the filter order se-

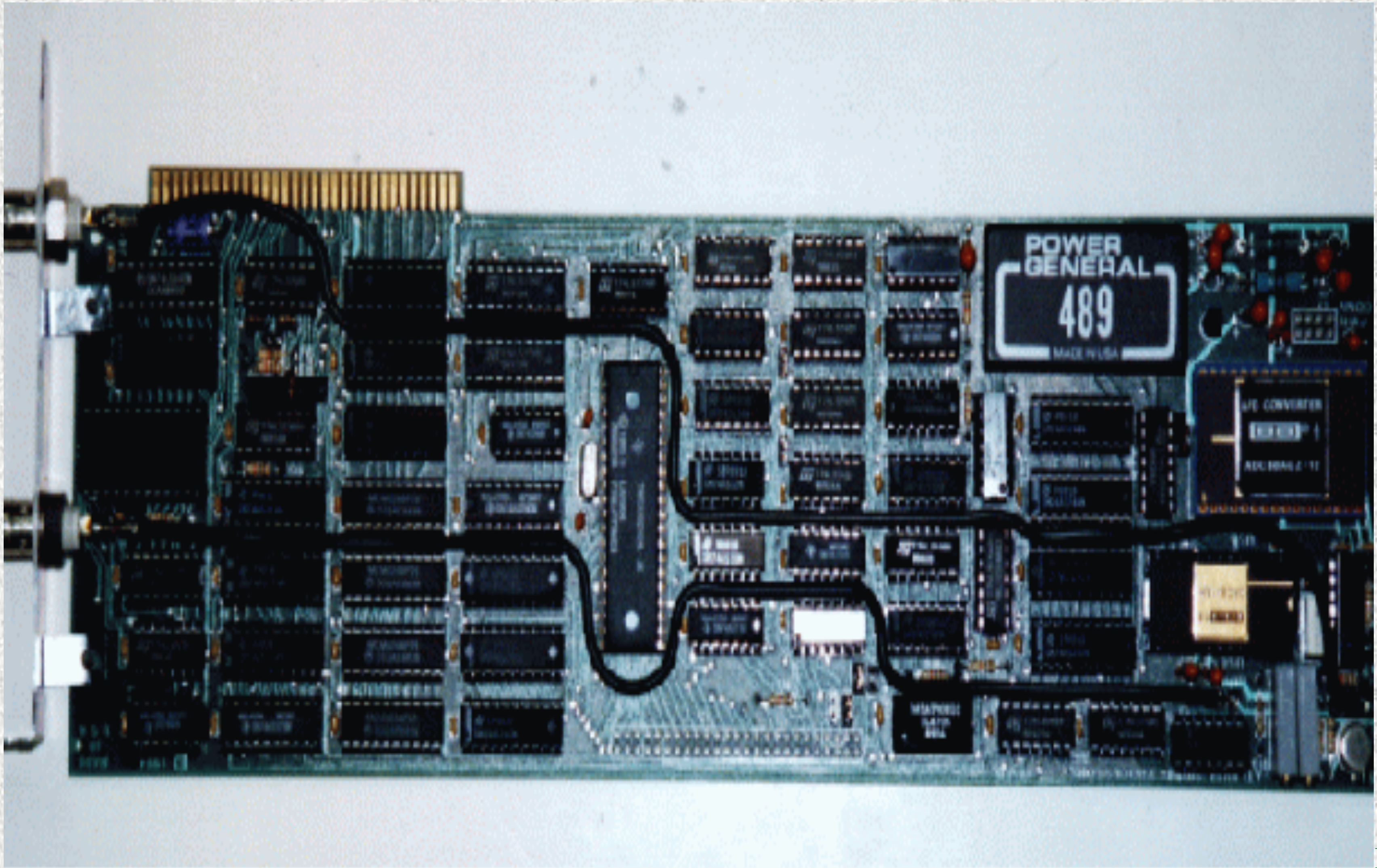
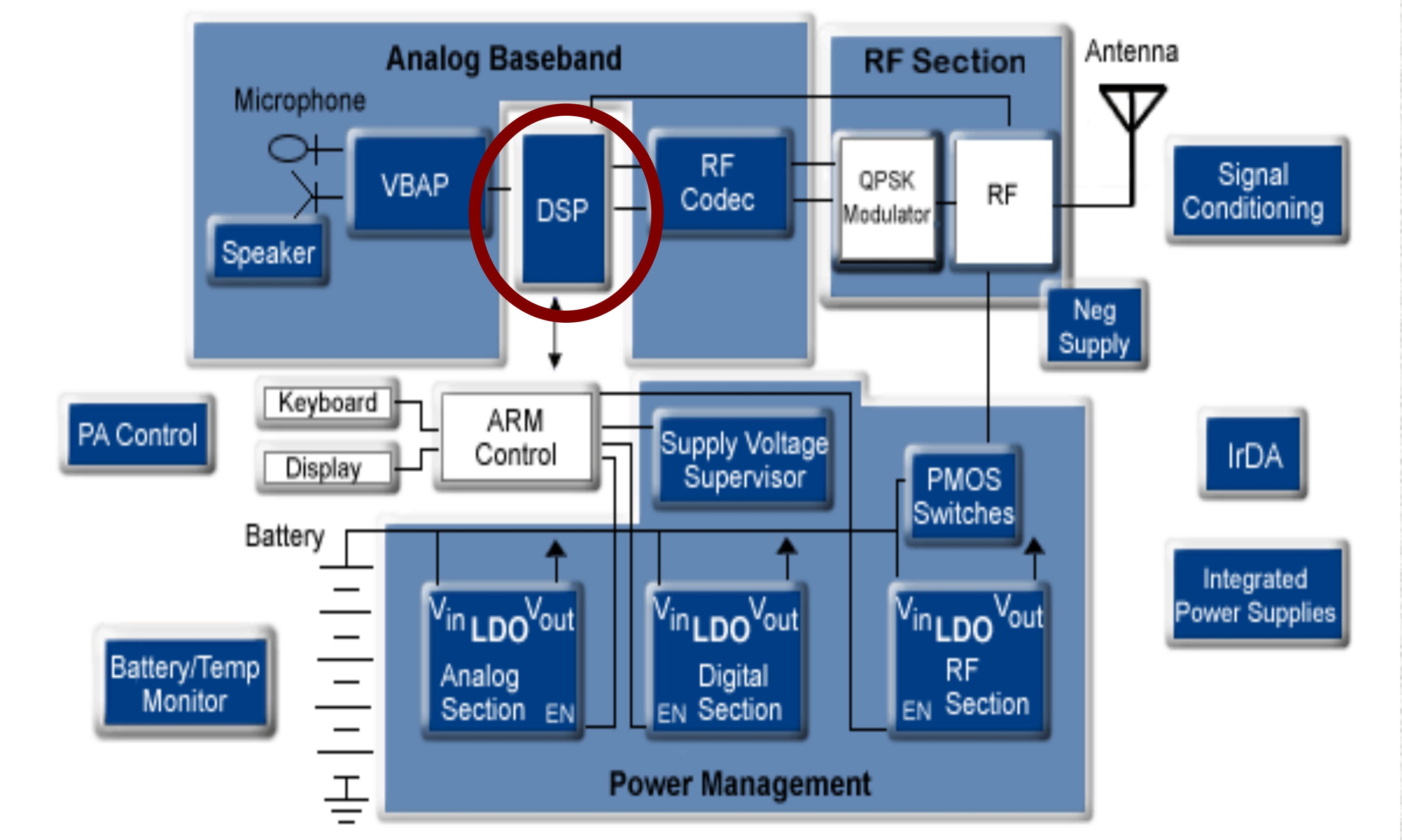


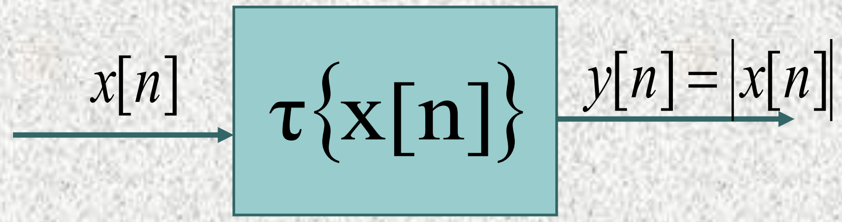
Fig.8.4

TMS32010, 1983: First PC plug-in board from Atlanta Signal Proc

Digital Cell Phone (ca. 2000)



