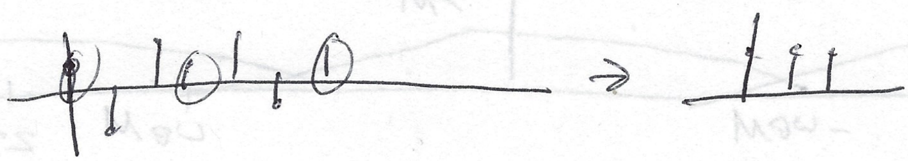
15 multirate signal processing and polyphase representations

Last time

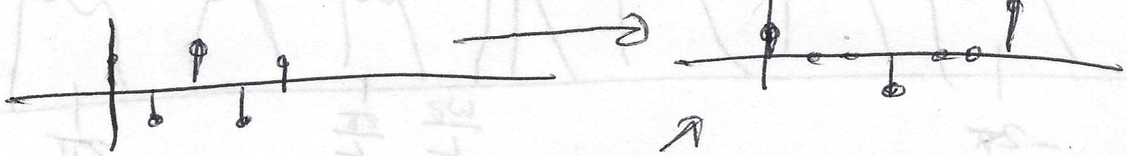
$x[n] \rightarrow \boxed{\downarrow M} \rightarrow x_{down}[n]$

compressor



$x[n] \rightarrow \boxed{\uparrow L} \rightarrow x_{up}[n]$

expander

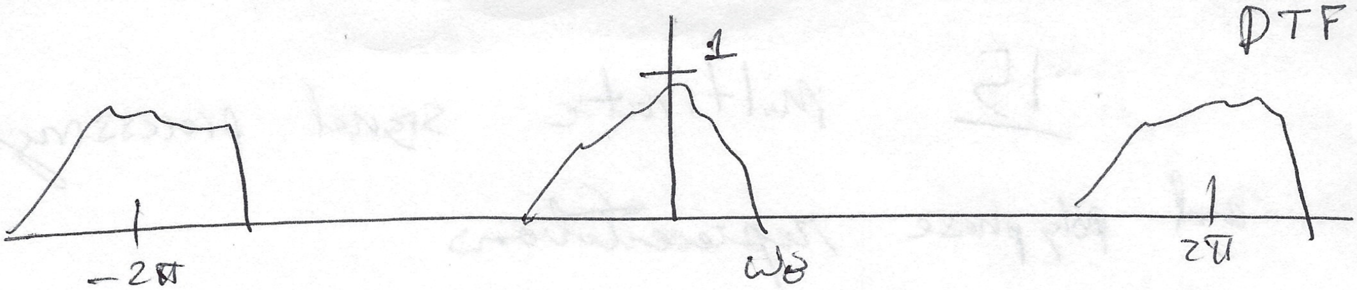


run this through a low pass filter, you could recover an interpolation of the signal that would leave the correct set of samples inbetween the elements (Assuming we did sample over the Nyquist rate)

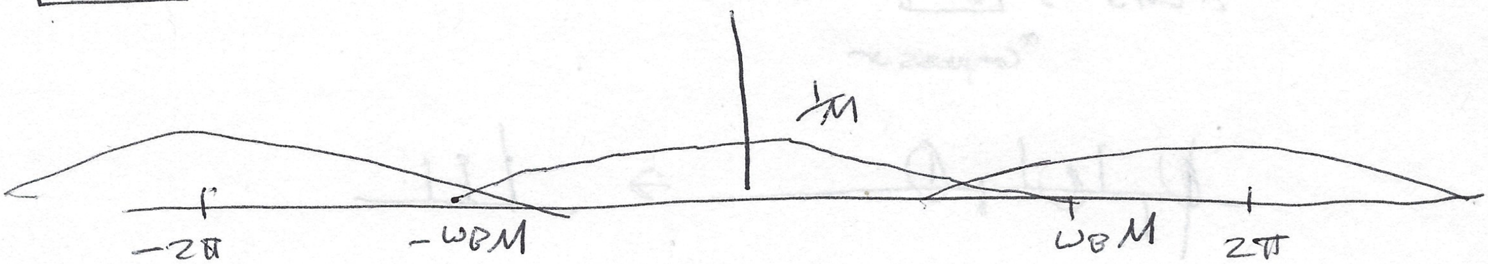
we could introduce ~~Down sample~~ Aliasing by down sampling

