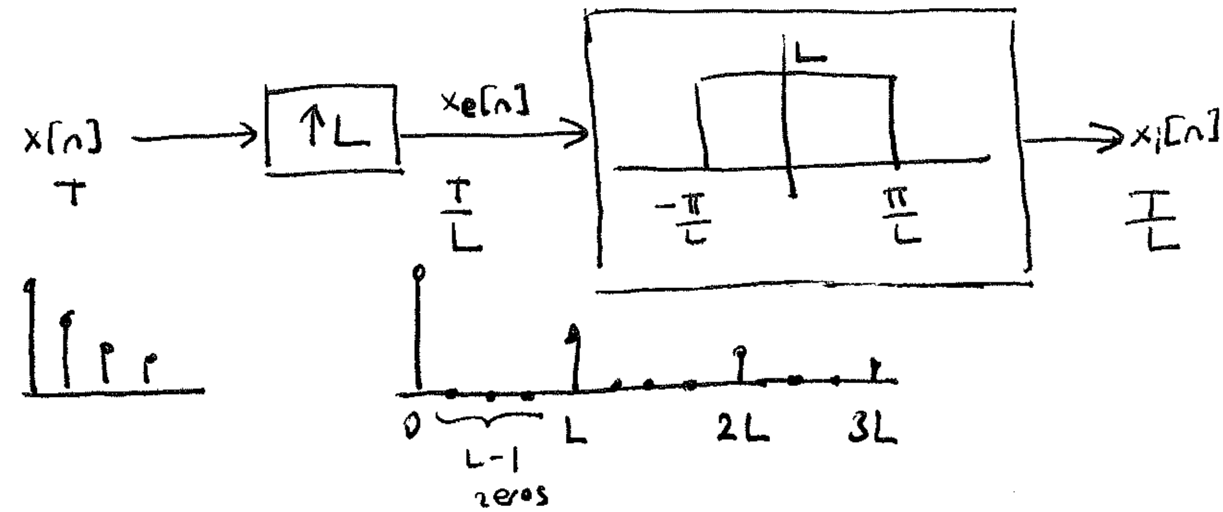
Today's Lecture

- Wrapping up downsampling and upsampling
- Polyphase and multirate signal processing

Readings : 11.1-4 Upsamping, downsampling
11.5-10 Multirate signal processing

Last lecture



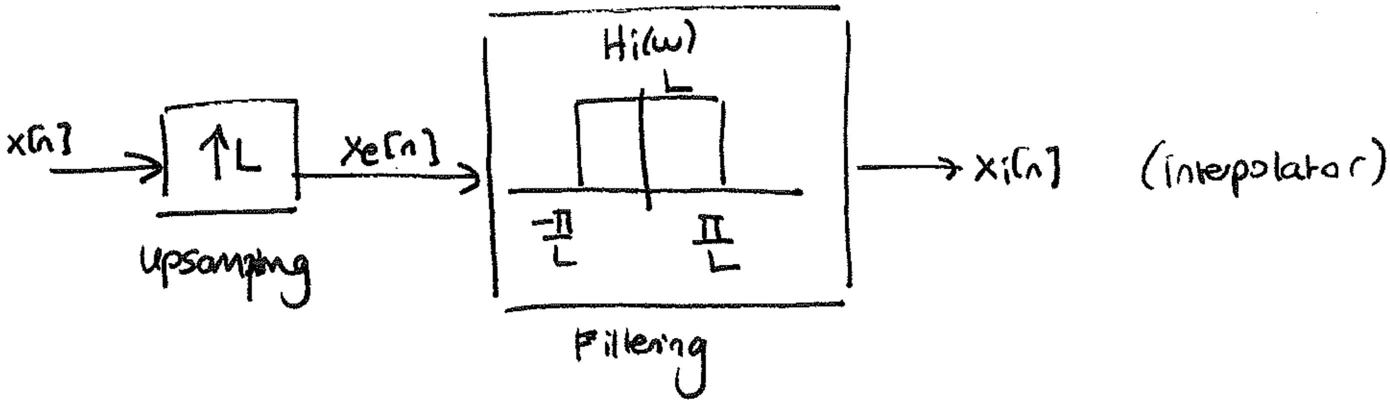
Announcements

Homework 5 posted

Homework 4 due Today

Midterm 2 11/16 on Webex

Matlab resources : ChangingSignalSampleRate.m



In time domain, $hi[n] = \text{sinc}\left(\frac{\pi n}{L}\right) = \frac{\sin(\pi n/L)}{\pi n/L}$

$$xe[n] * hi[n] = xi[n] = \sum_k xe[k]\, hi[n-k]$$

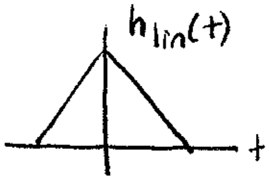
where $xe[n] = \begin{cases} x\left[\frac{n}{L}\right] & , n = 0, \pm L, \pm 2L, \ldots \\ 0 & , \text{else} \end{cases}$

$$xi[n] = \sum_{k=-\infty}^{+\infty} x[k]\, \text{sinc}\left(\frac{\pi}{L}(n-kL)\right)$$