Example1: A Simple Discrete System



Output is the absolute value of input

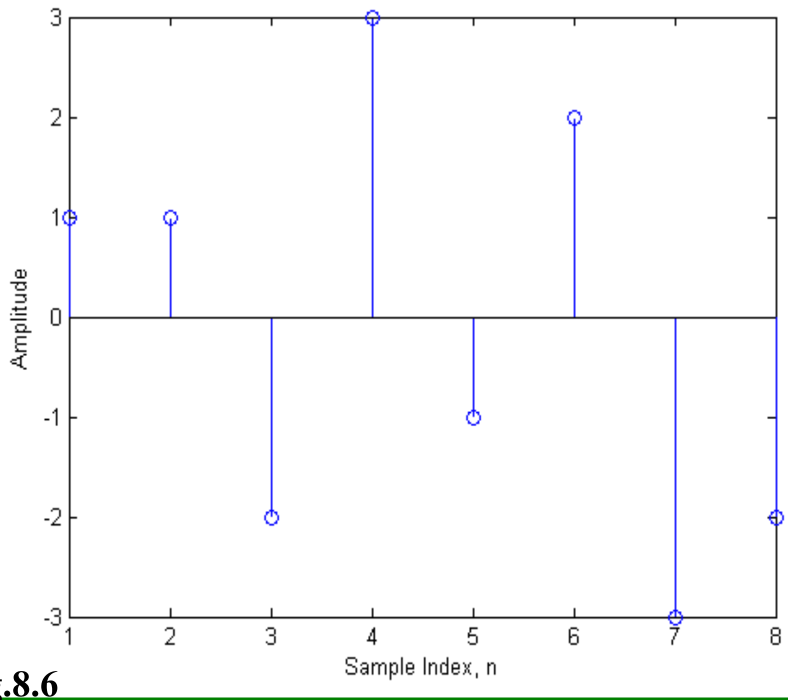


Fig.8.6

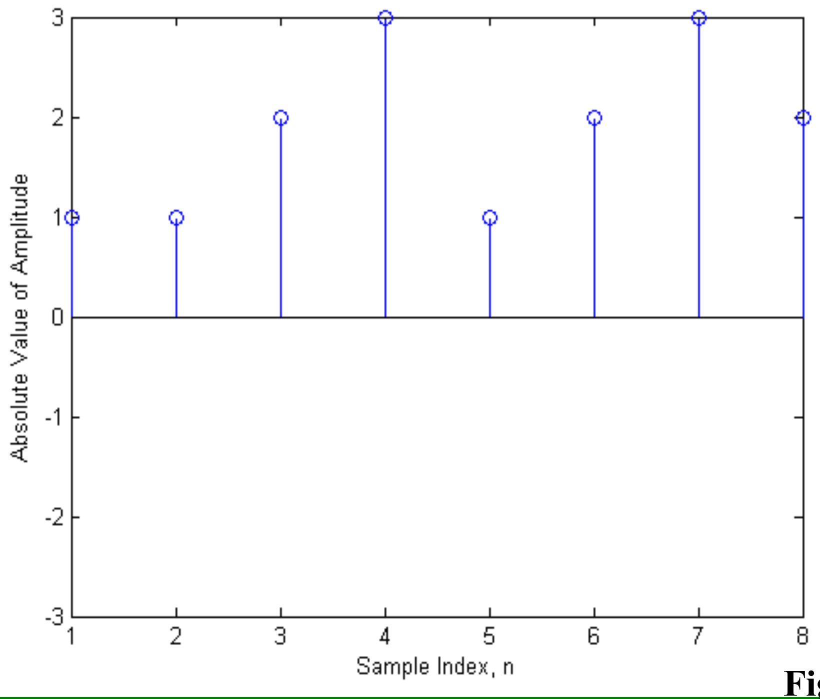


Fig.8.7

Example2: 3-point averaging method

Consider an input sequence x[n]

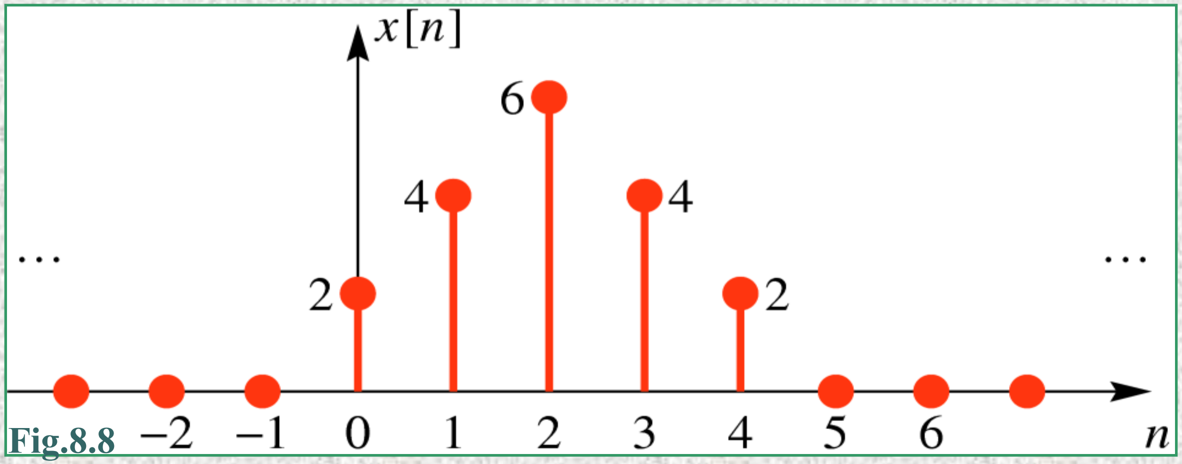


Fig.8.8

$$x[0]=2 \quad x[3]=4$$
$$x[1]=4 \quad x[4]=2$$
$$x[2]=6 \quad x[5]=0$$

Take the average of any three points in a sequence

$$\frac{\{2+4+6\}}{3}=4, \qquad \frac{\{4+6+4\}}{3}=14/3$$
$$\frac{\{6+4+2\}}{3}=4, \qquad \frac{\{4+2+0\}}{3}=2$$

Notation for the output is arbitrary, following the notation shown below leads to the output;

$$y[0] = \frac{\{x[0] + x[1] + x[2]\}}{3} \qquad y[1] = \frac{\{x[1] + x[2] + x[3]\}}{3}$$

n	$n < -2$	-2	-1	0	1	2	3	4	5	$n > 5$
$x[n]$	0	0	0	2	4	**6**	**4**	**2**	0	0
$y[n]$	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	**4**	2	$\frac{2}{3}$	0	0

n=0 $y[0] = \frac{1}{3}(x[0] + x[1] + x[2])$

n=1 $y[1] = \frac{1}{3}(x[1] + x[2] + x[3])$

Fig.8.9

Output

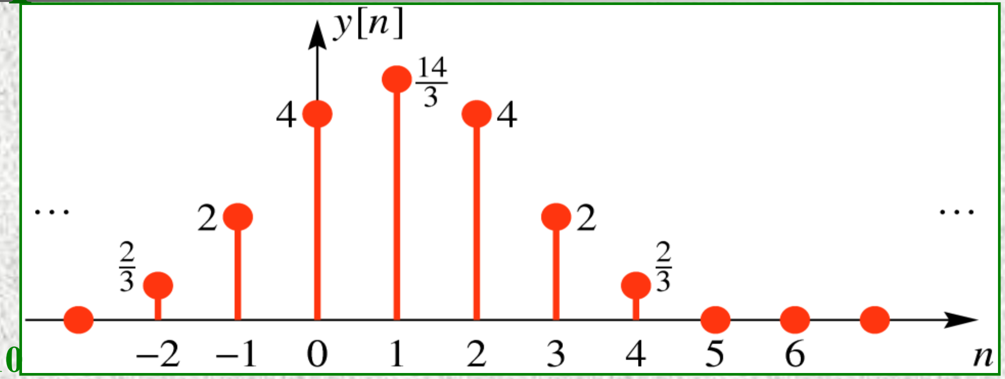


Fig.8.10

The above process (3-point average) generalizes to an important input-output equation known as ***difference equation***

$$y[n] = \frac{1}{3}\left(x[n] + x[n+1] + x[n+2]\right)$$

The above equation describes a very important class of discrete-time systems called *'FIR filters'*

However, the equation $y[n]=\frac{1}{3}\left(x[n]+x[n+1]+x[n+2]\right)$ **doesn't seem practical as we need two future samples to calculate present output**

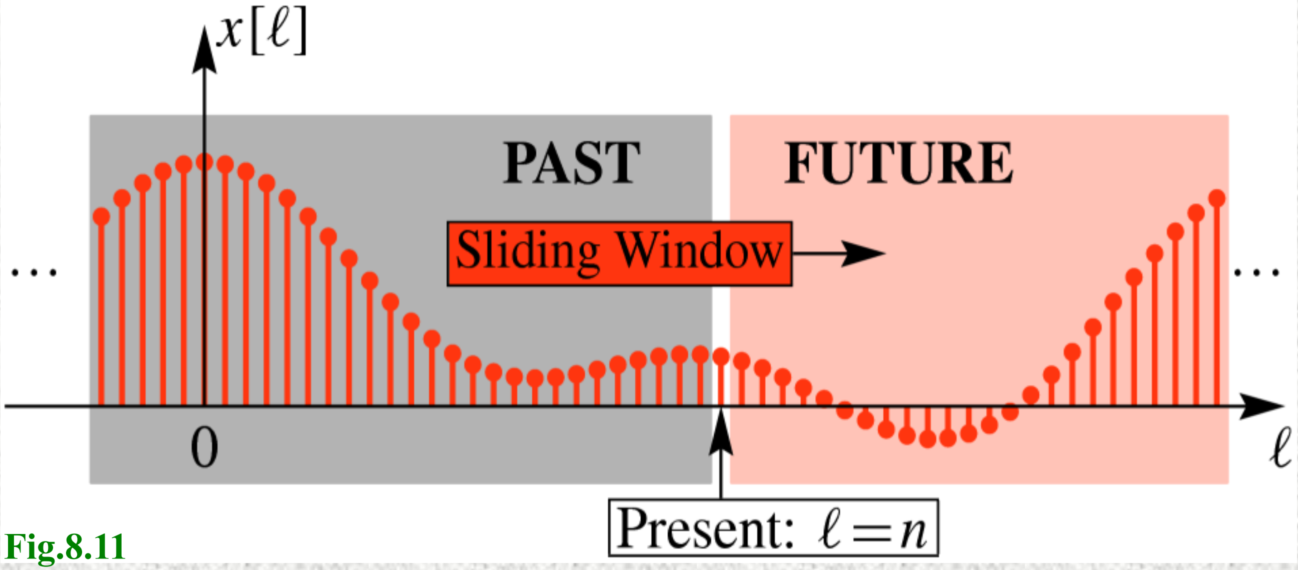


Fig.8.11

$$y[n]=\frac{1}{3}\left(x[n]+x[n-1]+x[n-2]\right)$$

Depends only on past values, a causal system

Difference equations

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k], \quad \textit{Recursive equation}$$

