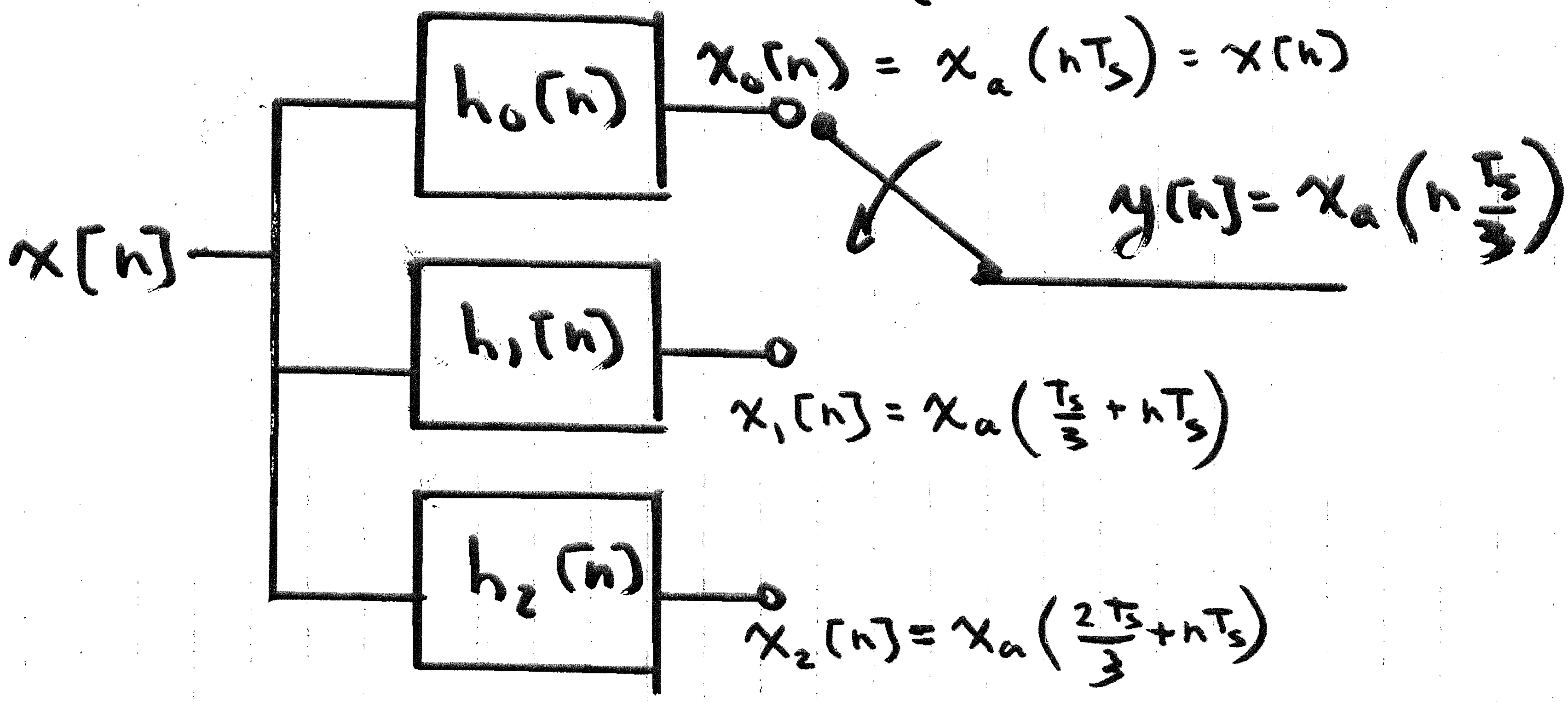
Recall: Single Signal 3x upsampling

$$h[n] = 3\,\frac{\sin\left(\frac{\pi}{3}n\right)}{\pi n}$$

$$h_0[n] = h[Ln]$$

$$h_1[n] = h[Ln+1]$$

$$h_2[n] = h[Ln+2]$$