$(k \to kL)$

$xe[kL] = x[k]$

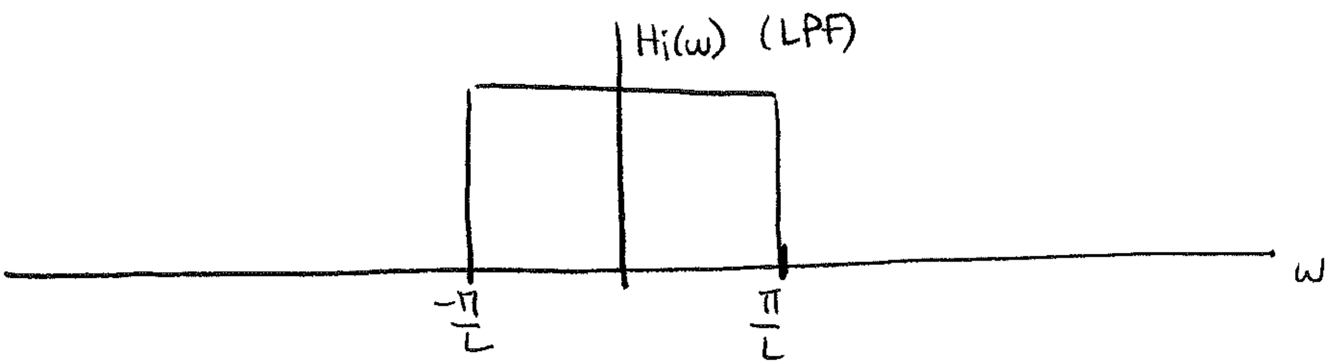
Recall that first order hold linearly interpolates 2 consecutive samples:

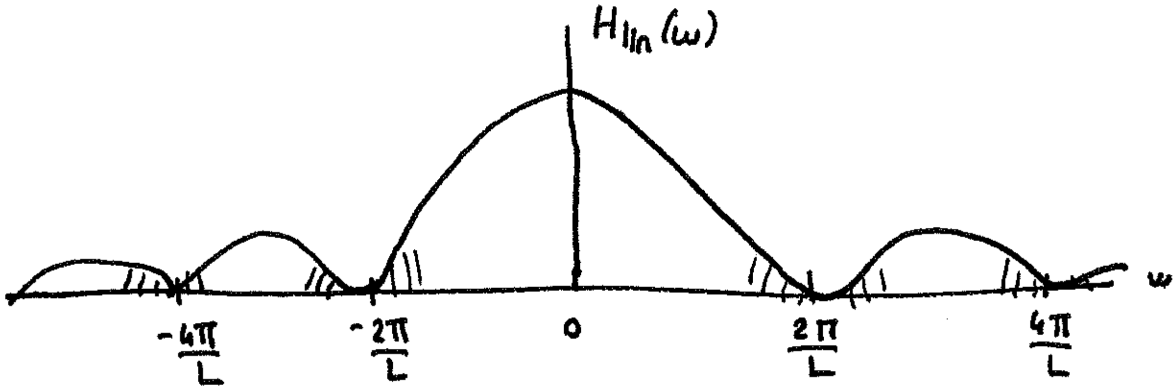


$\longrightarrow$ $H_{lin}(w) = \frac{1}{L}\left[\frac{\sin(wL/2)}{\sin(w/2)}\right]^2$

DTFT

Comparison of Linear interpolation with the low pass filtering (LPF)



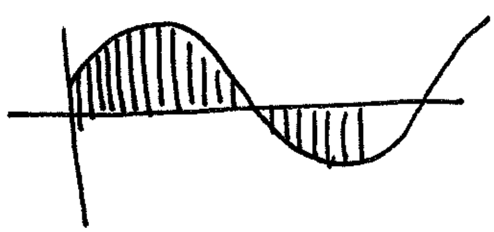


If the original signal was sampled at or near Nyquist rate, linear interpolation is not good. Why?