Represents a class of filters known as IIRfilters

$$y[n] = \sum_{k=0}^{M} b_k x[n-k], \quad \textit{Non - Recursive part}$$

FIR filters

$$First\ order\ discrete-time\ system$$

$$y[n] = a_1 y[n-1] + b_0 x[n] \quad a\ Recursive\ equation$$

$$Solution:$$

$$n = 1 \qquad y[1] = a_1 y[0] + b_0 x[1]$$

$$n = 2 \qquad y[2] = a_1 y[1] + b_0 x[2]$$

$$= a_1 \left(a_1 y[0] + b_0 x[1] \right) + b_0 x[2]$$

$$= a_1^2 y[0] + a_1 b_0 x[1] + b_0 x[2]$$

$$n = 3 \qquad y[3] = a_1 y[2] + b_0 x[3]$$

$$= a_1\left(a_1^2 y[0] + a_1 b_0 x[1] + b_0 x[2]\right) + b_0 x[3]$$

$$y[3] = a_1^3 y[0] + a_1^2 b_0 x[1] + a_1 b_0 x[2] + b_0 x[3]$$

Generalizing the 1st order discrete-time system,

$$y[r] = a_1^r y[0] + b_0\left[a_1^{r-1} x[1] + a_1^{r-2} x[2] + \ldots\ldots + a_1^0 x[r]\right]$$

$$y[r] = a_1^r y[0] + b_0 \sum_{m=1}^{r} a_1^{(r-m)} x[m]$$

0

For a causal system, $y[0] = a_1 y[-1] + b_0 x[0] = b_0 x[0]$

$$\therefore\ y_c[r] = a_1^r b_0 x[0] + b_0 \sum_{m=1}^{r} a_1^{(r-m)} x[m]$$

$$y_c[r] = b_0 \sum_{m=0}^{r} a_1^{(r-m)} x[m]$$

Example

$$Let\ x[n]=1 \quad for\ all\ n, \quad n\geq 0$$

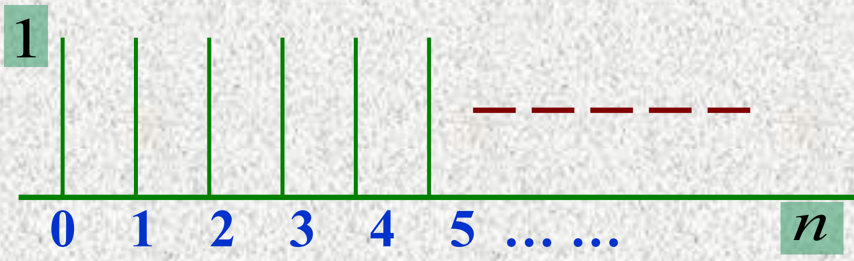


Fig.8.12

$$y[r]=a_1^r y[0]+b_0\left[a_1^{r-1}x[1]+a_1^{r-2}x[2]+.......+a_1^0 x[r]\right]$$

$$=a_1^r y[0]+b_0\left[a_1^{r-1}+a_1^{r-2}+.......+1\right]$$

$$y[r]=a_1^r y[0]+b_0\frac{a_1^r-1}{a_1-1} \qquad a_1\neq 1$$

$$y[r] = y[0] + b_0 r \qquad a_1 = 1$$

$$\because \lim_{a_1 \to 1} \frac{a_1^r - 1}{a_1 - 1} = \lim_{a_1 \to 1} \frac{\frac{d}{da_1}\left(a_1^r - 1\right)}{\frac{d}{da_1}\left(a_1 - 1\right)}$$

$$= \lim_{a_1 \to 1} \frac{r a_1^{r-1}}{1} = r 1^{r-1} = r$$

$$Let \quad a_1 = 1/2, \quad b_0 = 2, \quad y[0] = 3$$

$$y[n] = \left(1/2\right)^n 3 + 2 \frac{\left(1/2\right)^n - 1}{\left(1/2\right) - 1}$$

$$= \left(1/2\right)^n 3 - 4\left[\left(1/2\right)^n - 1\right] = 4 - \left(1/2\right)^n \quad n \geq 0$$

The General FIR Filter

Non-Recursive part of difference equation represents a general FIR filter

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

The above equation doesn't involve any past samples, so the system is a *causal* one. The moving average problem discussed earlier is an FIR filter

Filter Order = M: ***No. of memory blocks required in the filter implementation***

Filter Length, L = M+1: ***Total No. of samples required in calculating the output, M from memory (past) and one present sample***

Filter coefficients $\{b_k\}$: ***Completely define an FIR filter. All the properties of the filter can be understood through the coefficients***

