Sampling

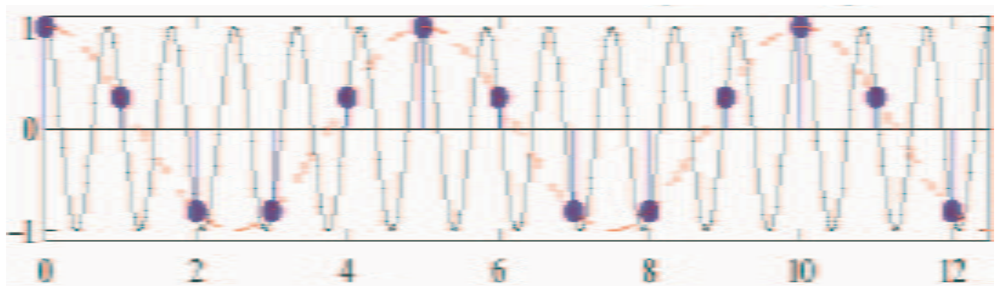
- SAMPLING PROCESS
  - Convert x(t) to numbers x[n]
  - “n” is an integer; x[n] is a sequence of values
  - Think of “n” as the storage address in memory
- UNIFORM SAMPLING at t = nTs
  - IDEAL: x[n] = x(nTs)
- SAMPLING RATE (fs)
  - fs =1/Ts
    - NUMBER of SAMPLES PER SECOND
  - Ts = 125 microsec,
    - fs = 8000 samples/sec (Hz)
- HOW OFTEN ?
  - DEPENDS on FREQUENCY of SINUSOID
  - ANSWERED by SHANNON/NYQUIST Theorem
  - ALSO DEPENDS on “**RECONSTRUCTION**”

Reconstruction

- Given the samples, draw a sinusoid through the values

When "n" is an integer

$\cos(0.4\pi n)$ or
$\cos(2.4\pi n)$



Time axis n

$x[n] = \cos(0.4\pi n)$

- CONVERT STREAM of NUMBERS to x(t)
  - "CONNECT THE DOTS"
  - INTERPOLATION
  - Math model

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]p(t - nT_s)$$

Linear Filtering

- Background: Signals and Systems
  - Let **δ[*k*]** be a discrete-time impulse function, a.k.a. the Kronecker delta function:

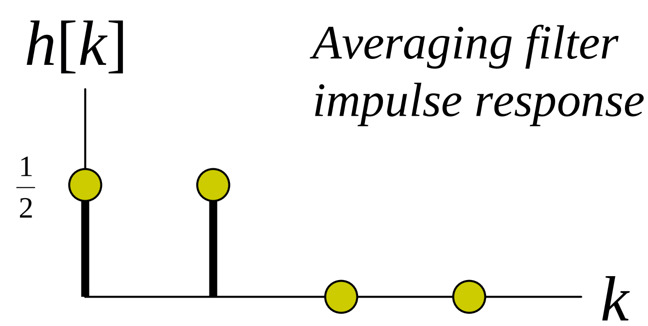
$$\delta[k] = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

  - Impulse response ***h*[*k*]**: response of a discrete-time LTI system to a discrete impulse function
  - We are interested in Finite impulse response filter
    - Non-zero extent of impulse response is finite
    - Can be in continuous time or discrete time
    - Also called a tapped delay line

Discrete-time Convolution

- By linear and time-invariant properties, linear convolution
  - For each value of *k*, compute a different (possibly) infinite summation for *y*[*k*]

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



Averaging filter impulse response

$$y[k] = h[0]\, x[k] + h[1]\, x[k-1]$$

$$= (\, x[k] + x[k-1]\,) \,/\, 2$$

Linear Time-Invariant Systems

- The Fundamental Theorem of Linear Systems
  - Inputs a complex sinusoid into an LTI system, the output
    - a complex sinusoid of the same frequency
    - scaled by the response of the LTI system at that frequency
  - Scaling may attenuate the signal and shift it in phase
  - Example in discrete time. Let $x[k] = e^{j \Omega k}$,

$$y[k] = \sum_{m=-\infty}^{\infty} e^{j \Omega (k-m)} h[m] = e^{j \Omega k} \underbrace{\sum_{m=-\infty}^{\infty} h[m] e^{-j \Omega m}}_{H(\Omega)} = e^{j \Omega k} H(\Omega)$$