Things are good with up sampling

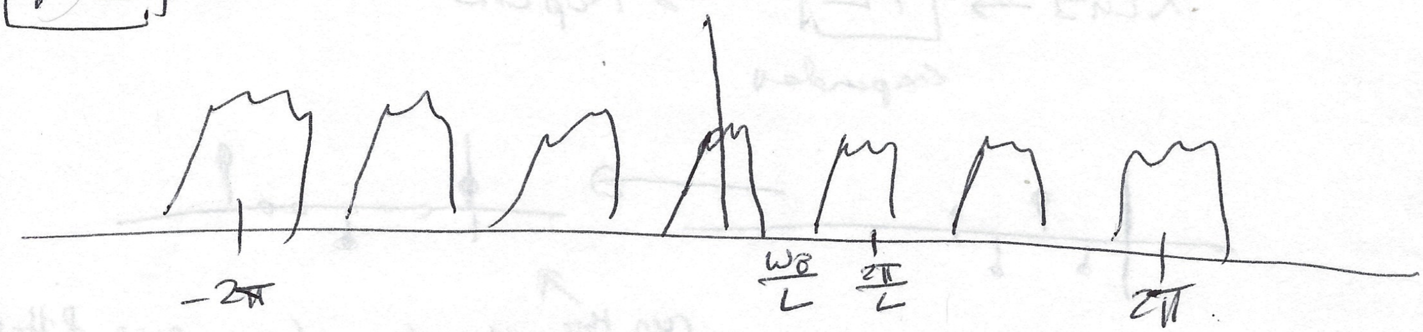
DTFT



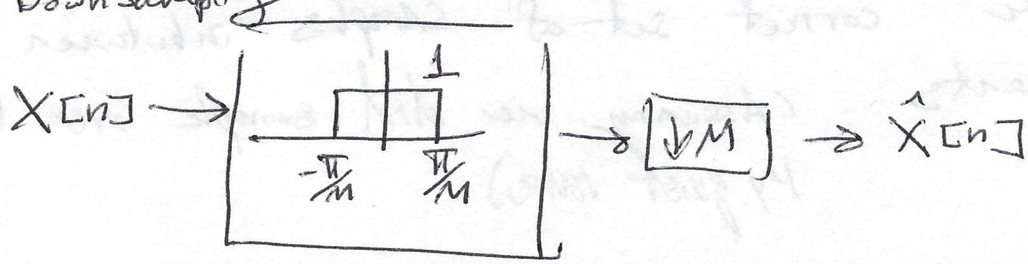
$\boxed{\downarrow M}$ spreads out by a factor of M and decreases by a factor of M



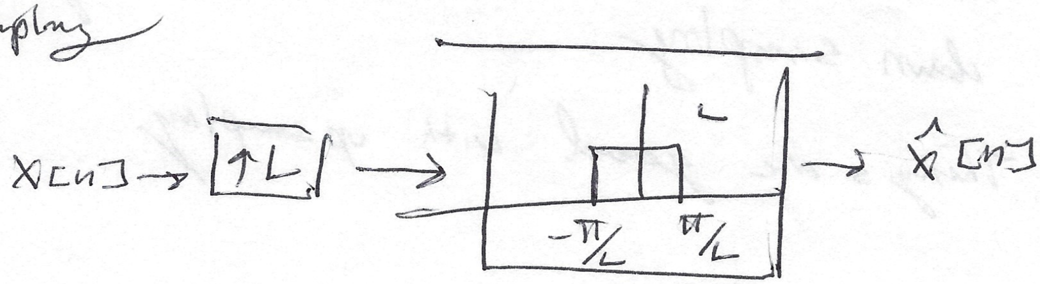
$\boxed{\uparrow L}$



Prefiltering to Prevent Aliasing when Downsampling



Low pass filter to interpolate missing values when upsampling



we made the assumption that M and L satisfied

$M > 0$, $L > 0$ $\quad M \in \mathbb{Z}$ $\quad L \in \mathbb{Z}$

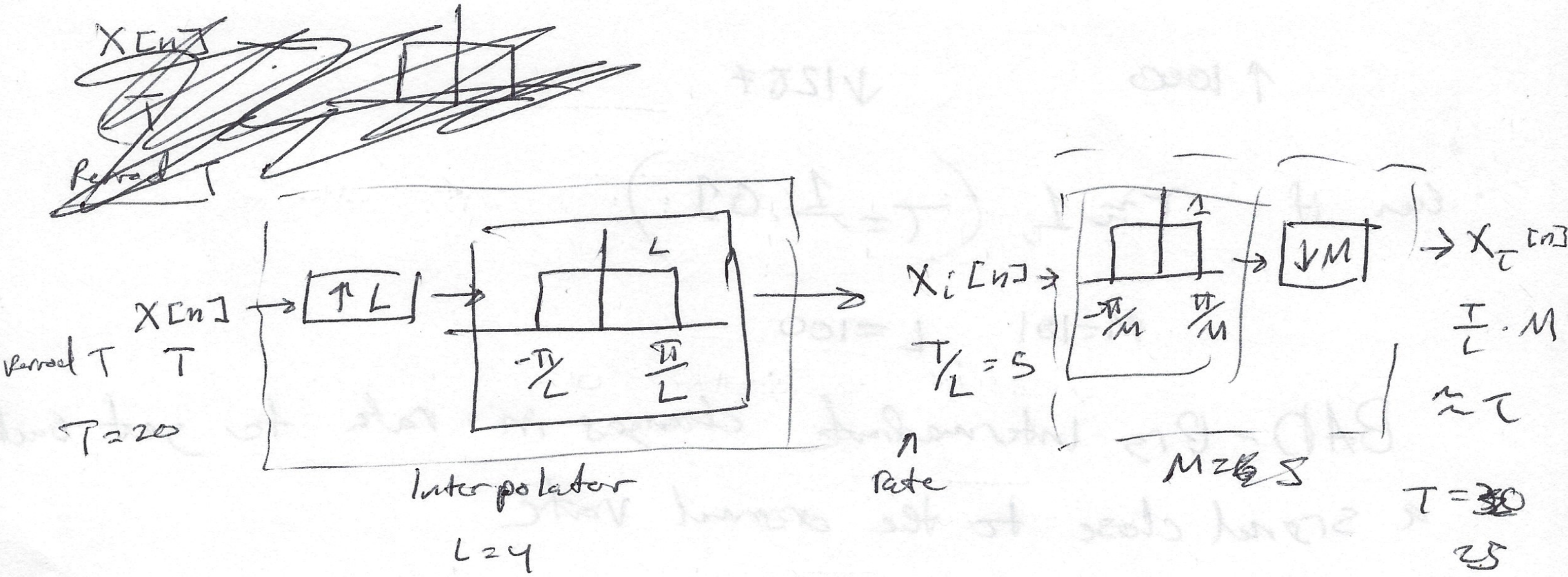
The Real world system may expect a certain rate, So we need to up or down Sample to that rate for that block of system to work

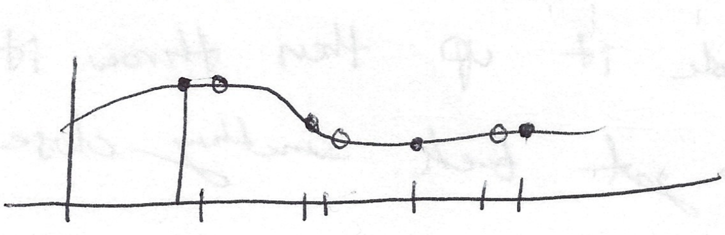
Say that we need to go from 10 KHz to 12 KHz

What about changing the Sampling rate by a non integer factor τ?