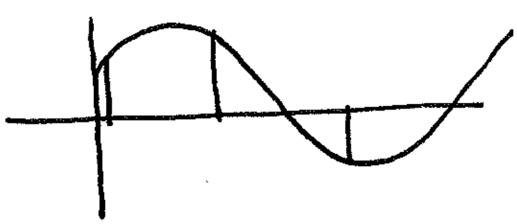
Because it passes through the signals on either side of $\frac{2\pi}{L}$.

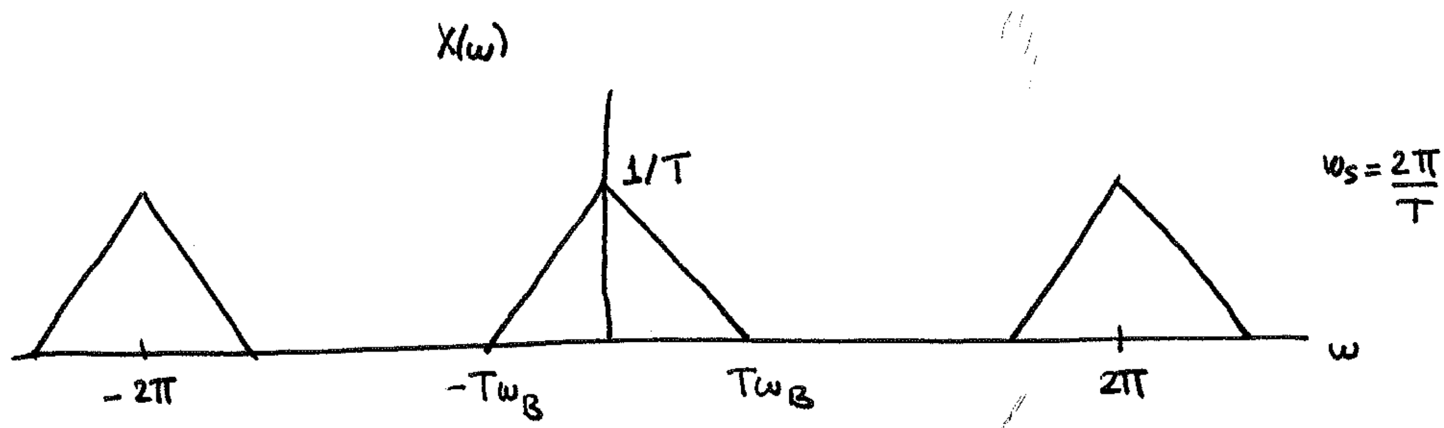
If the original signal was oversampled, the copies of the original signal will be narrow (no aliasing) in frequency domain, and linear interpolation will be good.

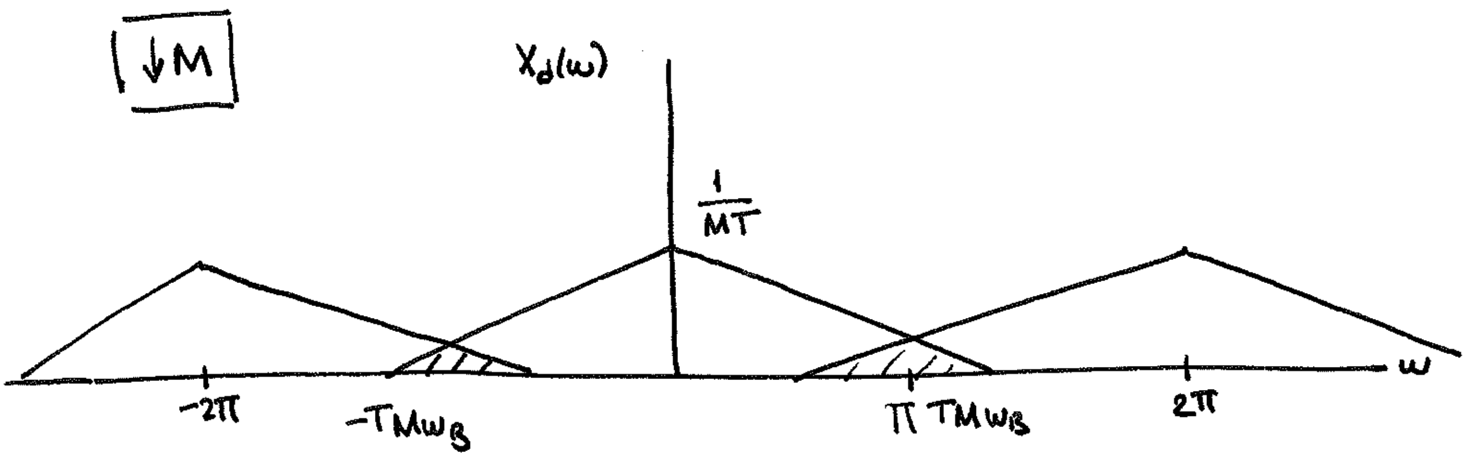
Time domain interpretation: If a signal is oversampled, the adjacent copies are very close to each other and linear interpolation will be quite accurate.