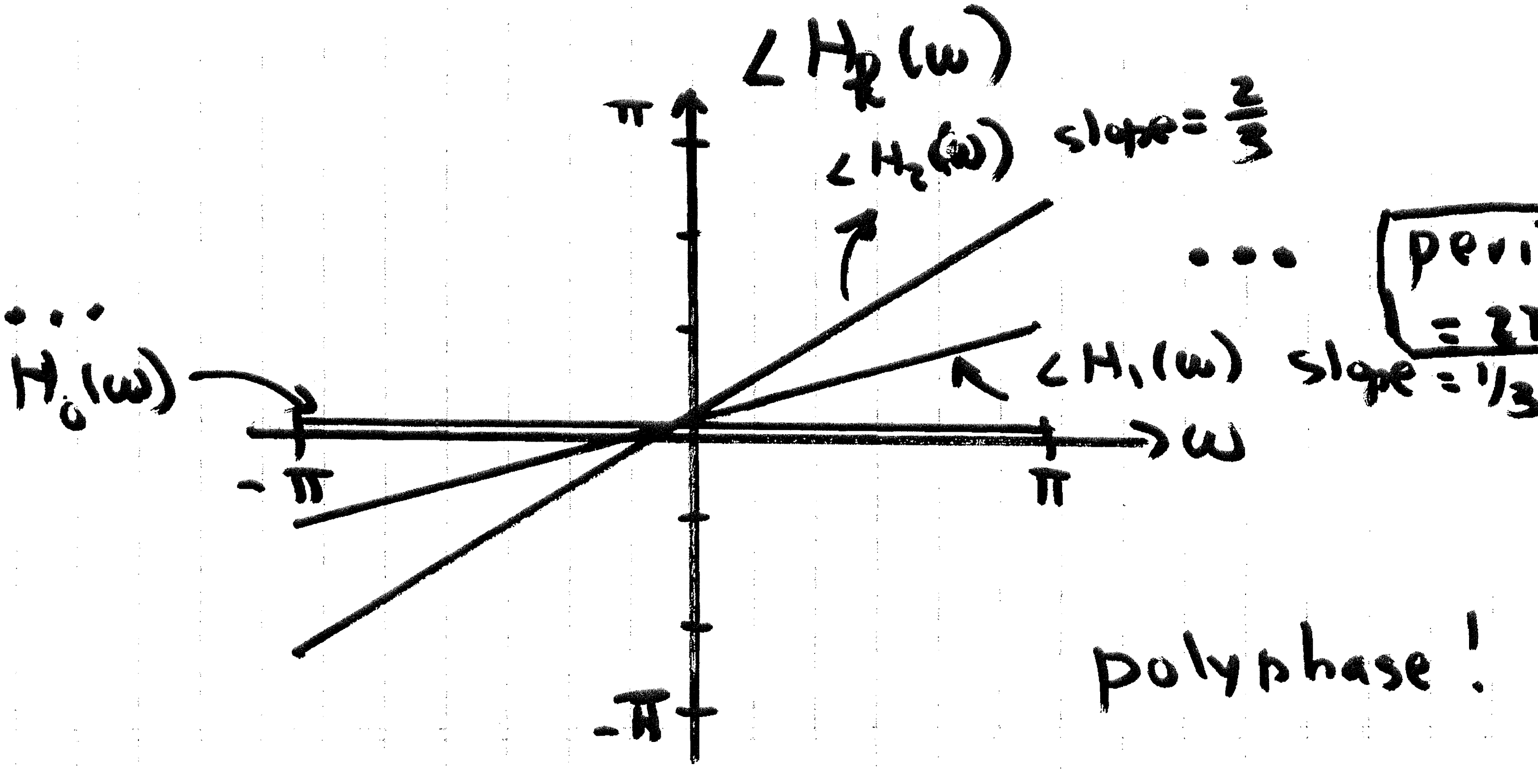


Recall: Single Signal $\Rightarrow$ 3x upsampling

for $|\omega| < \pi$: $H_0(\omega) = 1$

$H_1(\omega) = e^{j\frac{1}{3}\omega} \Rightarrow \angle H_1(\omega) = \frac{\omega}{3}$