Graphical view of a general FIR filter

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

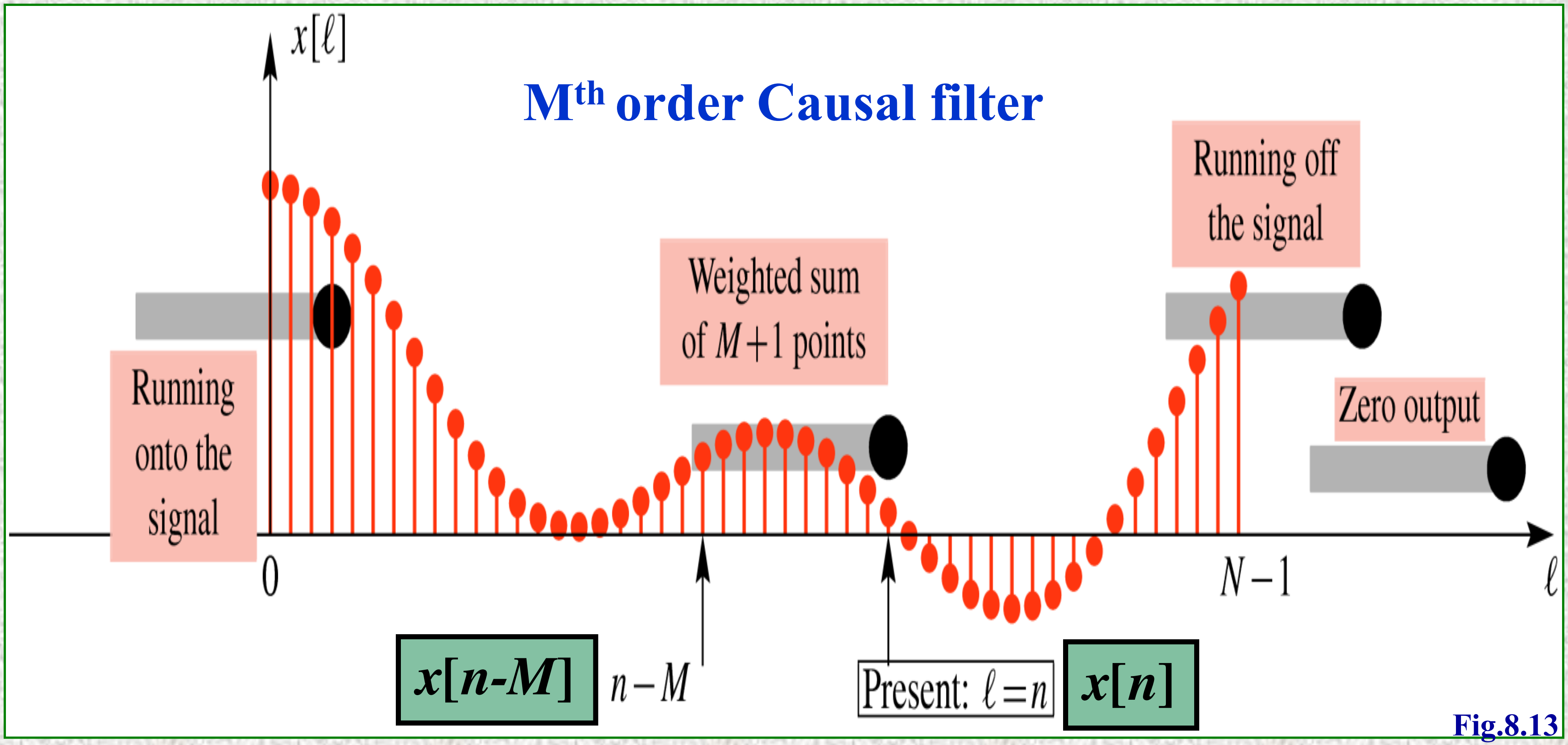


Fig.8.13

Block Diagrams: An Implementation view of FIR filters

Building blocks required

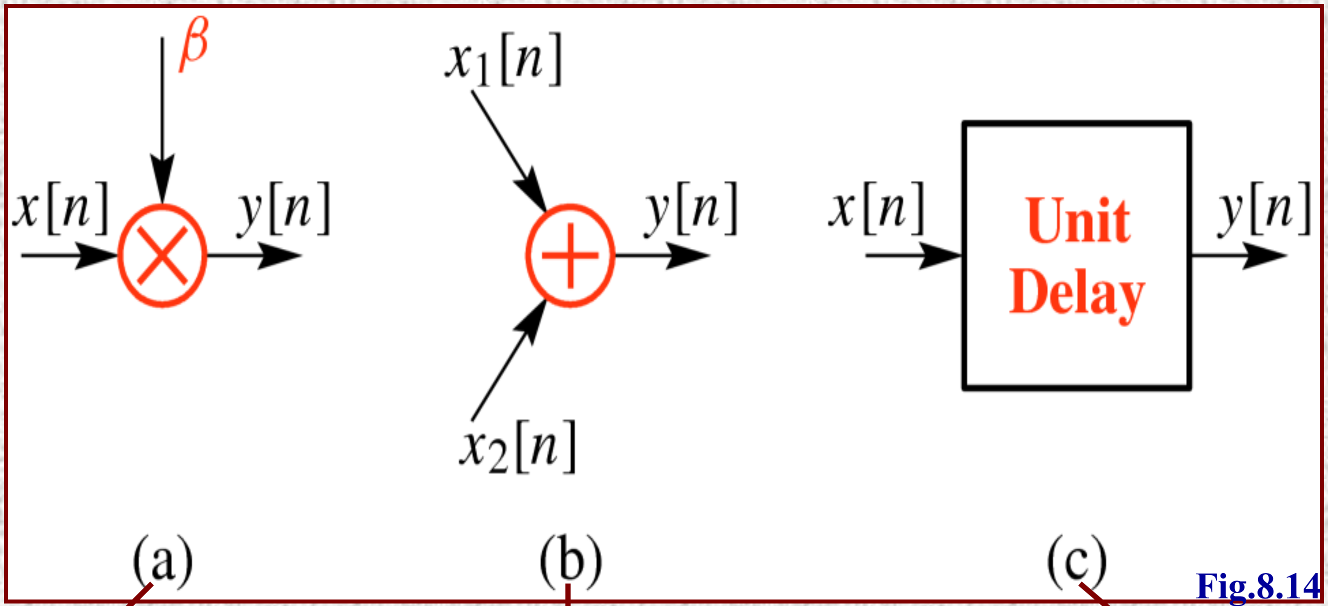


Fig.8.14

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

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

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

direct form **of block diagram**

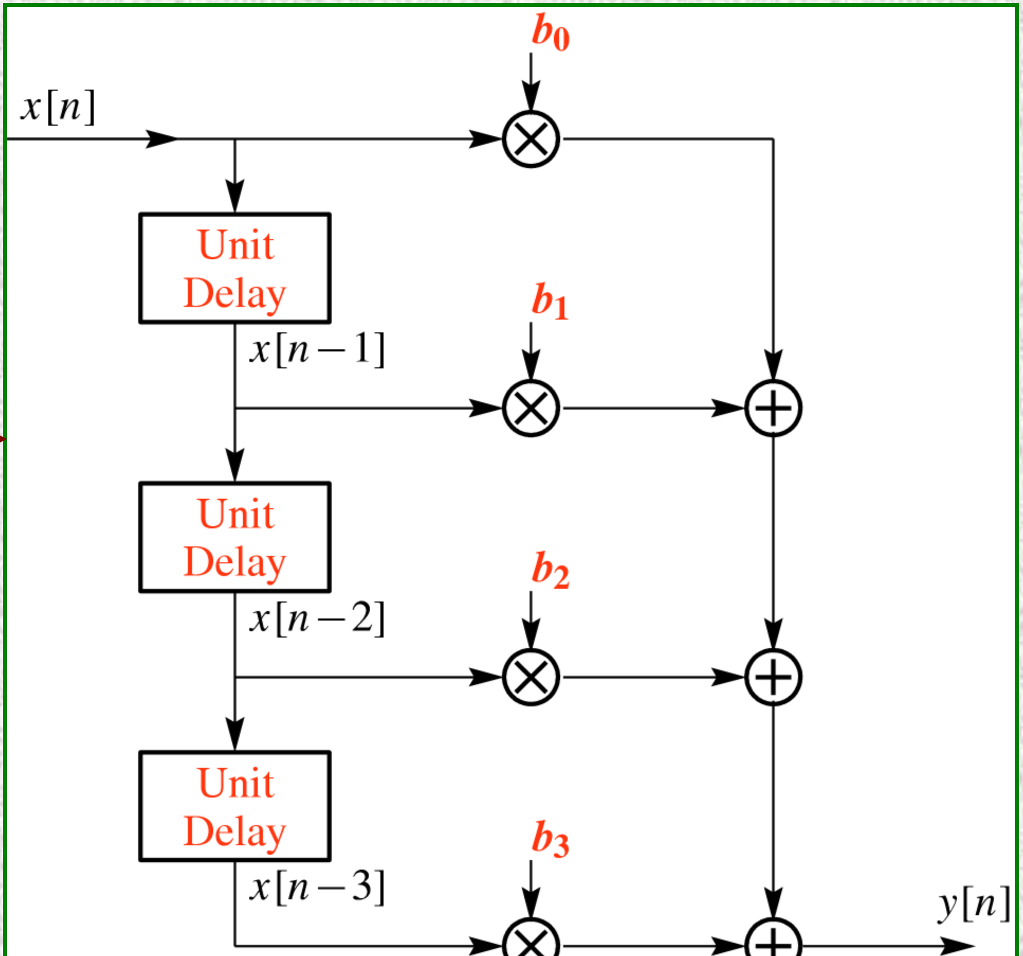


Fig.8.15

Difference equation

$$y[n]=\Big(\big(\big(b_0x[n]+b_1x[n-1]\big)+b_2x[n-2]\big)+b_3x[n-3]\Big)$$

***transposed form* of block diagram**

$$y[n]=b_0x[n]+v_1[n-1]$$
$$v_1[n]=b_1x[n]+v_2[n-1]$$
$$v_2[n]=b_2x[n]+v_3[n-1]$$
$$v_3[n]=b_3x[n]$$

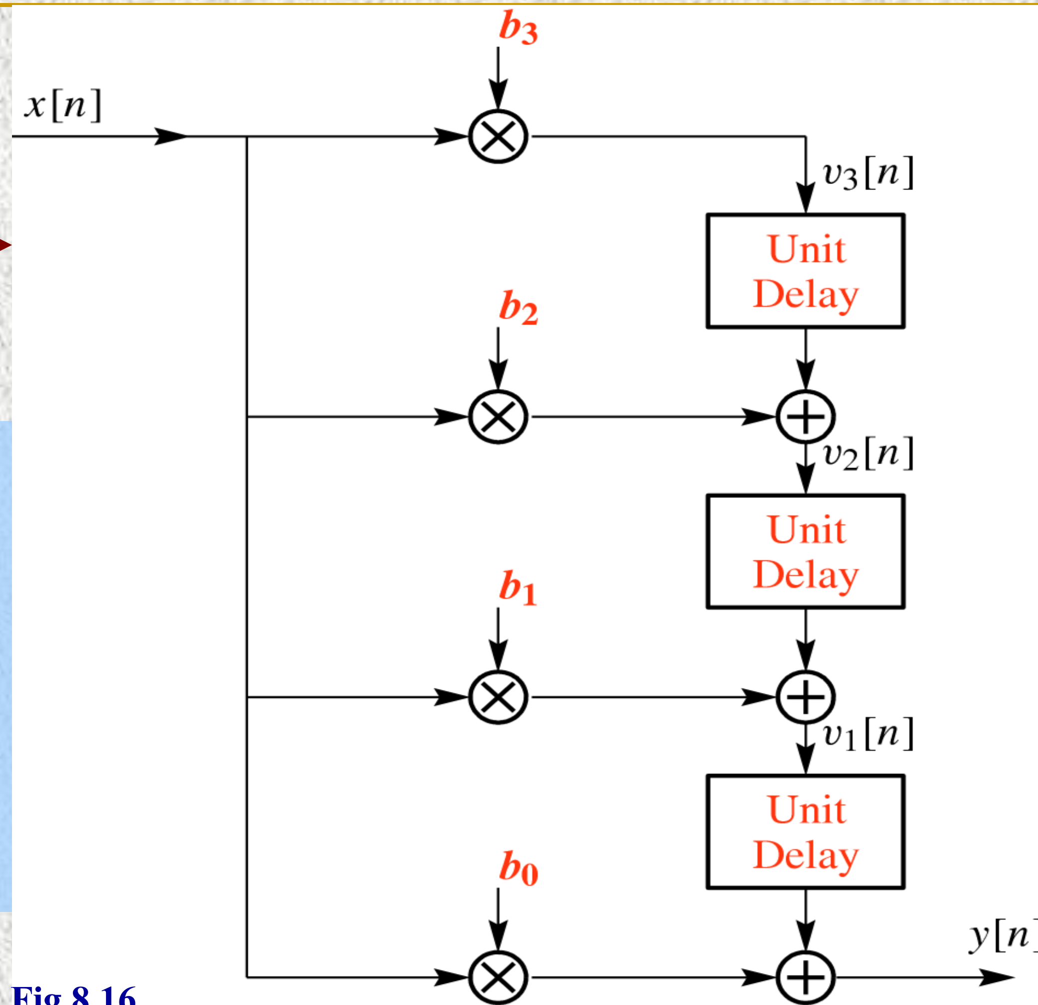


Fig.8.16

$$y[n]=\left(\left(\left(b_0x[n]+b_1x[n-1]\right)+b_2x[n-2]\right)+b_3x[n-3]\right)$$

Same Result!!!