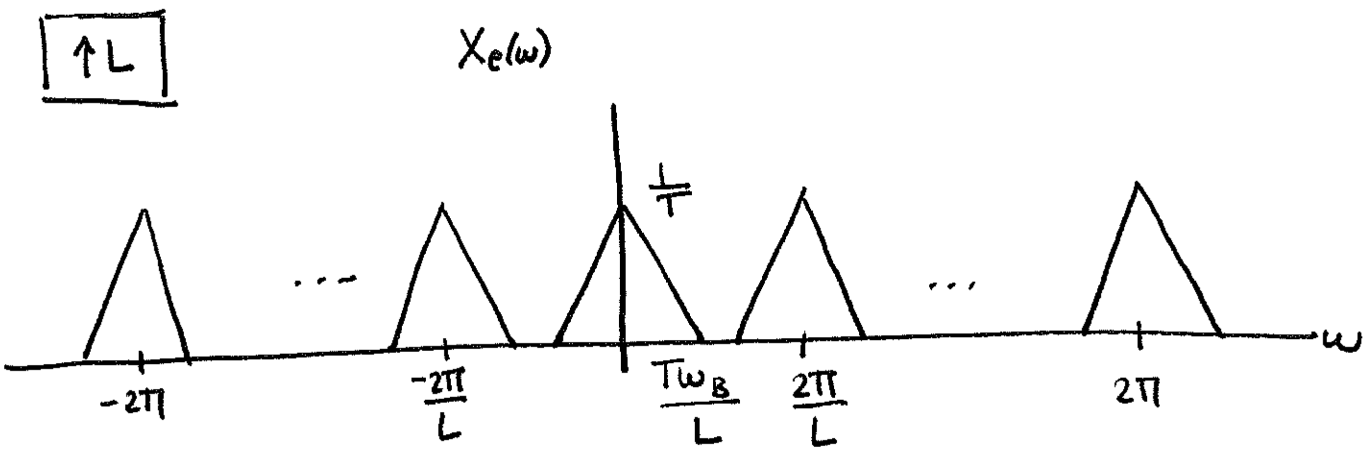


→ linear interpolation is good

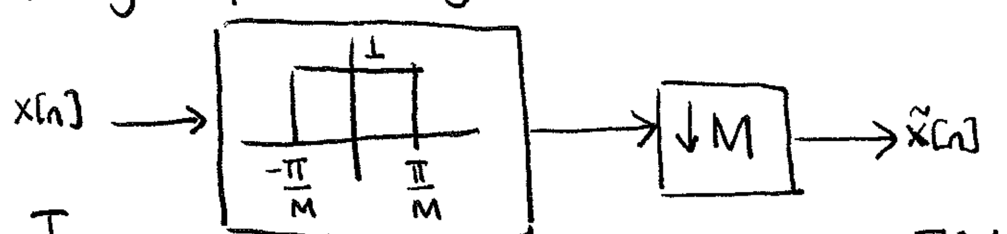






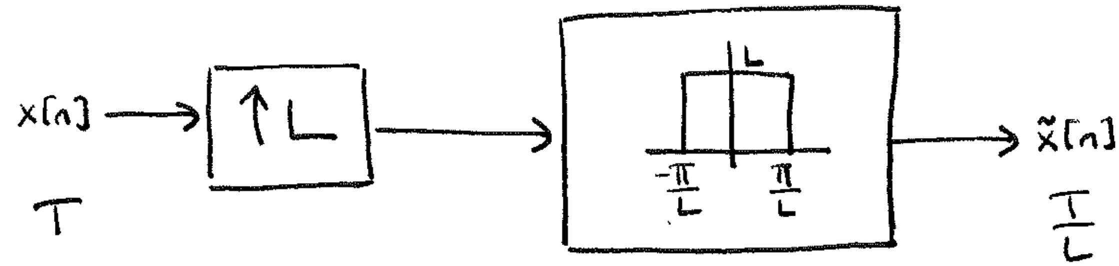


Prefiltering to prevent aliasing when downsampling



Loose high frequency components

Lowpass filter to interpolate missing values when upsampling



Question: Can we change the sampling rate by a non-integer value z?