$H_2(\omega) = e^{j\frac{2}{3}\omega} \Rightarrow \angle H_2(\omega) = \frac{2}{3}\omega$



$\boxed{\text{period } = 2\pi}$

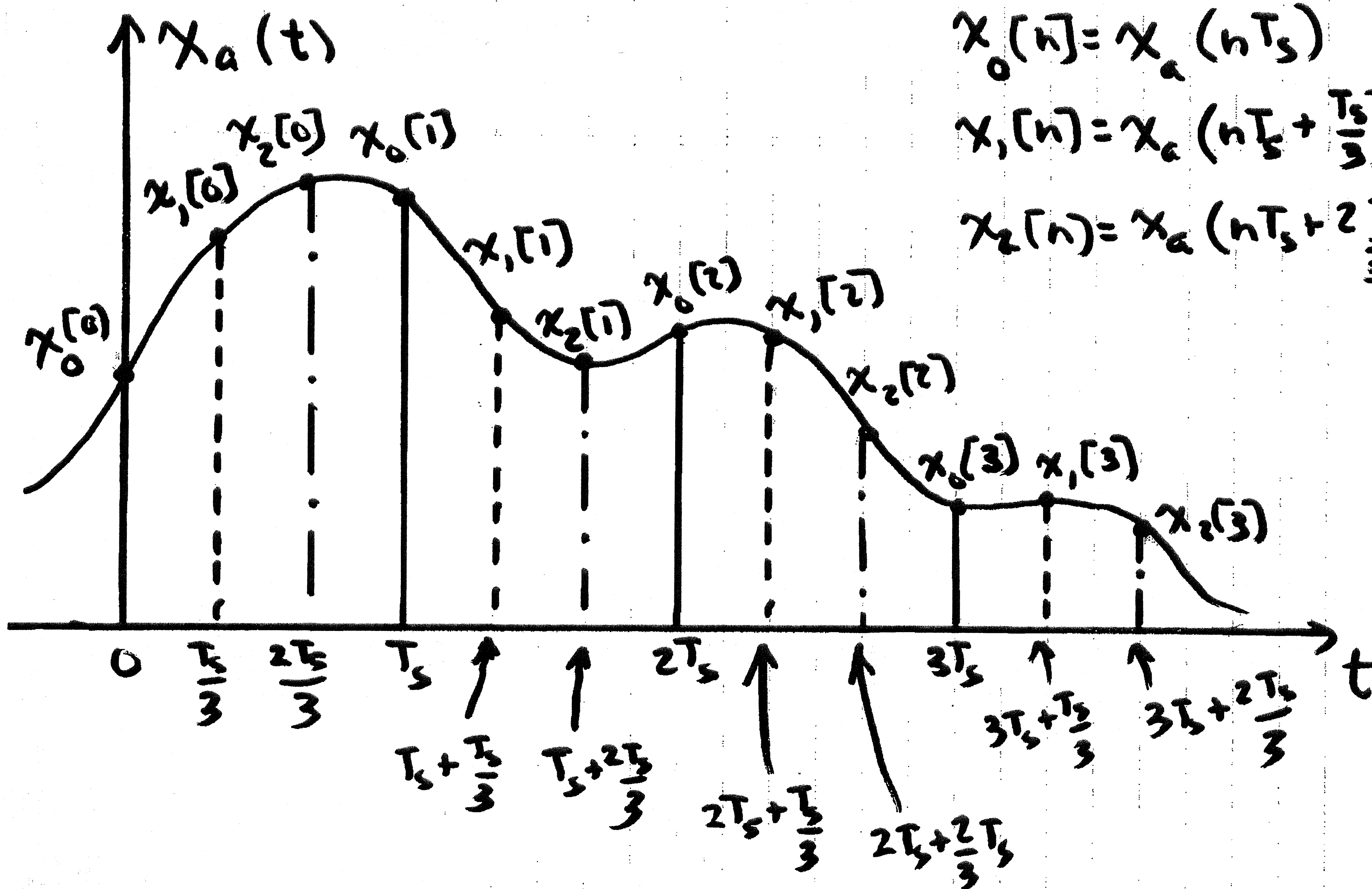
polyphase!