  - $H(\Omega)$ is the discrete-time Fourier transform of $h[k]$ and is also called the frequency response

Frequency Response

- For discrete-time systems, response to complex sinusoid is

frequency response → $H\left(e^{j\omega}\right)$

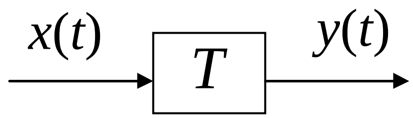
$$e^{j\omega k} \rightarrow H\left(e^{j\omega}\right) e^{j\omega k}$$

$$\cos(\omega k) \rightarrow \left| H\left(e^{j\omega}\right) \right| \cos\left(\omega k + \angle H\left(e^{j\omega}\right)\right)$$

Example: Ideal Delay

- Continuous Time

Delay by *T* seconds



$$y(t) = x(t - T)$$

Impulse response

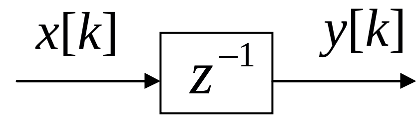
$$h(t) = \delta(t - T)$$

Frequency response

$$H(\Omega) = e^{-j\Omega T}$$
$$|H(\Omega)| = 1$$
$$\angle H(\Omega) = -\Omega T$$

- Discrete Time

Delay by 1 sample



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

Impulse response

$$h[k] = \delta[k - 1]$$

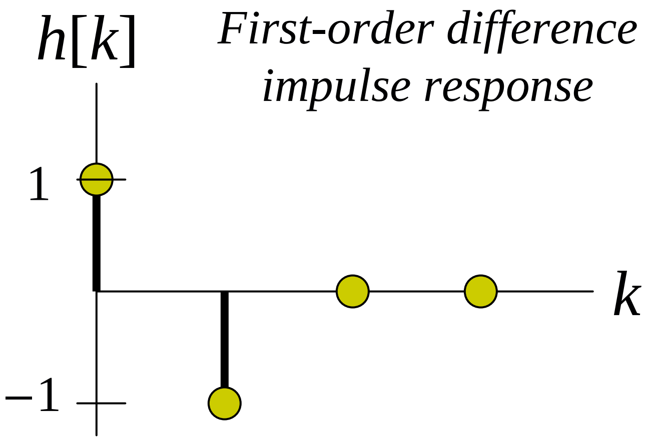
Frequency response

$$H(\omega) = e^{-j\omega}$$
$$|H(\omega)| = 1$$
$$\angle H(\omega) = -\omega$$

First-order difference FIR filter

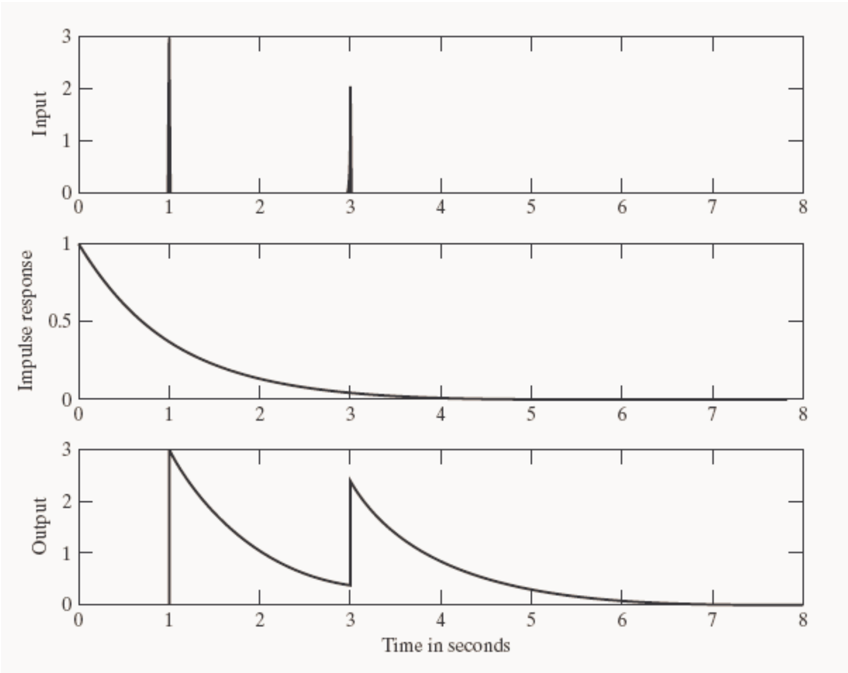
- Highpass filter (sharpens input signal)
  - Impulse response is {1, -1}