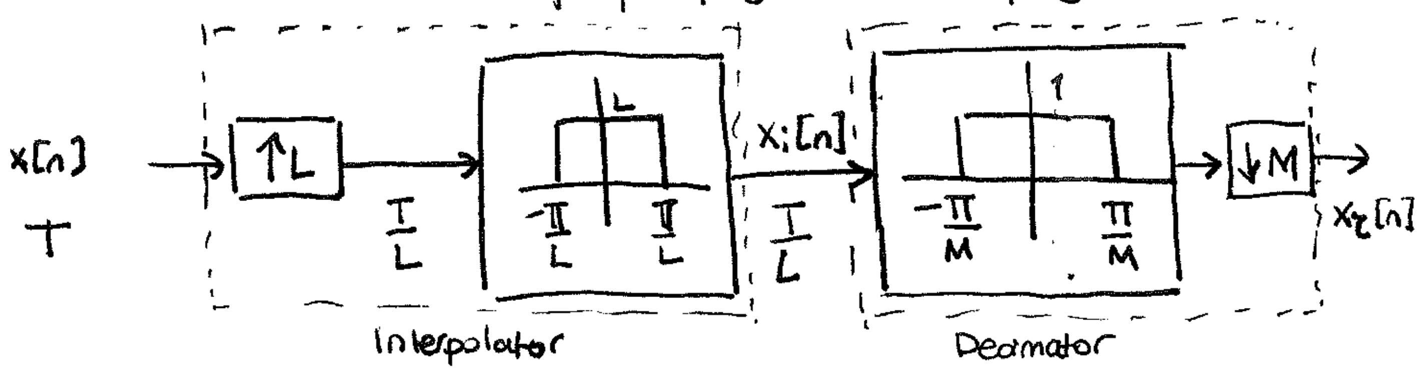
MATLAB example

Sampling rate of <u>compact disks</u> 44.1 KHz

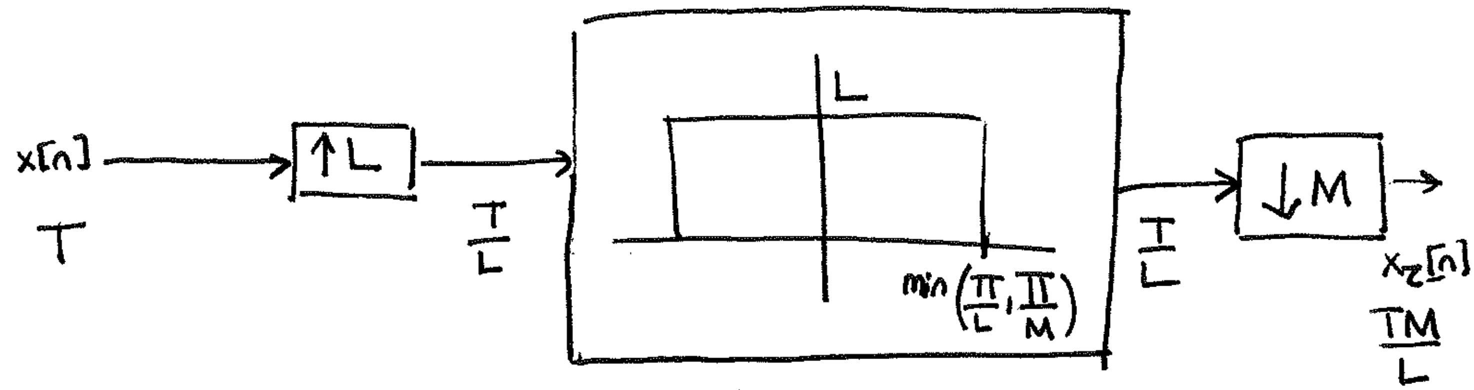
Sampling rate of <u>audio tape</u> 48 KHz

$$\frac{48}{44.1} = \frac{160}{147} = \frac{M}{L} = z$$

This is a combination of upsampling and downsampling.



$$z = \frac{TM}{L}$$



When $M > L$, net reduction in sampling rate

(low pass filter to prevent aliasing)

$M < L$, net increase in sampling rate

(I don't need the second filter)

Cons: $z = 1.001$, $M = 1001$, $L = 1000$

Big intermediate changes in rate to get almost the same signal.

⇒ Alternative: Multi-rate signal processing to reduce computations in A/D or D/A conversion and sample rate conversion.

Interchanging Filtering and Down/Up Sampling

1. Equivalent systems

$$x[n] \rightarrow \boxed{\downarrow M} \rightarrow x_a[n] \rightarrow \boxed{H(z)} \rightarrow y_a[n]$$

$$x[n] \rightarrow \boxed{H(z^M)} \rightarrow x_b[n] \rightarrow \boxed{\downarrow M} \rightarrow y_b[n]$$

We will next show $y_a[n] = y_b[n]$.