Single Signal: Upsampling 3X

$$x_0[n] = x_a(nT_s)$$

$$x_1[n] = x_a\left(nT_s + \frac{T_s}{3}\right)$$

$$x_2[n] = x_a\left(nT_s + 2\frac{T_s}{3}\right)$$



Straightforward Modulation After Digital Upsampling

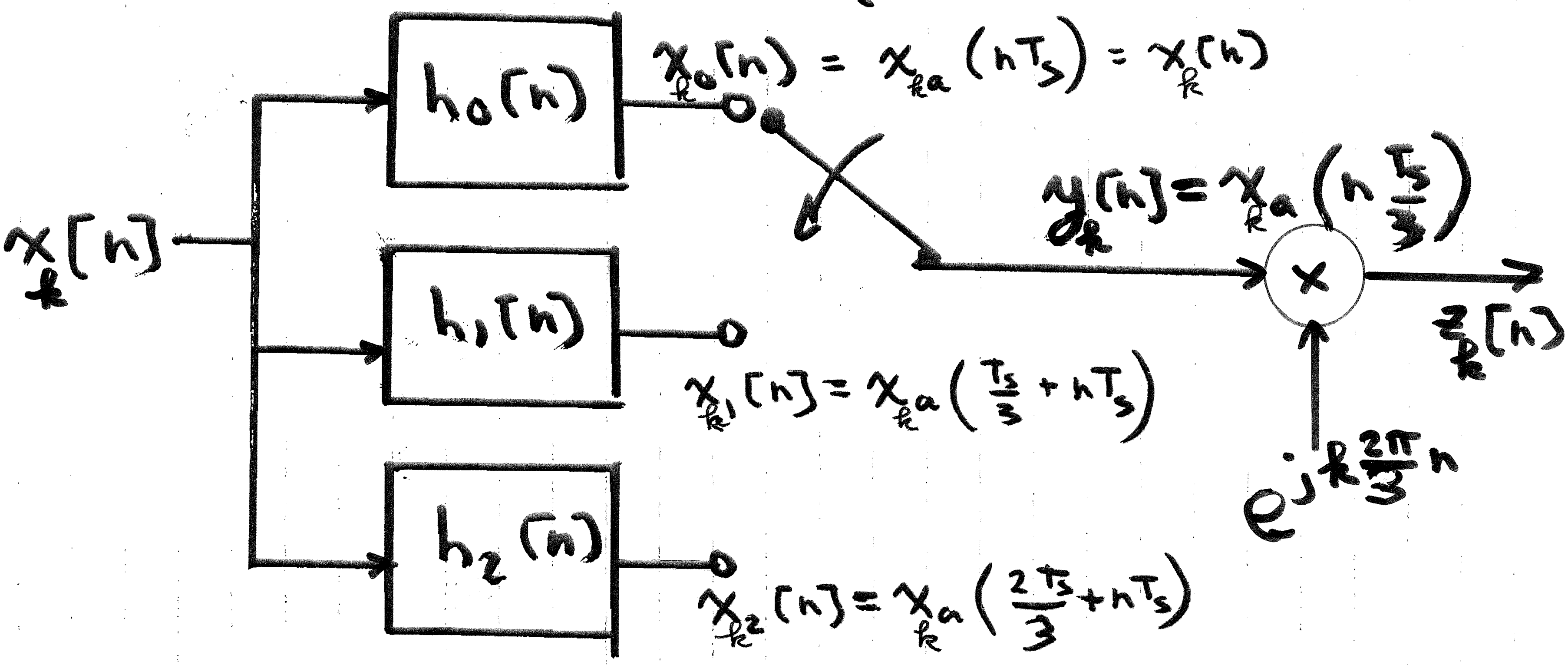
$$h[n] = \frac{3\sin\left(\frac{\pi}{3}n\right)}{\pi n}$$