Example: FIR filter application

Averaging of a sequence with different filter lengths

$$x[n] = (1.02)^n + \cos(2\pi n/8 + \pi/4) \quad \text{for } 0 \le n \le 40$$

Signal — **Noise,** ***cosine part***

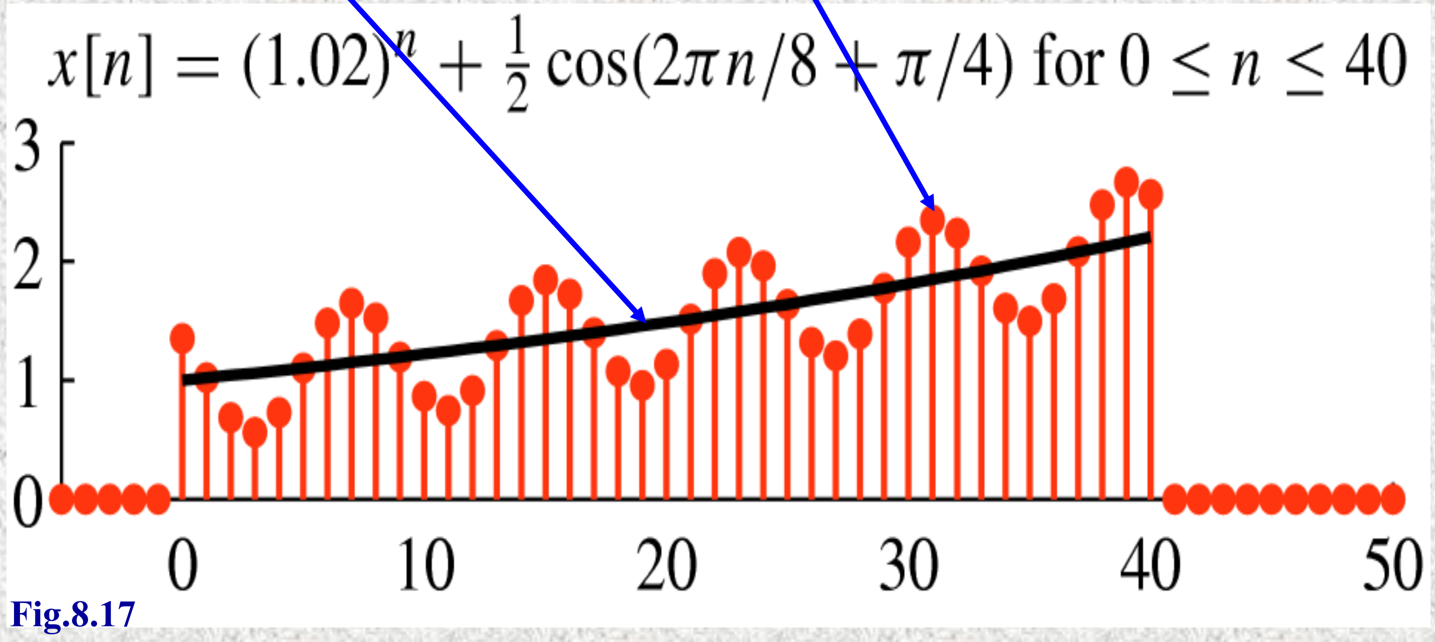


Fig.8.17

$$y_3[n] = \sum_{k=0}^{2} \left(\tfrac{1}{3}\right) x[n-k]$$

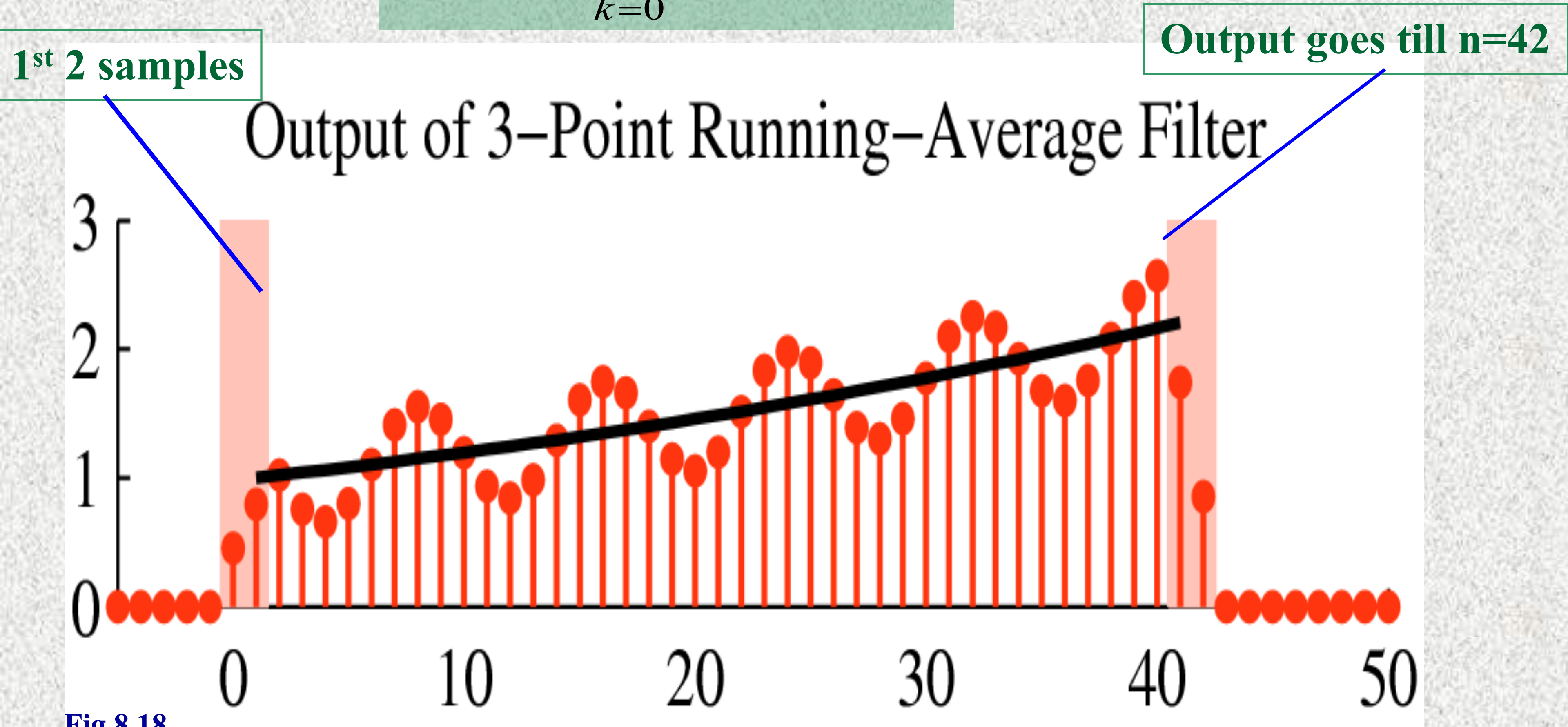


Fig.8.18

Notice that the output is smoother or its noise level is low than input

$$y_7[n] = \sum_{k=0}^{6} \left(\tfrac{1}{7}\right) x[n-k]$$

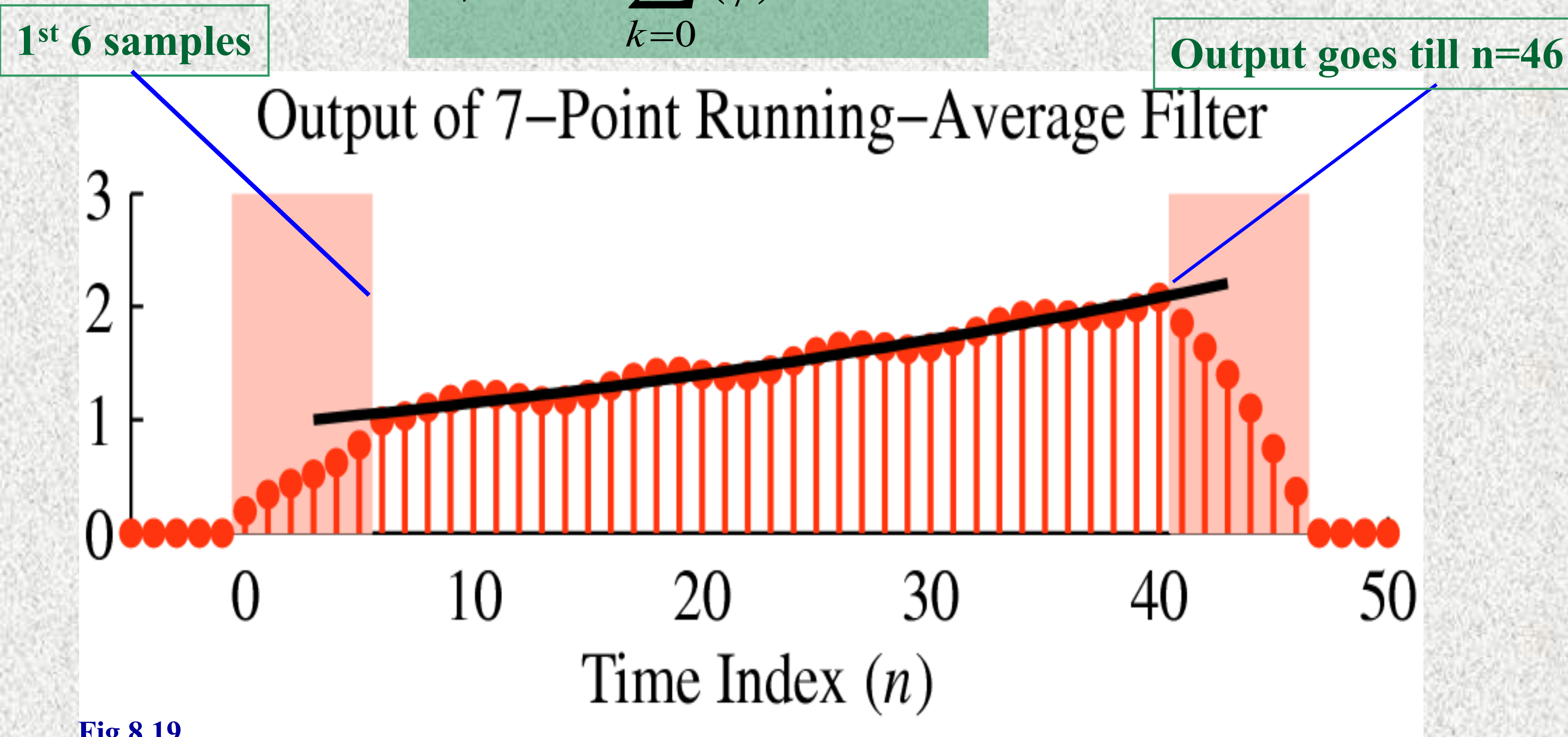


Fig.8.19

Notice that the output is much smoother than 3-point averaging method, noise level is low