$x_a[n] = x[Mn]$

$X_a(\omega) H(\omega) = Y_a(\omega)$ DTFT

$H(z^M)$?

$$= \sum_{k=\infty}^{\infty} h[k] z^{-kM}$$

$$= \sum_{n=0, \pm M, \pm 2M \dots} h\left[\frac{n}{M}\right] z^{-n} = \sum_{n} h_e[n] z^{-n}$$

$z = e^{j\omega}$

$z^M = e^{j\omega M}$

$$\Rightarrow X(\omega) H(\omega M) = X_b(\omega)$$

$$Y_b(\omega) = \frac{1}{M} \sum_{m=0}^{M-1} X_b\left(\frac{\omega}{M} - \frac{2\pi m}{M}\right)$$ (last 2 lectures)

$$= \underbrace{\frac{1}{M} \sum_{m=0}^{M-1} X\left(\frac{\omega}{M} - \frac{2\pi m}{M}\right)}_{X_a(\omega)} \underbrace{H(\omega - 2\pi m)}_{= H(\omega) \text{ because } 2\pi \text{ periodic}}$$ DTFT

$$= H(\omega) X_a(\omega)$$

$$= Y_a(\omega)$$

2. Equivalent systems

$$x[n] \longrightarrow \boxed{H(z)} \longrightarrow x_b[n] \longrightarrow \boxed{\uparrow L} \longrightarrow y_b[n]$$

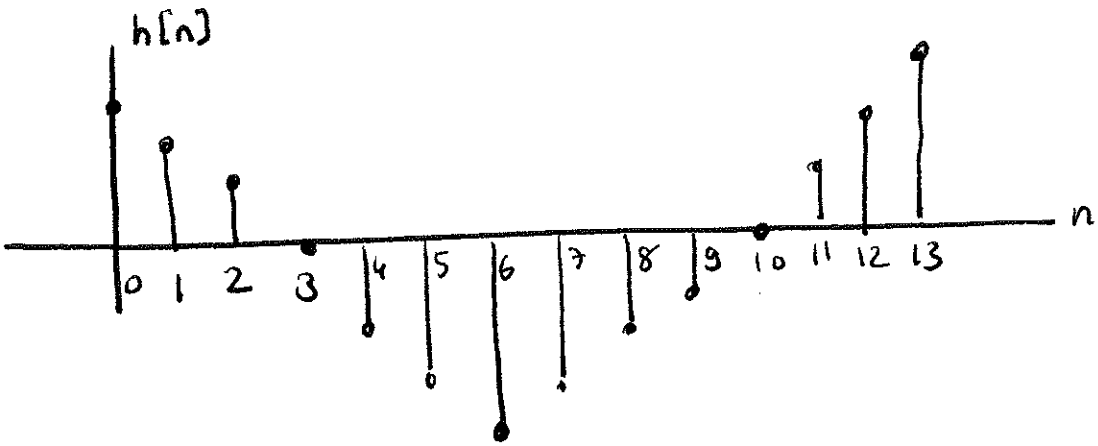
$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow x_a[n] \longrightarrow \boxed{H(z^L)} \longrightarrow y_a[n]$$

We will next show $y_a[n] = y_b[n]$.

$$Y_b(\omega) = X_b(\omega L) = X(\omega L) H(\omega L)$$

$$X_a(\omega) = X(\omega L) \implies Y_a(\omega) = X_a(\omega) H(\omega L) \qquad \text{DTFT}$$
$$= X(\omega L) H(\omega L)$$
$$= Y_b(\omega)$$

Polyphase Decompositions



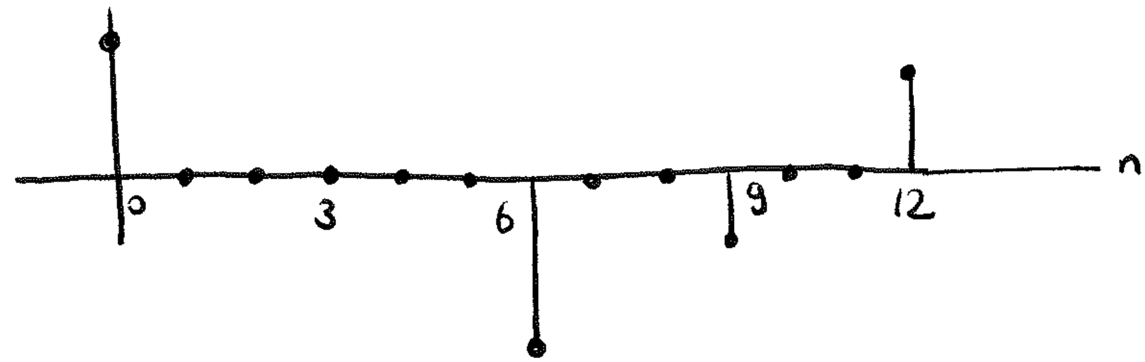
Decompose as a sum of M subsequences

$$h_k[n] = \begin{cases} h[n+k], & n = aM \quad \text{where } a \text{ is an integer} \\ 0, & \text{else.} \end{cases}$$