$$h_0[n] = h[Ln]$$

$$h_1[n] = h[Ln+1]$$

$$h_2[n] = h[Ln+2]$$

$$L=3$$



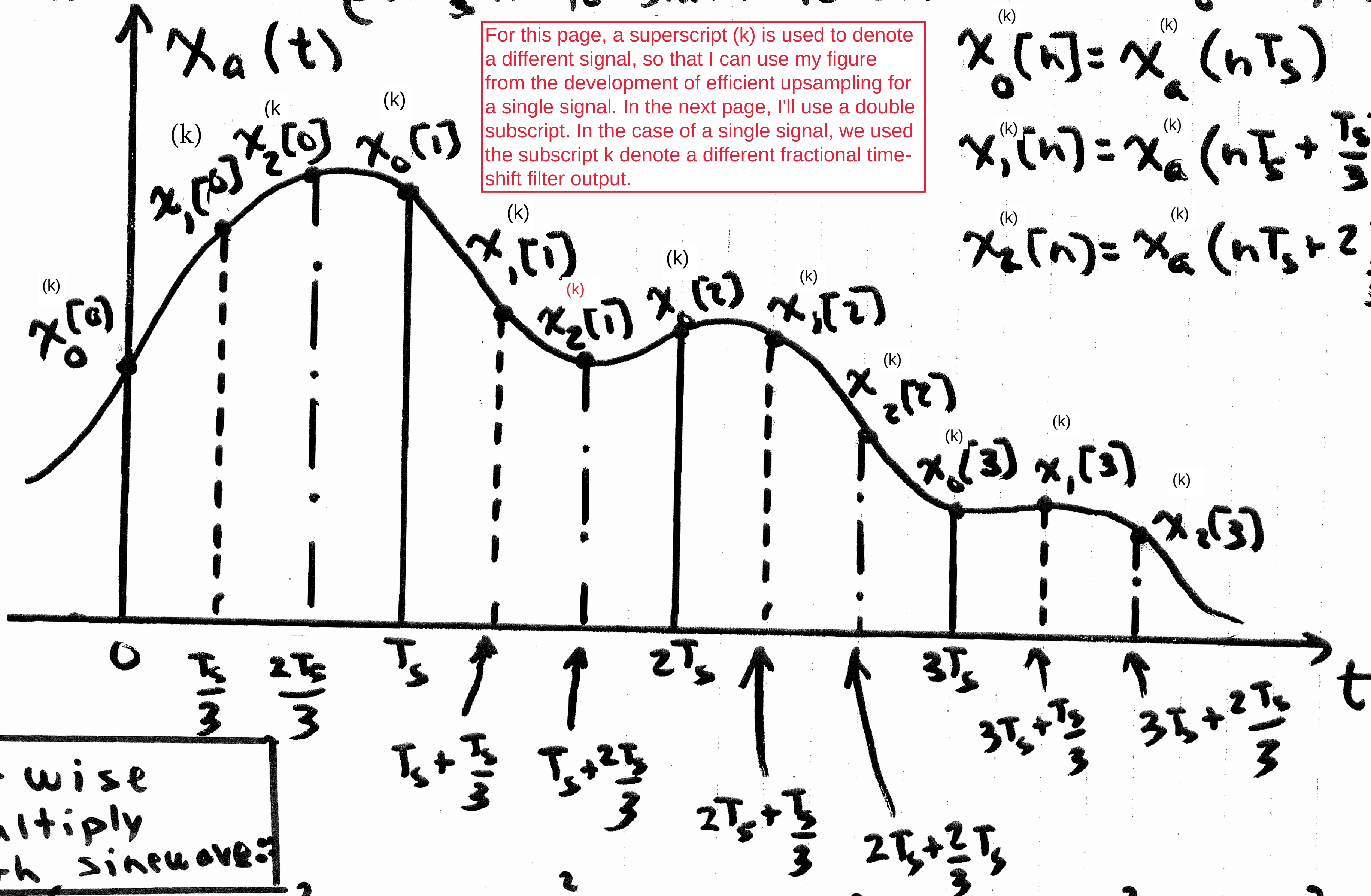
After 3x upsampling, pt-wise multiply by sinewave $e^{jk\frac{2\pi}{3}n}$ to shift to different frequency band