Approximate τ as some rational number $\tau = \frac{M}{L}$

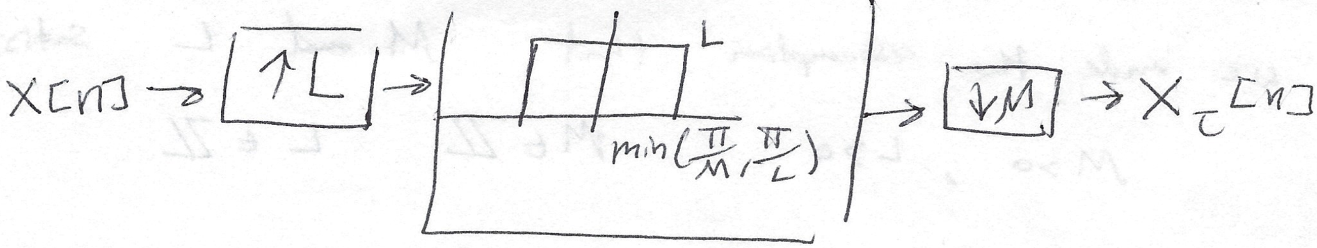
Combination of upsa





$$T \frac{M}{L} = T_\tau$$

④



$\tau = \frac{M}{L}$, if $M > L$ → Net reduction in Sampling rate

→ $\frac{\pi}{M}$ will be the bandwidth of the filter

(we need prefilter to possibly prevent aliasing)

$M < L$, Net increase in Sampling rate

→ $\frac{\pi}{L}$ is more present

(No need to prefilter) (No Aliasing can occur)

Say $\tau = 1.2$, upsample by 5, down sample by 6

τ ~ inversely proportional to the Sample rate

what if 1.287?

↑1000 ↓1287

Even if $\tau \approx 1$, ($\tau = 1.01$)

$M = 101$, $L = 100$

BAD - Big intermediate changes in rate to get back a signal close to the original rate

Why sample it up then throw it all away to get back something close?

CAN This be done More efficiently ?

Yea!

→ Multirate Signal processing → Explore multirate stuff as a Project

→ Lets us avoid unnecessary computations

USEFul Identities

$$X[n] \rightarrow \boxed{\downarrow M} \rightarrow X_a[n] \rightarrow \boxed{H(z)} \rightarrow Y_a[n]$$

AND

$$X[n] \rightarrow \boxed{H(z^m)} \rightarrow X_b[n] \rightarrow \boxed{\downarrow M} \rightarrow Y_b[n]$$

Are equivalent systems

[What is $H(z^M)$? $\quad H(z) = \sum_{n=-\infty}^{\infty} X[n] z^{-n}$

$$H(z^M) = \sum_{n=-\infty}^{\infty} h[n] z^{-Mn}$$

We have multiples of z^M, $\underset{n=1}{+z^{-M}}$, $\underset{n=2}{z^{-2M}}$, $\underset{n=3}{z^{-3M}}$, $\underset{n=-1}{z^{M}}$, $\underset{n=-2}{z^{2M}}$

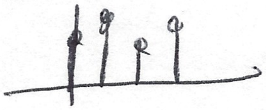
Time domain: $\delta[n-1] \sim z^{-1}$ $\quad \delta[n-2] \sim z^{-2}$

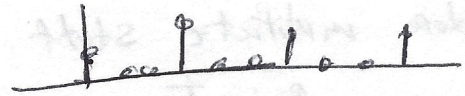
$\delta[n-M] \sim z^{-M}$ $\quad \delta[n-2M \sim z^{-2M}$.

We See $\quad H(z^M) = Z(h_e[n])$

$\quad$ ↖ expanded $h[n]$

(6)

 $h[n]$

 $h_c[n]$ where $M=3$

$$h[0] \rightarrow z^0$$

what used to be $h[1] \longrightarrow z^{-3}$ as $h[3] \rightarrow z^{-3}$

as $h_e[n] = h[\frac{n}{M}]$

Z transform of a signal that has the zeros inserted into the middle

Proof:

$z = e^{jw}$ $\quad H(wM) = H(e^{jwM})$

$$X_b(\omega) = X(\omega)H(\omega M)$$
$$= X(e^{jw})H(e^{jwM})$$

$$Y_b(\omega) = \frac{1}{M}\sum_{i=0}^{M-1} X_b\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)$$ spread out and shrunk copies