Example: DJIA signal

similar approach, averaging

$$y[n] = \left(\tfrac{1}{51}\right)\sum_{k=0}^{50} x[n-k]$$

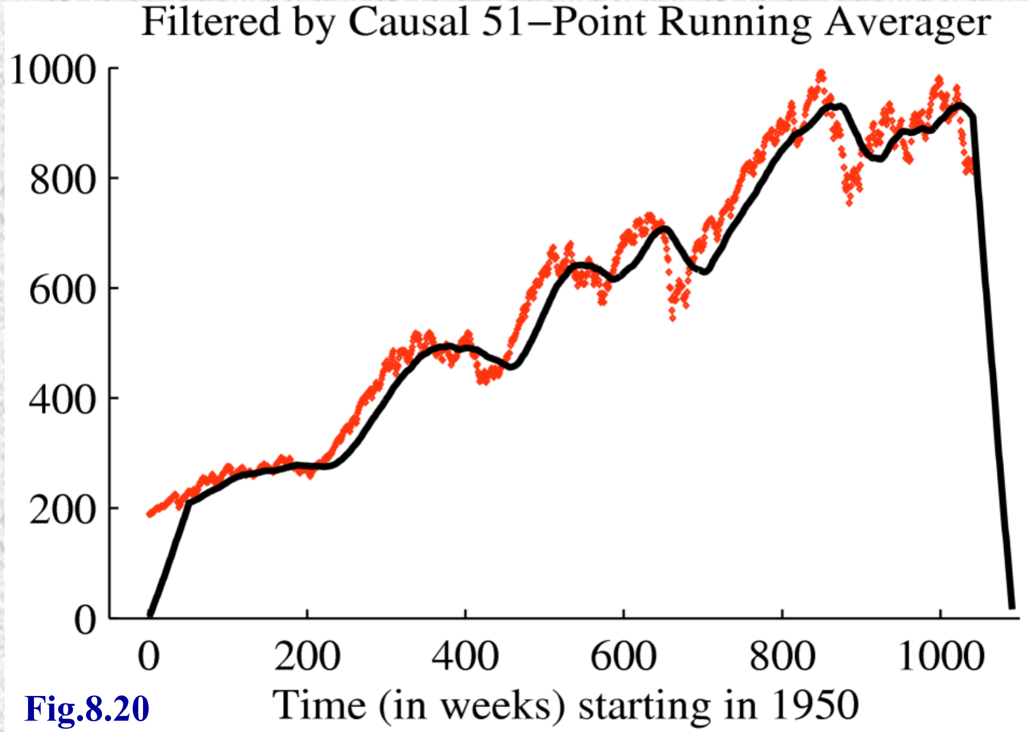
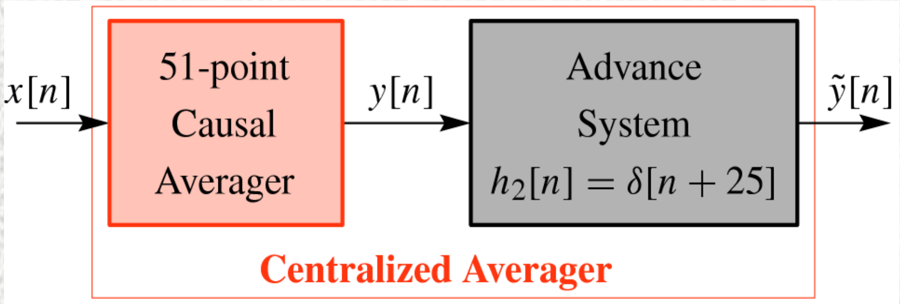


Fig.8.20

Compensating for delay,



$$\tilde{y}[n] = \left(\frac{1}{51}\right) \sum_{k=-25}^{25} x[n-k]$$

Fig.8.21

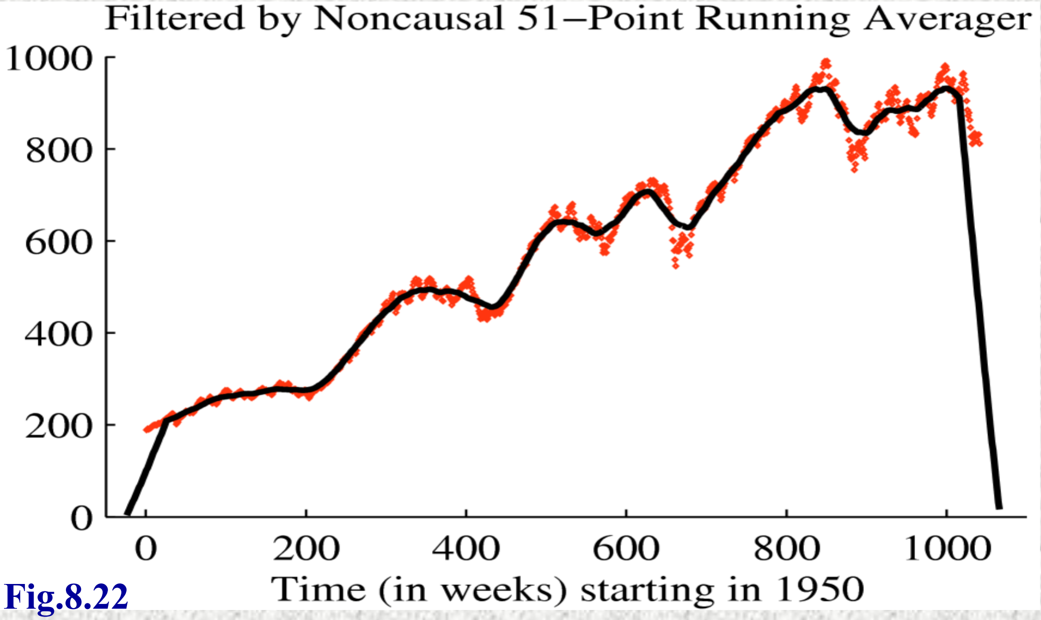


Fig.8.22

In these two examples, FIR filters are shown to remove rapid fluctuations

Discrete-Time Unit Impulse Sequence

Unit Impulse is the simplest sequence with only one nonzero value at n=0

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

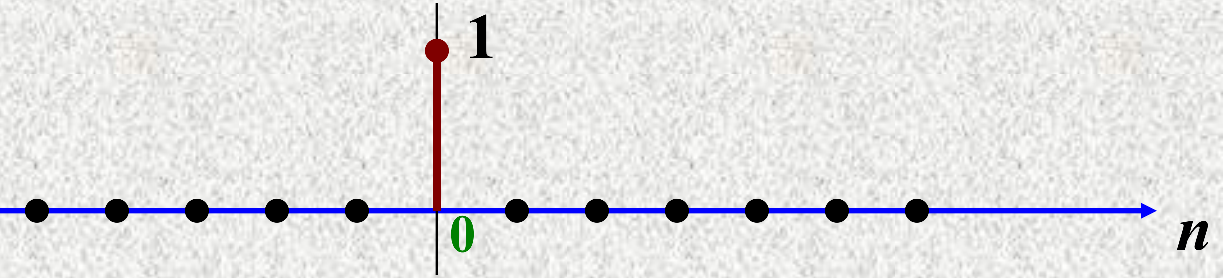


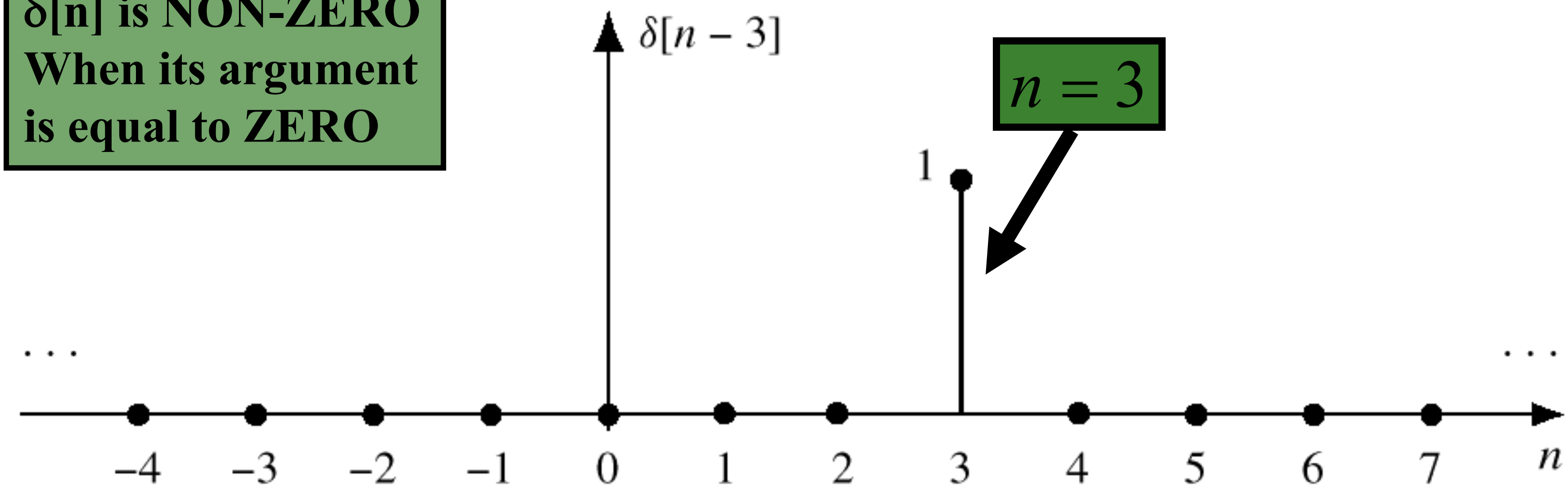
Fig.8.23

$\delta[n]$ **Is known as Kronecker delta function**

Tabular form and a shifted version of unit impulse

n	...	−2	−1	0	1	2	3	4	5	6	...
$\delta[n]$	0	0	0	1	0	0	0	0	0	0	0
$\delta[n-3]$	0	0	0	0	0	0	1	0	0	0	0

δ[n] is NON-ZERO When its argument is equal to ZERO



Shifted impulse sequence, $\delta[n-3]$.