For this page, a superscript (k) is used to denote a different signal, so that I can use my figure from the development of efficient upsampling for a single signal. In the next page, I'll use a double subscript. In the case of a single signal, we used the subscript k denote a different fractional time-shift filter output.

$$x_0^{(k)}[n] = x_a^{(k)}(nT_s)$$

$$x_1^{(k)}[n] = x_a^{(k)}\left(nT_s + \frac{T_s}{3}\right)$$

$$x_2^{(k)}[n] = x_a^{(k)}\left(nT_s + 2\frac{T_s}{3}\right)$$



pt-wise multiply with sinewave:

$$.* \{ 1 \quad \gamma_k \quad \gamma_k^2 \quad 1 \quad \gamma_k \quad \gamma_k^2 \quad 1 \quad \gamma_k \quad \gamma_k^2 \quad 1 \quad \gamma_k \quad \gamma_k^2 \quad \ldots \}$$

where: $\gamma_k = e^{jk\frac{2\pi}{3}}$, $k = 0, 1, 2$

period of each sinewave = 3

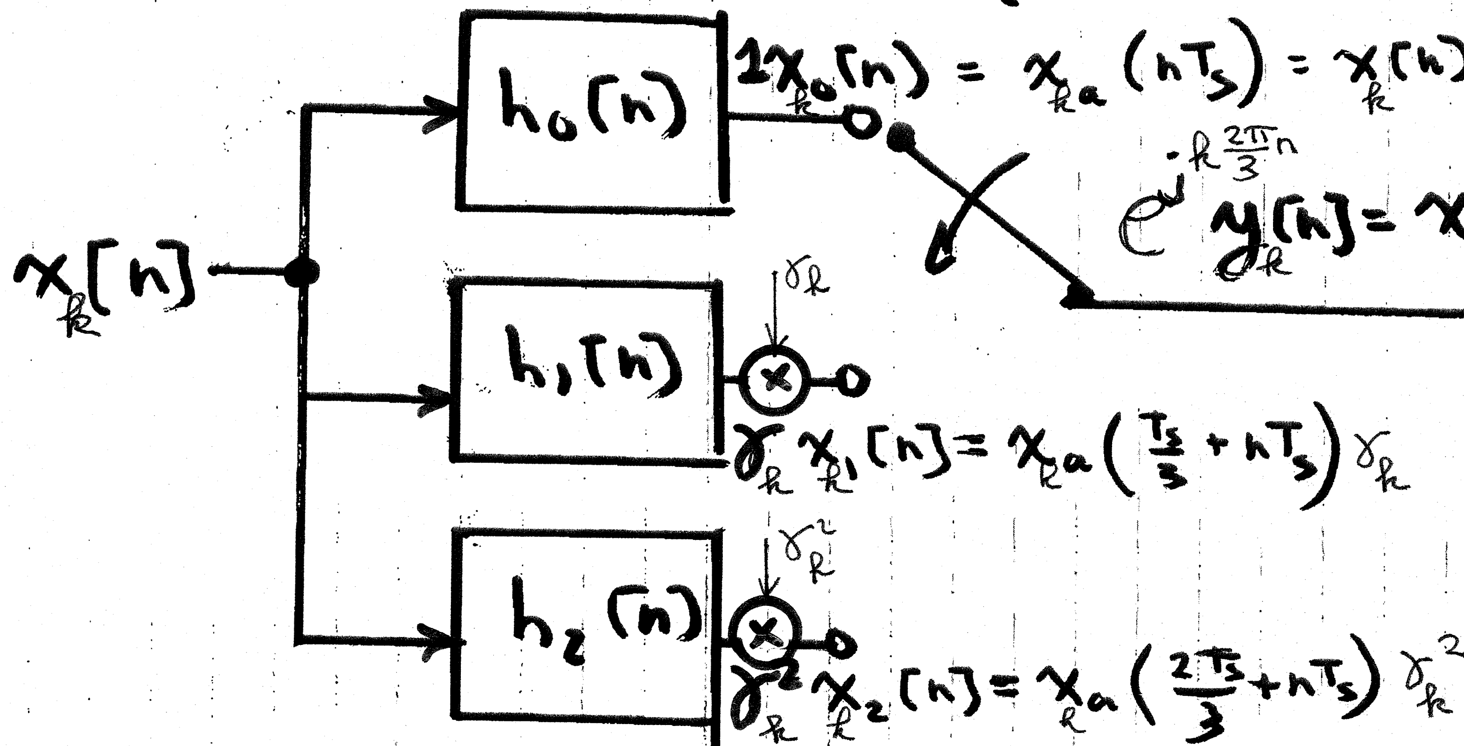
Post Up-Sampling Modulation Exploiting Sinewave Period 3