~~$H(\omega - 2\pi i)$~~

~~$= H(\omega) \frac{1}{M} \sum^{M-1}$~~

$$Y_b(\omega) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)\underbrace{H(\omega - 2\pi i)}_{H(\omega)}$$ 2π periodic by definition

$$= H(\omega)\frac{1}{M}\sum_{i=0}^{M-1} X\left(\frac{\omega}{M} - \frac{2\pi i}{M}\right)$$

$$= H(\omega)X_a(\omega)$$ if you down sampled 1st, you would get spread out copies of X

from $X_a[n] \rightarrow \boxed{H(z)} \rightarrow Y_b[n]$ or $Y_a[n]$

Also,

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

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

$Y_a(\omega) = X_a(\omega L)$ ← we shrink the frequency domain

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

$X_b(\omega) = X(\omega L)$ ← shrink down original X

$Y_b(\omega) = X(\omega L)H(z^L) \rightarrow \quad H(z^L) = H(e^{j\omega L})$

$$\rightarrow H(\omega L)$$

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

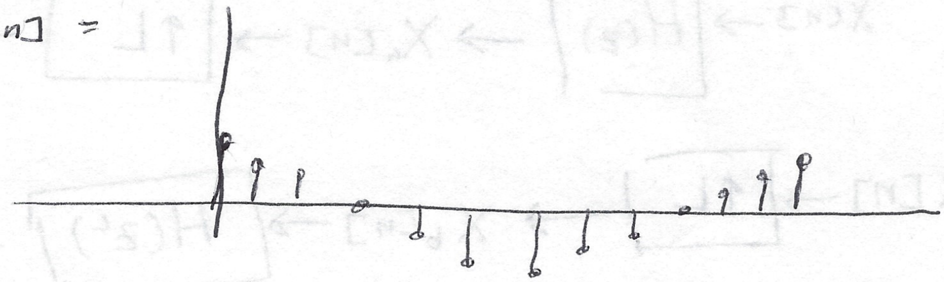
$$= Y_a(\omega)$$

The bottom branch is inefficient, to place in a bunch of zero's then filter it.

Just filter <u>then up sample</u> get the same result

⑧ Poly phase Decomposition

Consider $h[n]$ =



Decompose this as the sum of M ~~Subsequences~~ Subsequences

$$h_k[n] = \begin{cases} h[n+k] & n = (\text{integer})\, M \\ 0 & \text{Else} \end{cases}$$

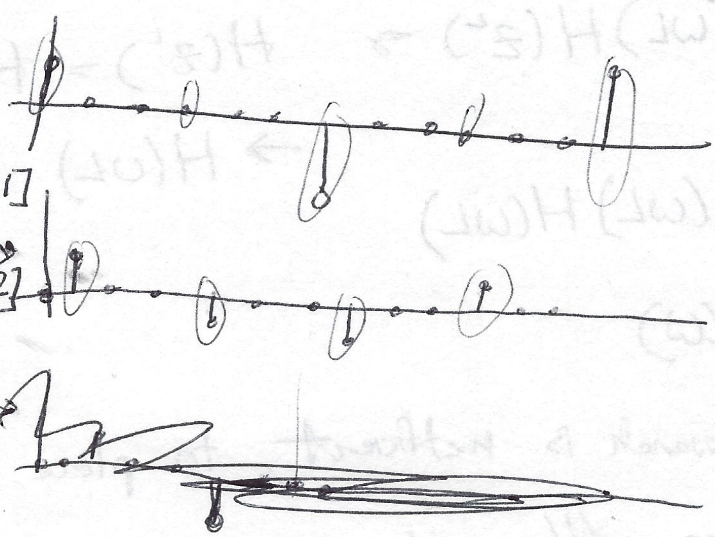
Ex $M = 3$

Take every 3rd element starting at 0 → $h_0[n]$

$h_1[n-1]$

$h_2[n-2]$

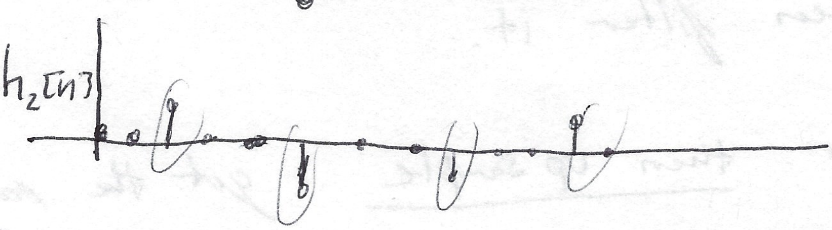
Same but starting at 1



These are wrong! They should start at zero!