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



Example

- input: $u(t) = 3\delta(t-1) + 2\delta(t-3)$
- impulse response:
  - $h(t) = e^{-t}$ for $t \geq 0$ and 0 Otherwise
- output: $y(t) = 3h(t-1) + 2h(t-3)$



Mandrill Demo (*DSP First*)

- From lowpass filter to highpass filter
  - original → blurry → sharpened
- From highpass to lowpass filter
  - original → sharpened → blurry
- Frequencies that are zeroed out (e.g. DC) can never be recovered
- Order of two LTI systems in cascade can be switched under the assumption that the computations are performed in exact precision

Finite Impulse Response (FIR) Filters

- Duration of impulse response *h*[*k*] is finite,

$$y[k] = x[k] * h[k] = \sum_{m=-\infty}^{\infty} h[m]\, x[k-m] = \sum_{m=0}^{N-1} h[m]\, x[k-m]$$

  - Output depends on current input and previous *N*-1 inputs
  - *N* input samples in the vector

$$\{\, x[k], x[k-1], \ldots, x[k-(N-1)] \,\}$$

  - *N* nonzero values of the impulse response in vector

$$\{\, h[0], h[1], \ldots, h[N-1] \,\}$$

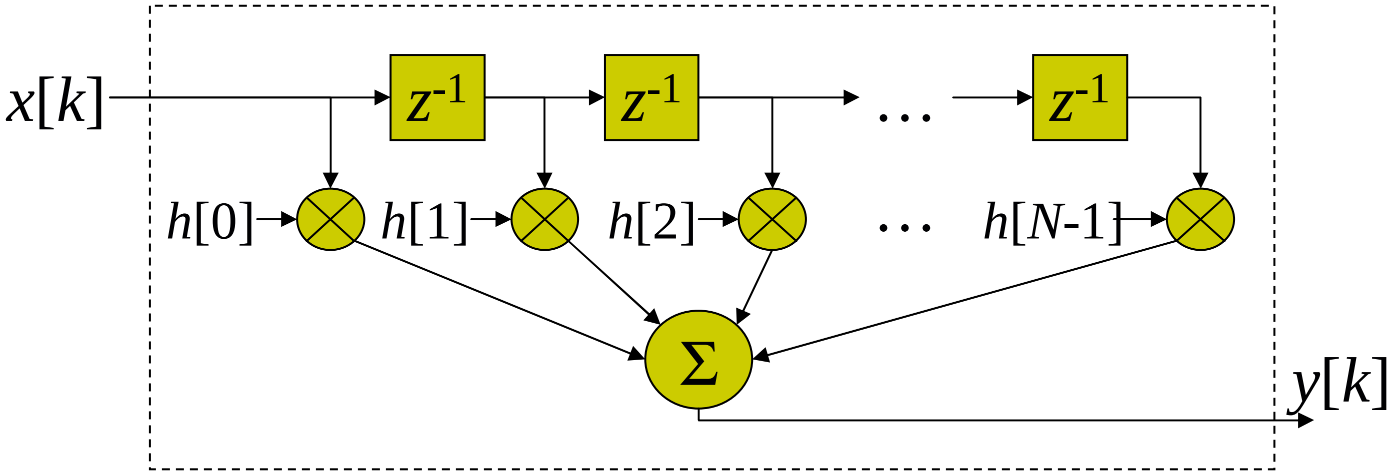
- *What instruction set/architecture features would you add to accelerate FIR filtering?*