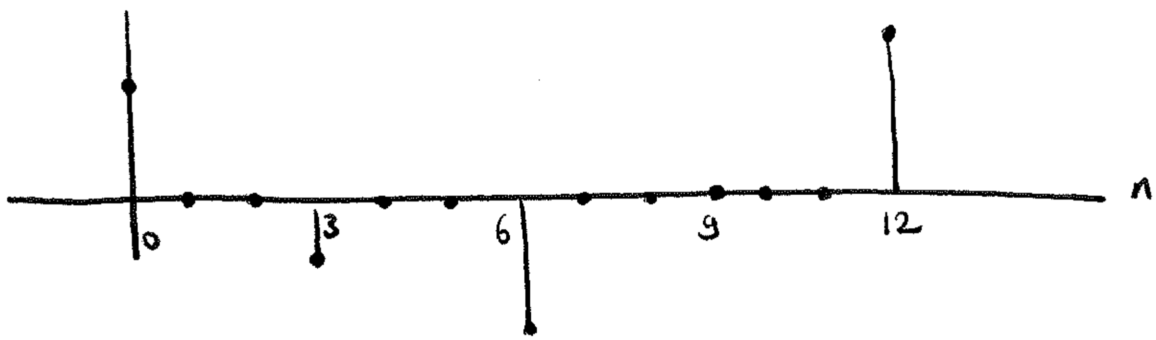
Example

$\underline{M=3}$ (# of subsequences)

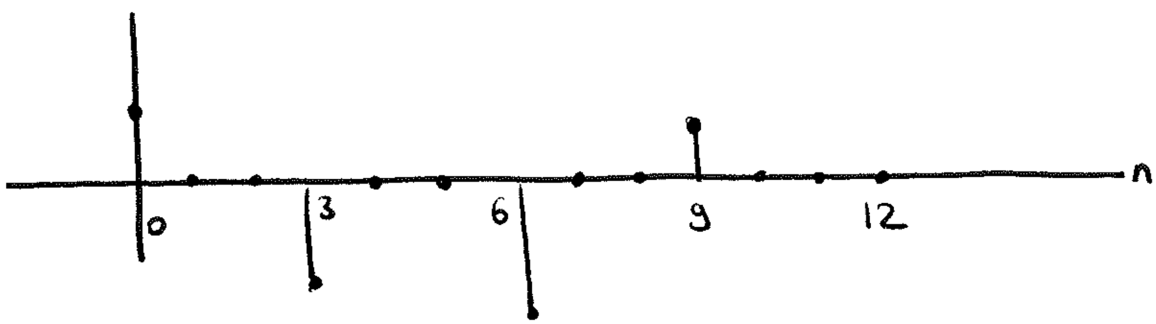
$h_0[n]$



$h_1[n]$



$h_2[n]$

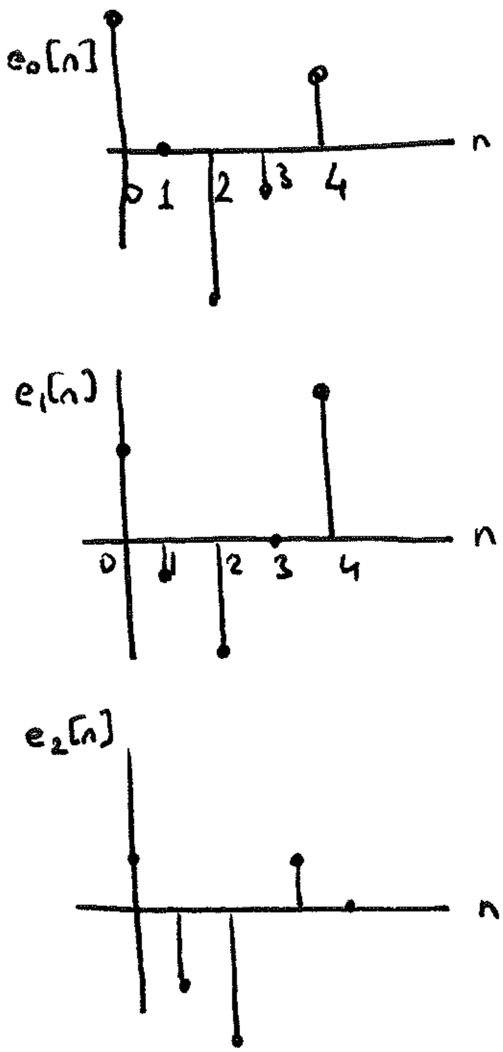


We can write $h[n]$ as sum of delayed subsequences:

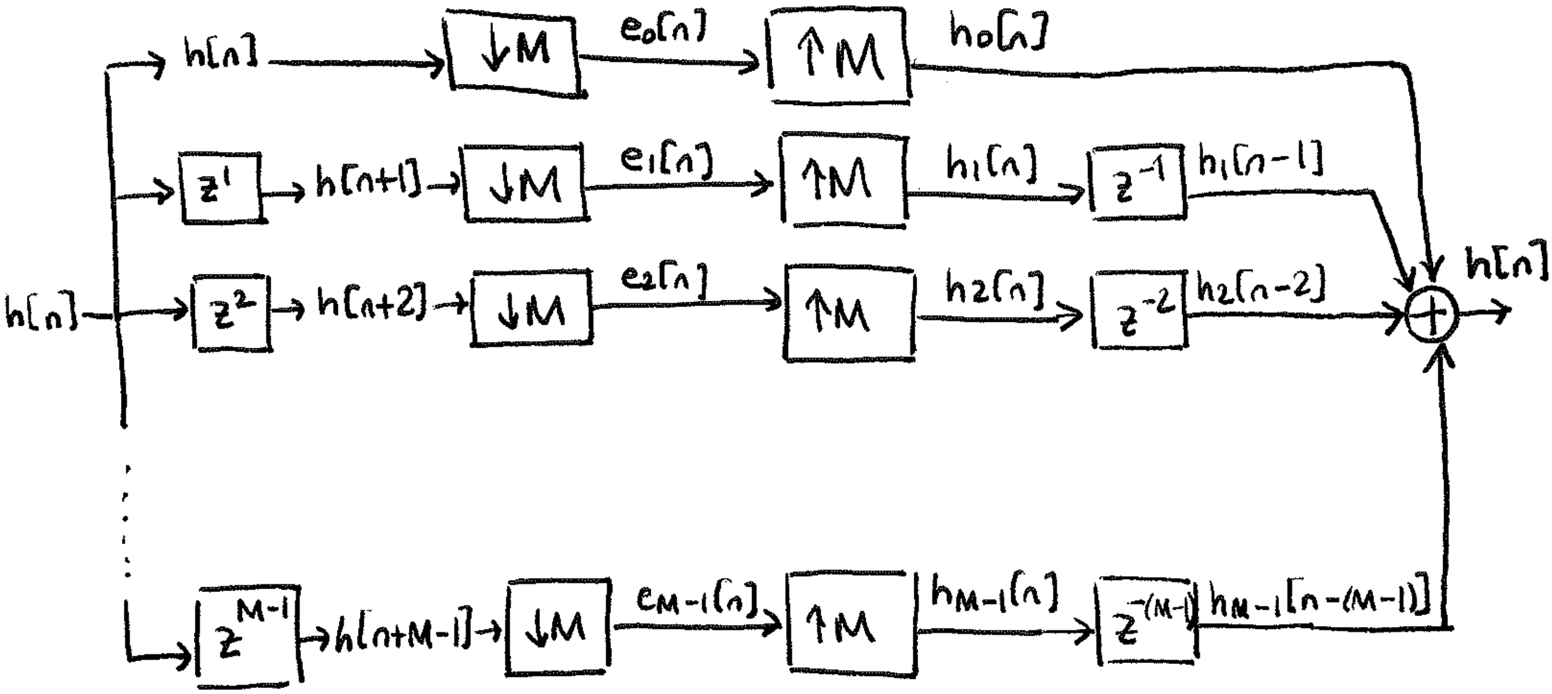
$$h[n] = h_0[n] + h_1[n-1] + h_2[n-2]$$

$$= \sum_{k=0}^{M-1} h_k[n-k]$$

$\boxed{e_k[n] = h_k[nM] = h[nM+k]}$ are called the polyphase components of $h[n]$.



$M = 3$



Recall $\quad h[n] \rightarrow \boxed{z} \rightarrow h[n+1] \qquad h[n] \rightarrow \boxed{z^{-1}} \rightarrow h[n-1]$

We can chain the delay elements and show that