Non zero →

$h_2[n]$



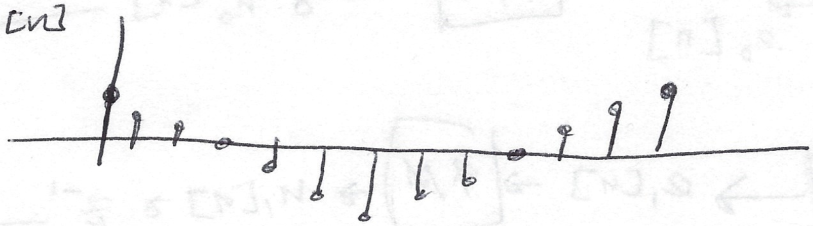
~~Write original signal~~

~~$h[n] = \sum^{M-1}$~~

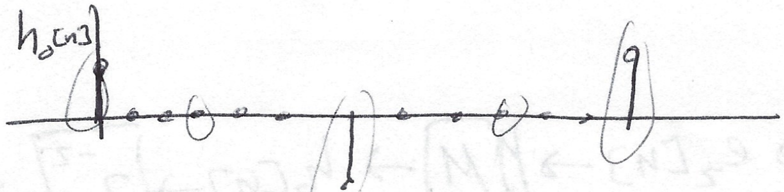
Correctly done

3 cr
3 cr
4 cr

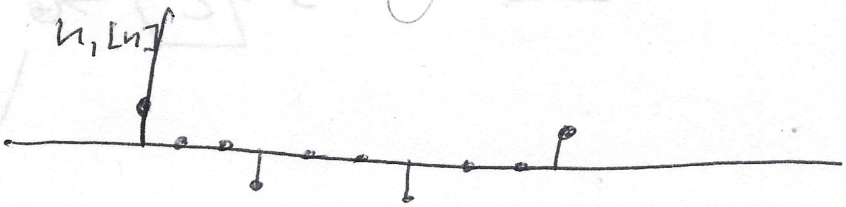
$h[n]$



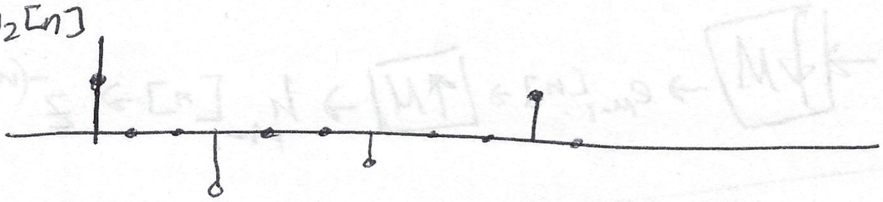
$h_0[n]$



$h_1[n]$



$h_2[n]$



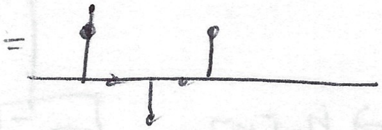
$$\& \quad h[n] = h_0[n] + h_1[n-1] + h_2[n-2]$$

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

want to get rid of the zeros inbetween?