Discrete-time Tapped Delay Line

- Assuming that $h[k]$ has finite duration from $k = 0, \ldots, N-1$

$$y[k] = \sum_{m=0}^{N-1} h[m]\, x[k-m]$$

- Block diagram of an implementation (*Direct Form*)

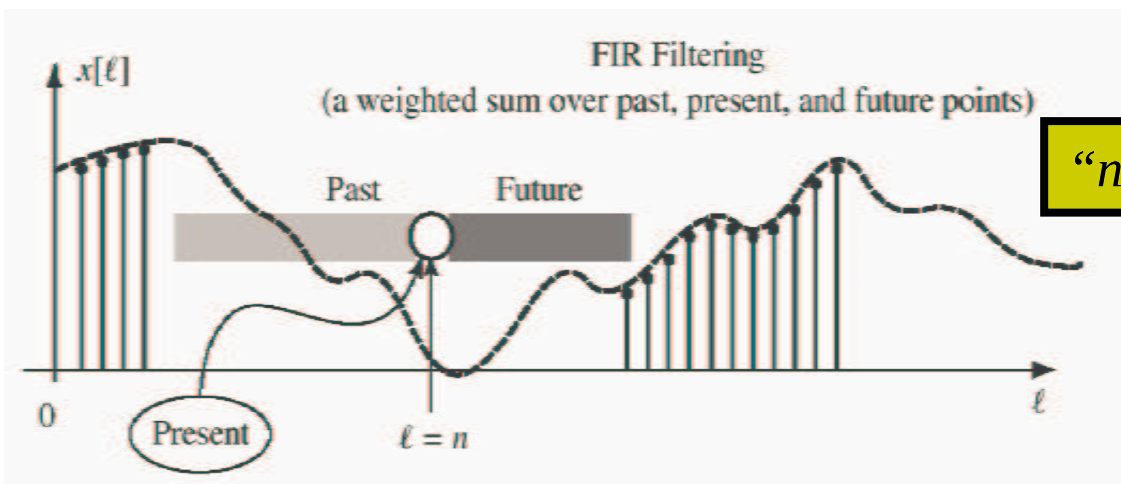


Operation of FIR Filter

- The filter output calculation within a sliding window
- x[n] is a list of numbers indexed by "*n*"

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

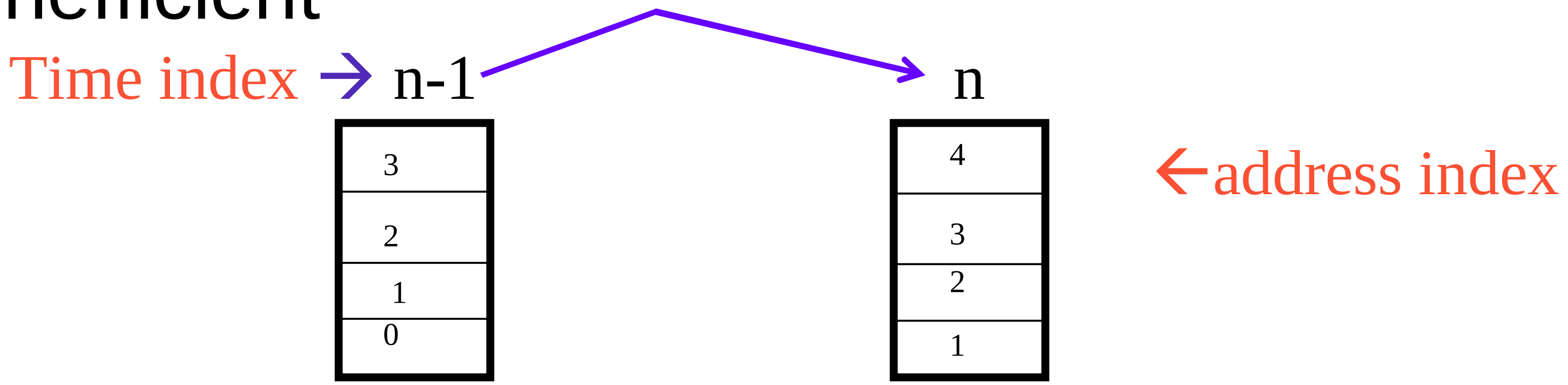
n	$n < -2$	-2	-1	0	1	2	3	4	5	$n > 5$
$x[n]$	0	0	0	2	4	**6**	**4**	**2**	0	0
$y[n]$	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	**4**	2	$\frac{2}{3}$	0	0