Fig.8.24

Unit Impulse Response Sequence

The response of an FIR filter to a unit impulse sequence is called as unit impulse response or simply ***'impulse response'***

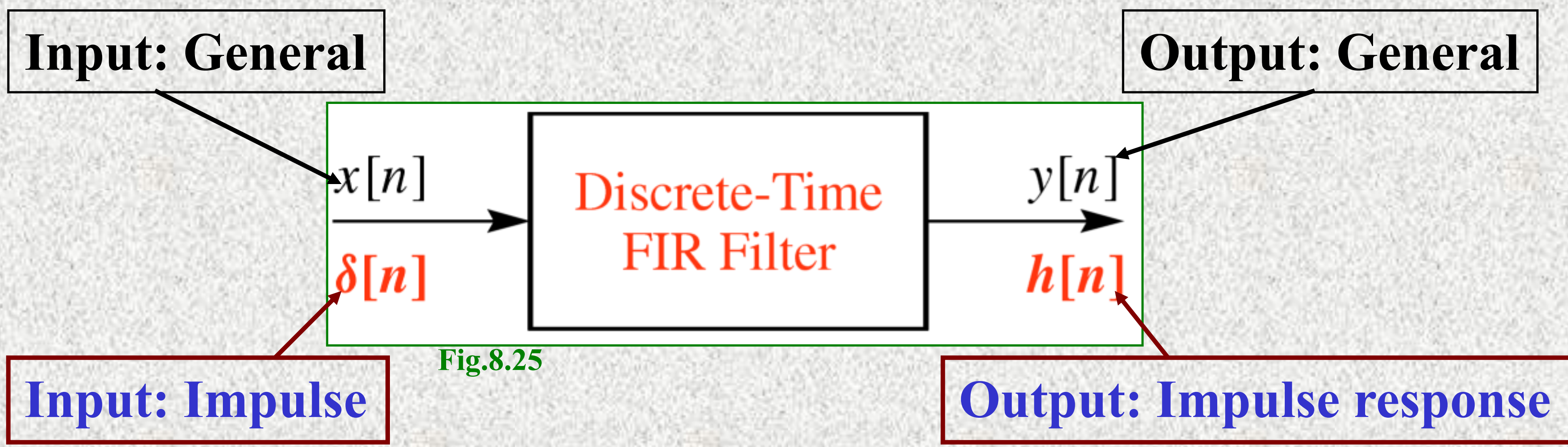


Fig.8.25

General FIR equation

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

Impulse Response

$$x[n] = \delta[n], \qquad y[n] = h[n]$$

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k] = \begin{cases} b_n & n = 0, 1, 2, \ldots M \\ 0 & otherwise \end{cases}$$

The sum evaluates to a single term for each value of n, as $\delta[n-k]$ is nonzero only when *n=k*

Tabular form for '*Impulse Response*' equation

n	$n<0$	0	1	2	3	$\ldots$	M	$M+1$	$n>M+1$
$x[n]=\delta[n]$	0	1	0	0	0	0	0	0	0
$y[n]=h[n]$	0	b_0	b_1	b_2	b_3	$\ldots$	b_M	0	0

Fig.8.26

In the above table $h[n]=0$ *for* $n<0$ *and* $n>M$,

The length of impulse response sequence is finite.

This is why the system is called a

Finite Impulse Response (FIR) system

Example1: 3-point average filter

$$y_3[n] = \left(\tfrac{1}{3}\right)\sum_{k=0}^{2} x[n-k] \text{ compare with standard equation}$$

$$y[n] = \sum_{k=0}^{M} b_k x[n-k], \quad h[0] = b_0,\ h[1] = b_1,\ h[2] = b_2$$

$$\therefore h[0] = \frac{1}{3},\ h[1] = \frac{1}{3},\ h[2] = \frac{1}{3}$$

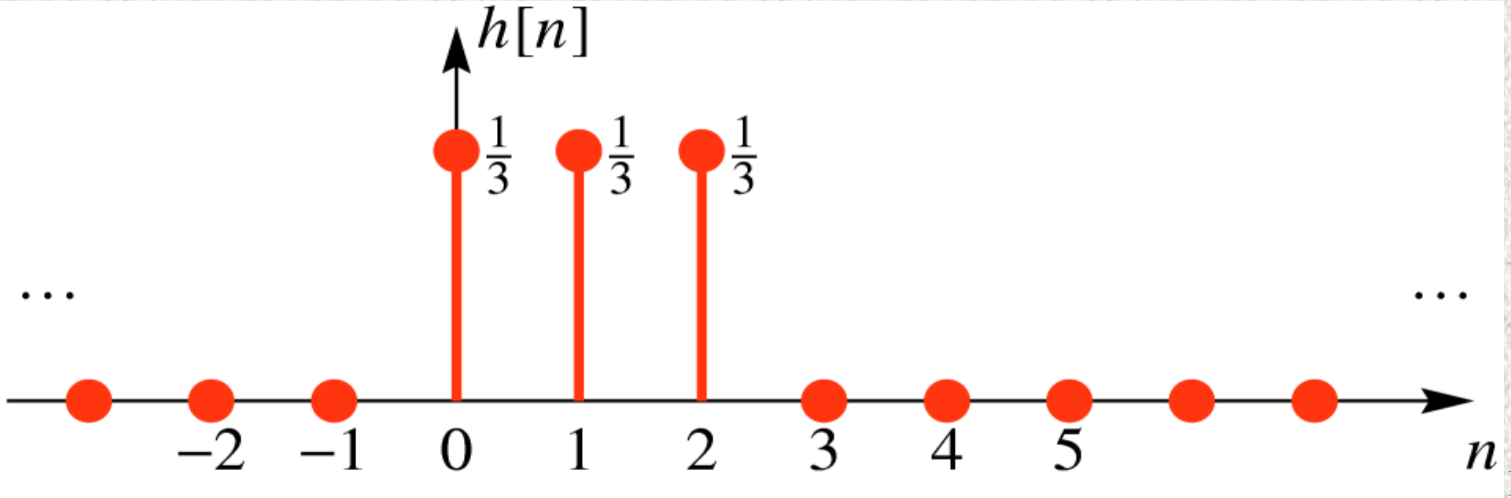


Fig.8.27

Example2

Find the difference equation governing the input − output relation with FIR filter coefficients $\{3,-1,2,1\}$

FIR filter coefficients $h[n]=\{3,-1,2,1\}$

$h[n]=b_k \quad for \quad k=0,1\ldots N$

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

$$=\sum_{k=0}^{3} b_k x[n-k]$$

$$=b_0x[n]+b_1x[n-1]+b_2x[n-2]+b_3x[n-3]$$

$$=3x[n]-x[n-1]+2x[n-2]+x[n-3] \text{ difference equation}$$

Representation of a general sequence x[n]

Any sequence can be obtained by adding shifted impulses

$$x[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$$

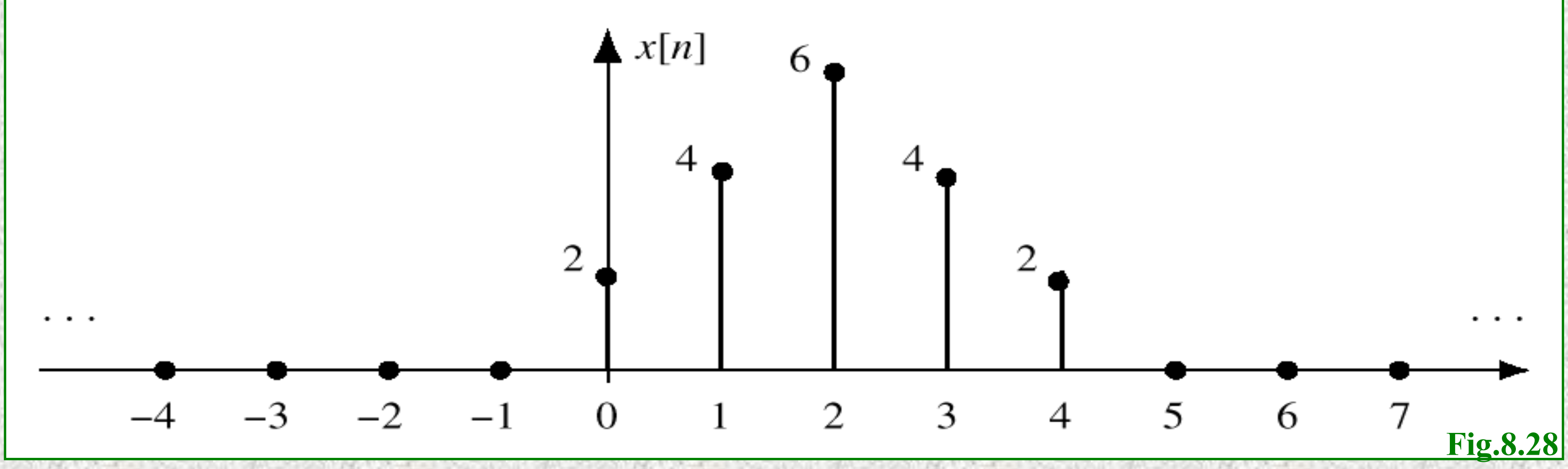


Fig.8.28

Tabular form: Breaking a sequence into shifted impulses

n	$\dots$	-2	-1	0	1	2	3	4	5	6	$\dots$
$2\delta[n]$	0	0	0	2	0	0	0	0	0	0	0
$4\delta[n-1]$	0	0	0	0	4	0	0	0	0	0	0
$6\delta[n-2]$	0	0	0	0	0	6	0	0	0	0	0
$4\delta[n-3]$	0	0	0	0	0	0	4	0	0	0	0
$2\delta[n-4]$	0	0	0	0	0	0	0	2	0	0	0
$x[n]$	0	0	0	2	4	6	4	2	0	0	0