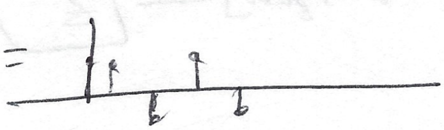
$$e_k[n] = h_k[nM]$$

$e_0[n] =$

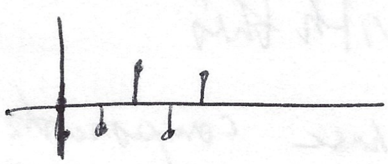


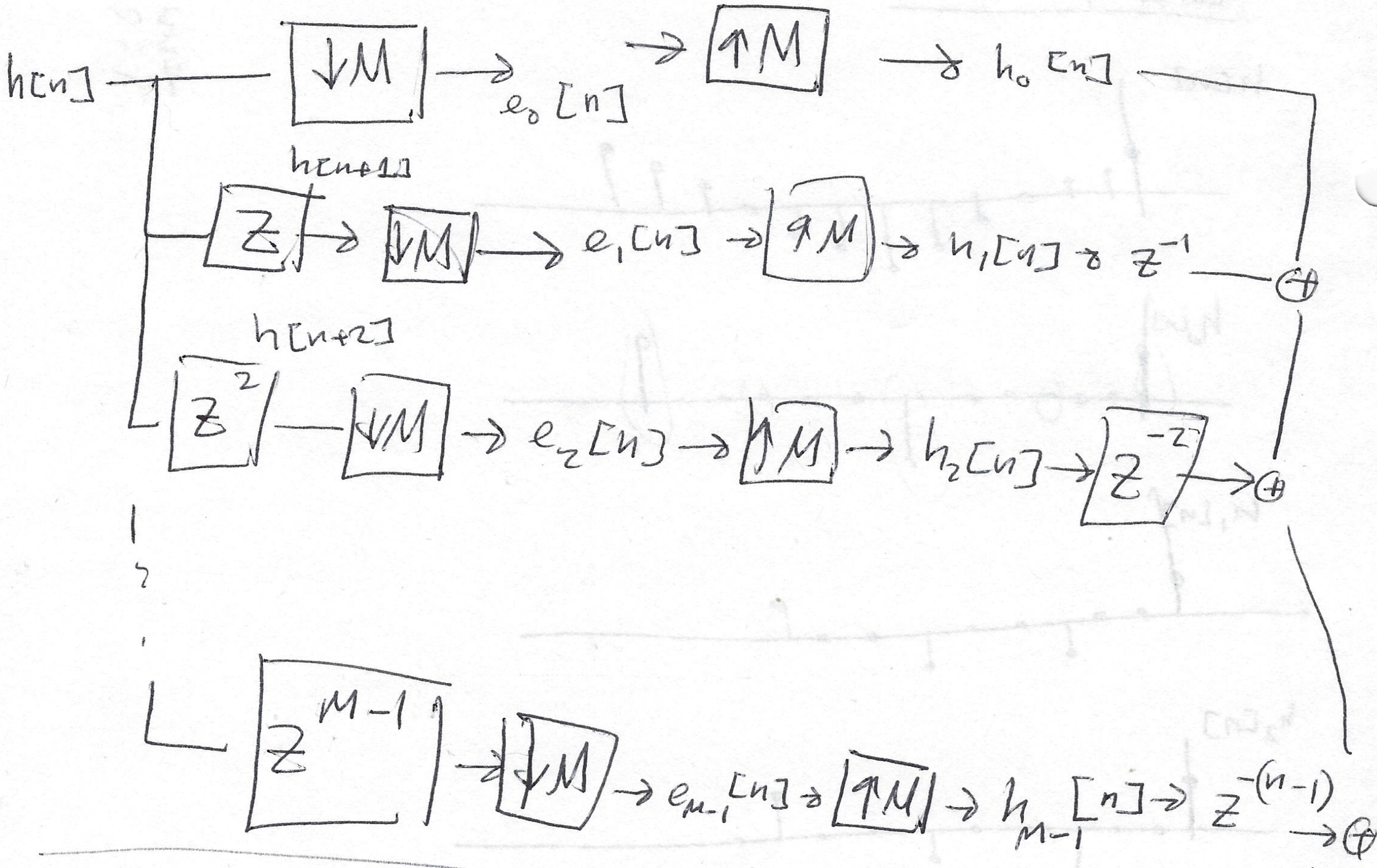
removing the zeros from $h_n[n]$

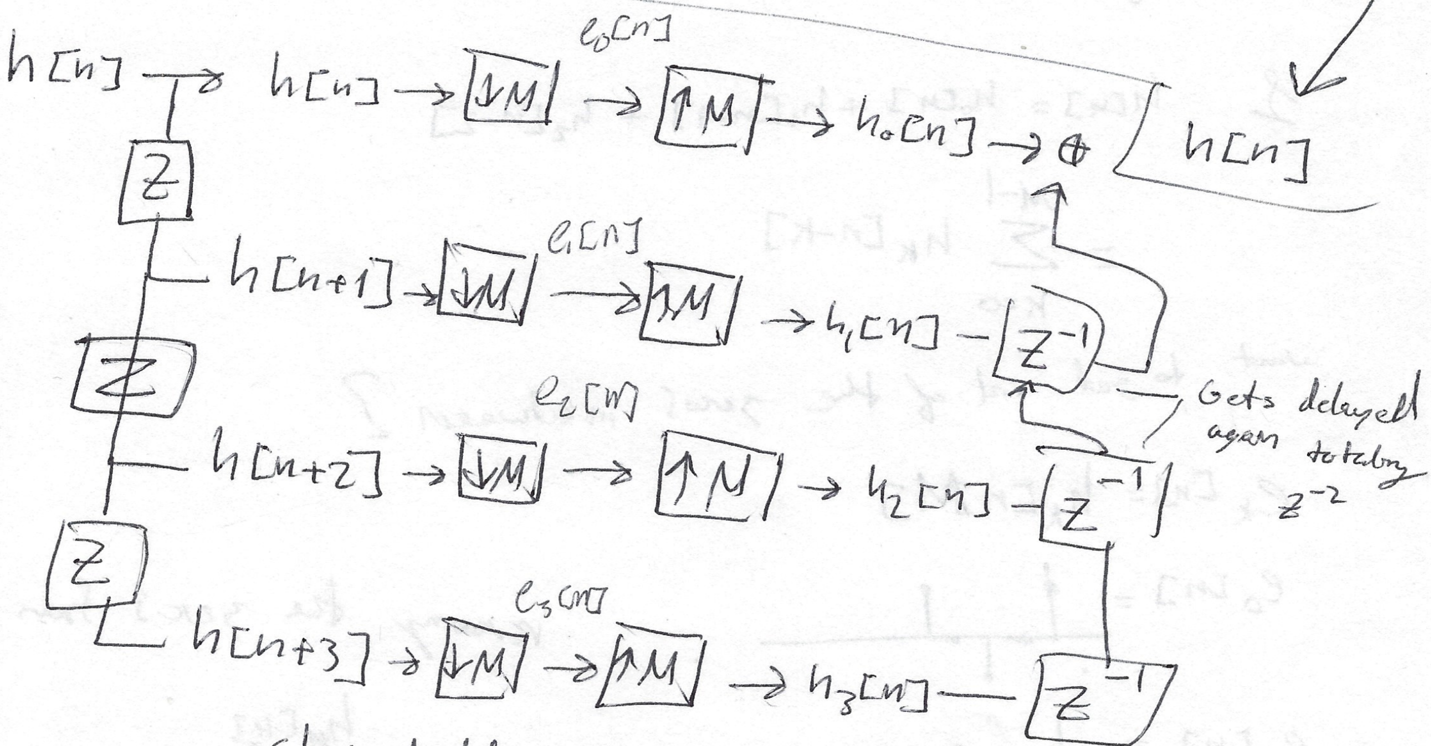
$e_1[n] =$



$e_2[n] =$







why bother with this

The e's polyphase components of h

$$e_k[n] = h[nM+k] = h_k[nM]$$

Are called the polyphase components of $h[n]$

$$H(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k}$$

$$= \sum_{l=-\infty}^{\infty} h[lM]z^{-lM} + \sum_{k=-\infty}^{\infty} h[kM+1]z^{-(kM+1)}$$

$$+ \cdots$$

$$= \sum_{k=0}^{M-1} \sum_{l=-\infty}^{\infty} h[lM+k]z^{-(lM+k)}$$

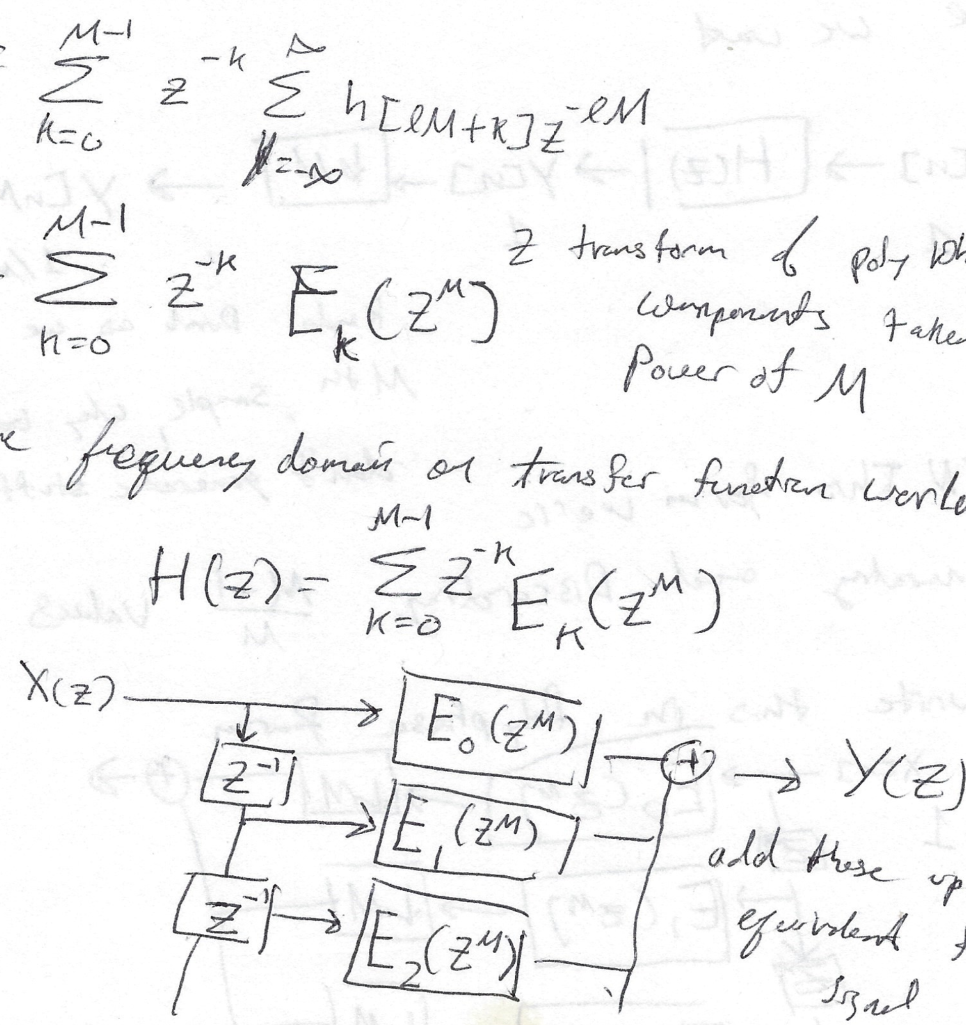
$$= \sum_{k=0}^{M-1} z^{-k} \sum_{l=-\infty}^{\infty} h[lM+k]z^{-lM}$$