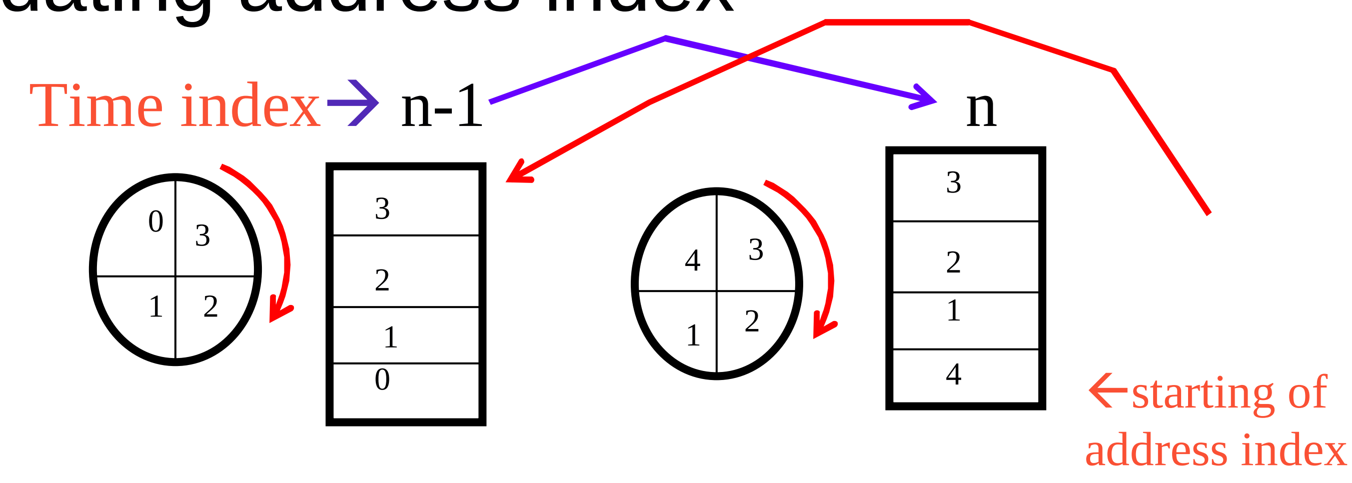
"*n*" is the time

FIR Implementation: Circular Buffer

- Shifting the elements in the entire array is inefficient



- Better approach is to use circular buffers and updating address index



Circular Buffer Implementation in C

- Oldest input sample x[n-(N-1)] is h[N-1] with the largest index
- The newest sample x[n] is multiplied by the h[0] with the smallest index.
- When a new sample is received at time n, it is written over the sample at location *oldest=newest+1* modulo N and *newest* is incremented modulo N

Array Index	Coeff. h[ ]	Circ buf x[ ]
0	h[0]	x[n-*newest*]
1	h[1]	x[n-*newest*+1]
:		
:		x[n-1]
newest		x[n]
oldest		x[n-N+1]
:		
:		
N-2	h[N-2]	x[n-*newest*-2]
N-1	h[N-1]	x[n-*newest*-1]

- Thus, data samples are written into the array in a circular fashion.

$$y[n] = \sum_{k=0}^{N-1} h[k]xcirc[mod(newest - k, N)]$$

Convolution Demos

- Johns Hopkins University Demonstrations
  - http://www.jhu.edu/~signals
  - Convolution applet to animate convolution of simple signals and hand-sketched signals
  - Convolving two rectangular pulses of same width gives a triangle whose width is twice the width of the rectangular pulses