$$x[n] = \sum_k x[k]\delta[n-k]$$

For any signal

$$= \dots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \dots$$

Fig.8.29

Discrete-Time Convolution Sum

General Discrete-System

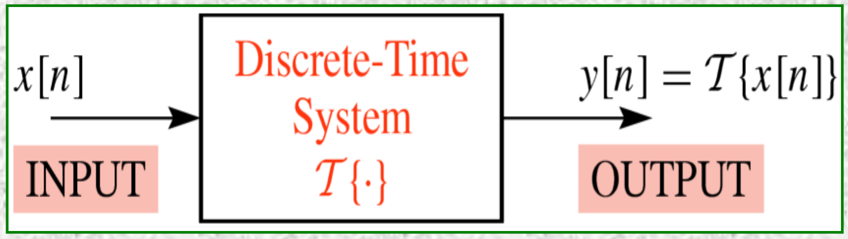
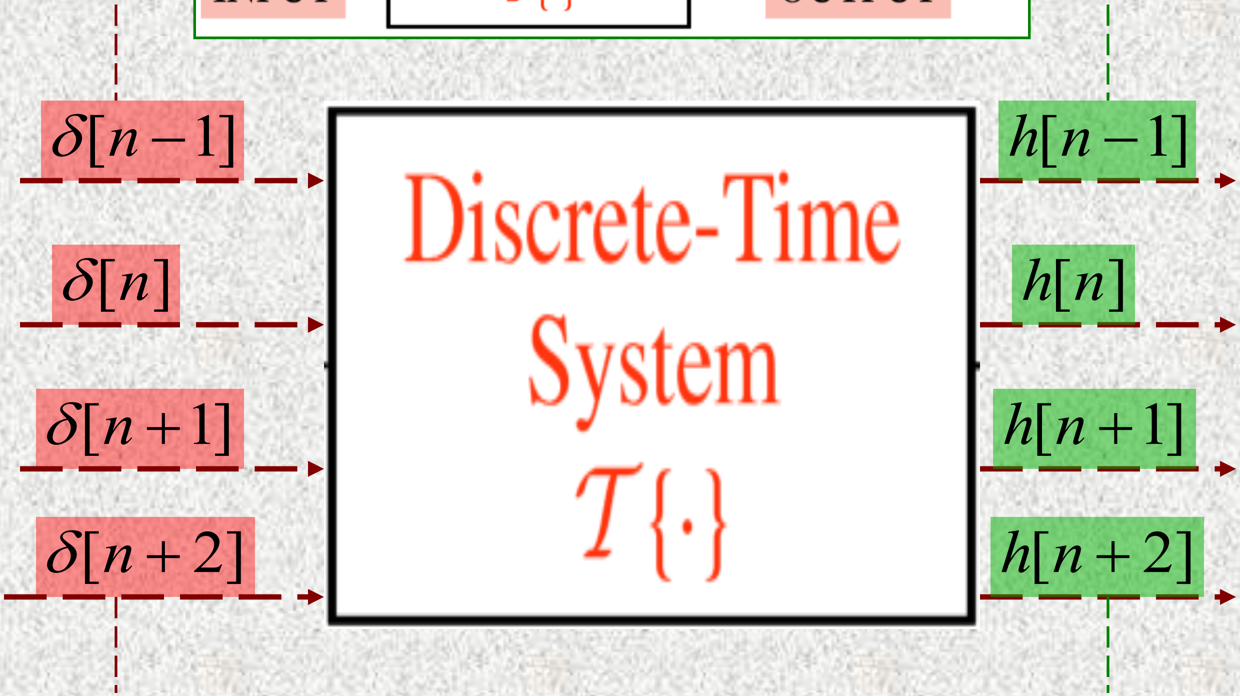


Fig.8.30



From the previous figure,

$$x[0]\delta[n] = x[0]h[n]$$

$$x[0]\delta[n-1] = x[0]h[n-1]$$

.

$$x[0]\delta[n-k] = x[0]h[n-k]$$

As shown previously using superposition,

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

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k], \quad \textit{Convolution Sum}$$

Example1: FIR from Convolution

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k], \quad If \;\; h[n] \; is \; non-zero$$

$only \; in \; the \; interval \quad 0 \le n \le M \;\; then,$

$$y[n] = \sum_{k=n-M}^{n} x[k]h[n-k],$$

which is a classic FIR filter

$About \;\; limits: \;\; 0 \le (n-k) \le M$

$\therefore (n-M) \le k \le n$

Example2: Computing the output

$$x[n] = \{2, 4, 6, 4, 2\}, \quad h[n] = \{3, -1, 2, 1\}$$

$$convolve\ x[n]\ and\ h[n]\ to\ get\ y[n]$$

- **Write out the signals x[n] and y[n] on separate rows**
- **The output is to be computed as sum of shifted rows**
- **Each shifted row is to be produced by multiplying the x[n] row by one of the h[k] values and,**
- **By shifting the result to the right so that it lines up with h[k] position**

Numerical convolution done through the above process is also called as synthetic polynomial multiplication

Tabular form describing the convolution

n	$n<0$	0	1	2	3	4	5	6	7	$n>7$
$x[n]$	0	2	4	6	4	2	0	0	0	0
$h[n]$	0	3	-1	2	1	0	0	0	0	0
$h[0]x[n]$	0	6	12	18	12	6	0	0	0	0
$h[1]x[n-1]$	0	0	-2	-4	-6	-4	-2	0	0	0
$h[2]x[n-2]$	0	0	0	4	8	12	8	4	0	0
$h[3]x[n-3]$	0	0	0	0	2	4	6	4	2	0
$y[n]$	0	6	10	18	16	18	12	8	2	0

$y[2]$

Fig.8.32

Try the demo on your CD

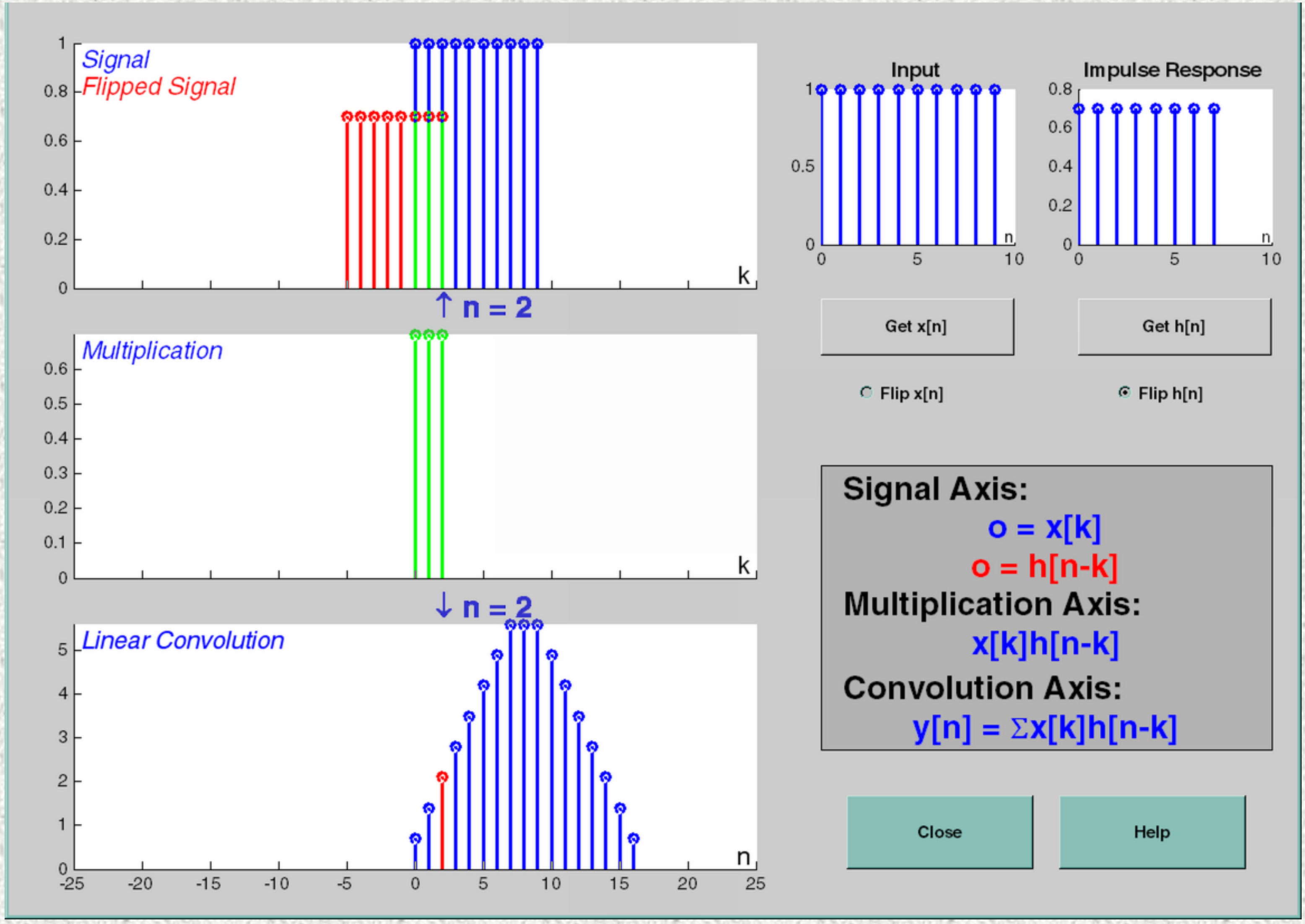


Fig.8.33

Reference

James H. McClellan, Ronald W. Schafer and Mark A. Yoder, “ 5.1-5.4 and 5.6--Signal Processing First”, Prentice Hall, 2003