$$= \sum_{k=0}^{M-1} z^{-k} E_k(z^M)$$

z transform of poly phase components taken to the power of M

IN the frequency domain or transfer function world

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M)$$



add these up to get equivalent filtered signal

(12) Take the Z transform apart into easier M Z transforms

Equivalent Z transforms of $Y(z)$

Polyphase realization of $H(z)$

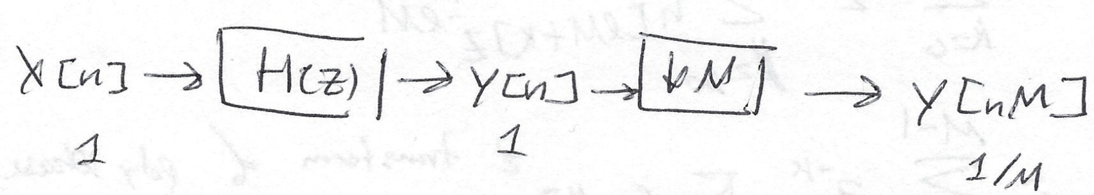
HW Problem, take a simple filter, and show me How to Take it apart into these, for example 3 Poly phase pieces.

why do this

Consider a system where we have to filter then down sample

Poly Phase implementation of Decimation or Interpolation filters

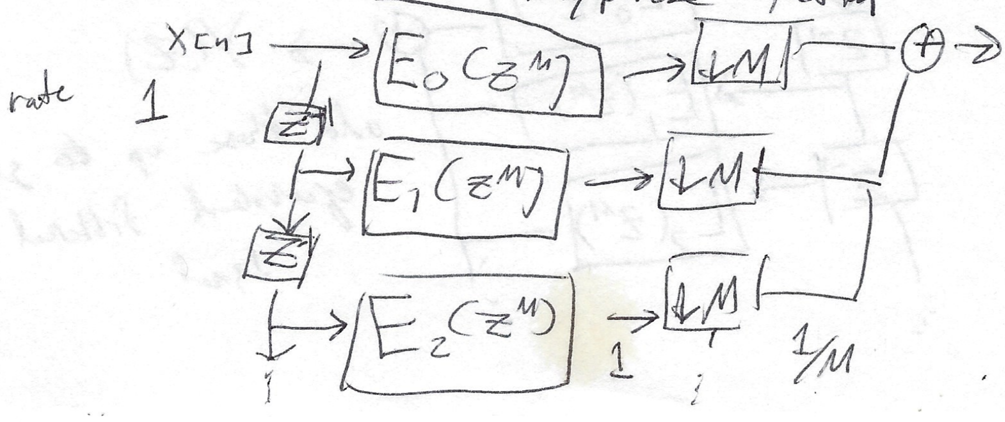
Suppose we want



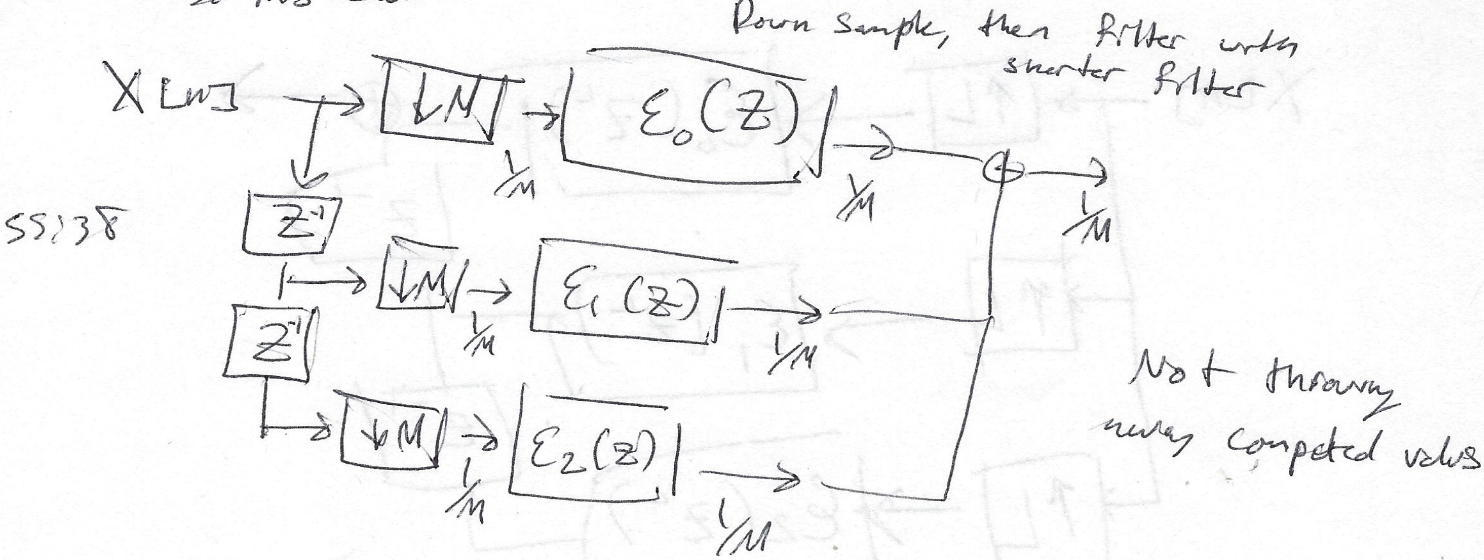
Kinda Dumb as we Keep only M th sample, why bother? don't generate stuff not used

IN This form we're Generating and Discarding $\frac{M-1}{M}$ Values

→ write this in Polyphase Form



Find A More Efficient Equivalent System
So instead



e.g. if original h[n] was an N-TAP FIR Filter, each of $e_i[n]$ is an N/M TAP filter

ORIGINAL Realization would involve N multiplications/unit time

Polyphase realization $\frac{N}{M}$ multiplications / unit time

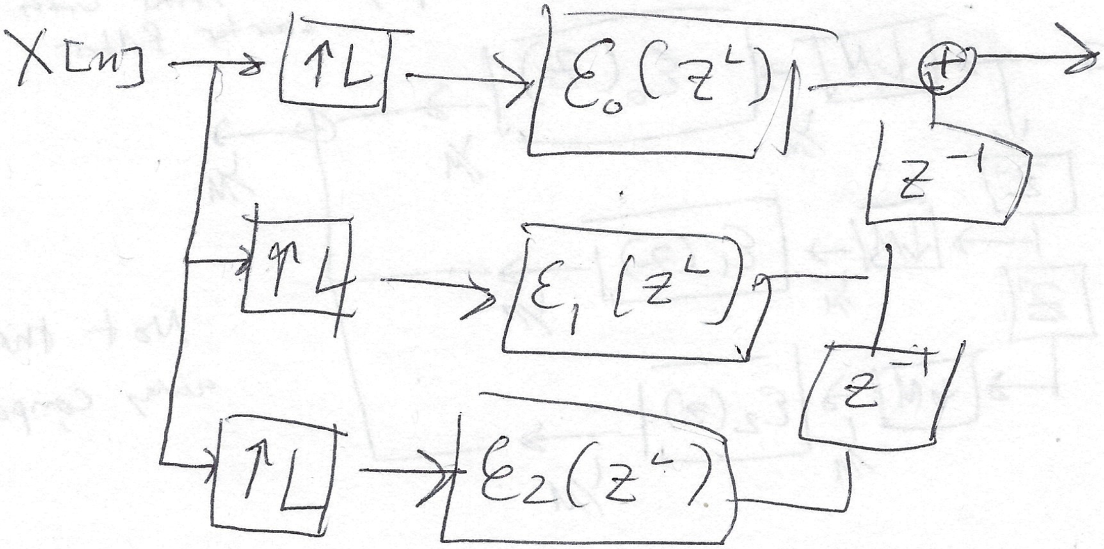
SF:22 Never filter anything not needed

Wasteful as well:

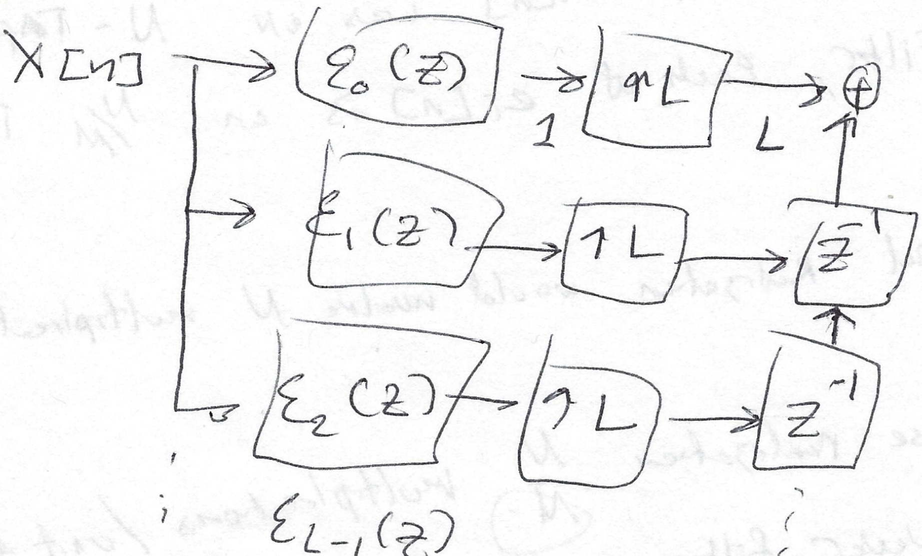
$\frac{L-1}{L}$ values $= 0$

we would be filtering mostly zeros

(14) Use Poly phase representation



Instead do



Here, No zero's are Explicitly filtere

→ efficient computations

Wavelets
Filter banks

1 example to make a long filter then make it into these three filters