$$h[n] = 3\,\frac{\sin\left(\frac{\pi}{3}n\right)}{\pi n}$$

$$h_0[n] = h[Ln]$$

$$h_1[n] = h[Ln+1] \qquad L=3$$

$$h_2[n] = h[Ln+2]$$



$$1\,x_{k0}[n] = x_{ka}(nT_s) = x_k[n]$$

$$e^{jk\frac{2\pi}{3}n}\, y_k[n] = x_{ak}\left(n\frac{T_s}{3}\right) e^{jk\frac{2\pi}{3}n}$$

$$\gamma_k\, x_{k1}[n] = x_{ka}\left(\frac{T_s}{3} + nT_s\right)\gamma_k$$

$$\gamma_k^2\, x_{k2}[n] = x_{ka}\left(\frac{2T_s}{3} + nT_s\right)\gamma_k^2$$

where: $\gamma_k = e^{jk\frac{2\pi}{3}}$

$k = 0, 1, 2$

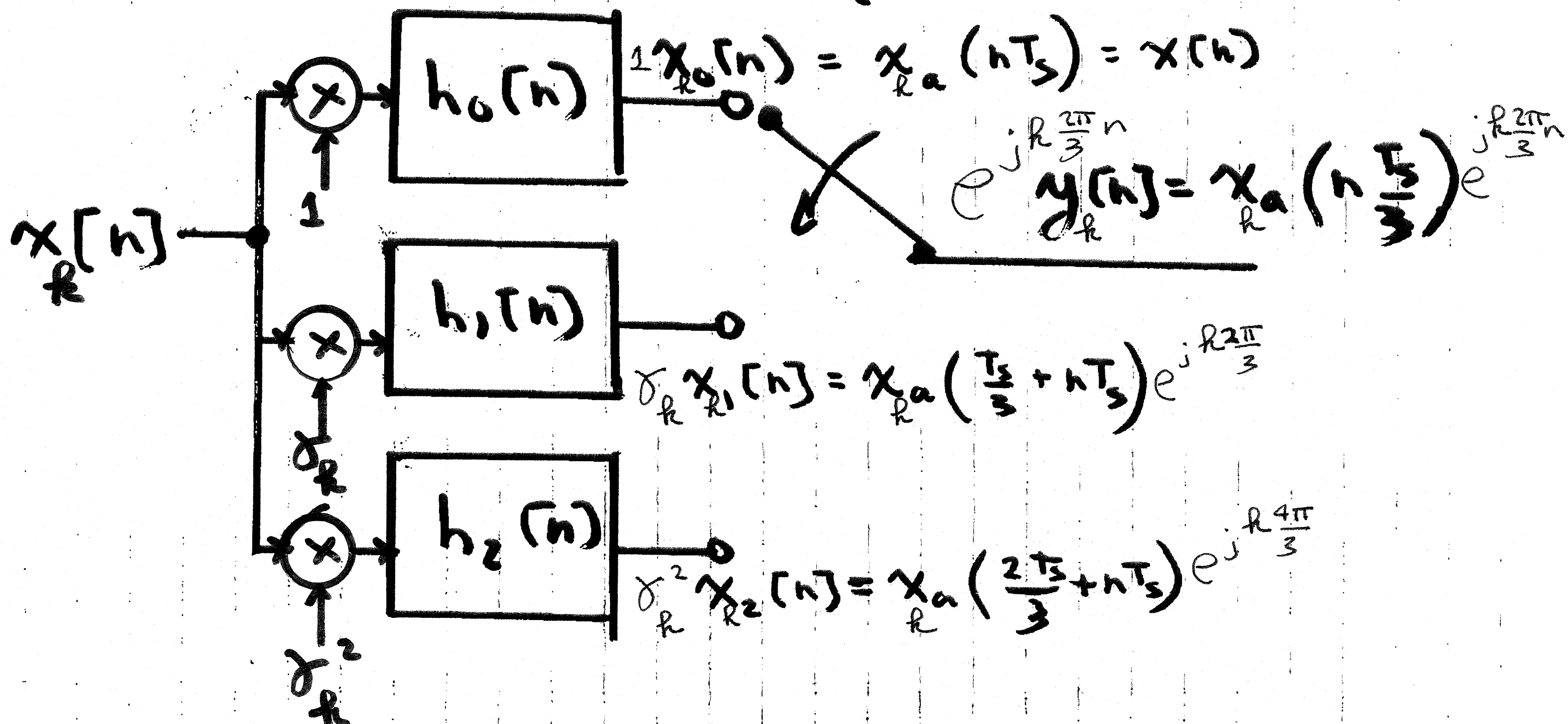
NEXT: Exploit Linearity to Move Scaling Before Filter

$$h[n] = 3\,\frac{\sin\left(\frac{\pi}{3}n\right)}{\pi n}$$

$$h_0[n] = h[Ln]$$

$$h_1[n] = h[Ln+1]$$

$$h_2[n] = h[Ln+2]$$



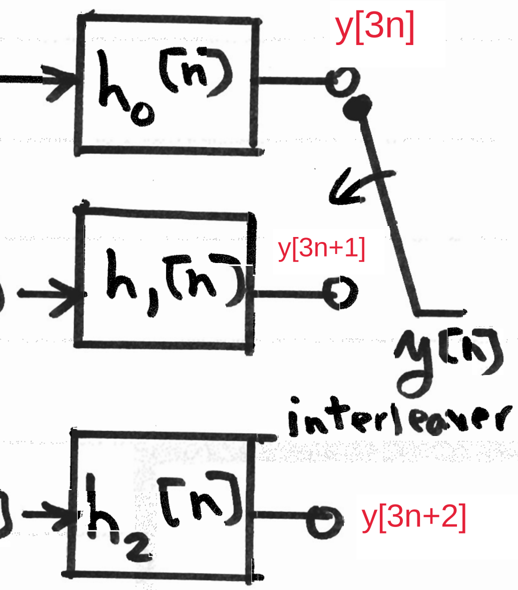
where: $\gamma_k = e^{jk\frac{2\pi}{3}}$ $\quad k = 0, 1, 2$

- **And recall**: we ultimately sum everything together since the signals are in different frequency bands

- **THUS**:

Below, x_0 [n] , x_1 [n] and x_2 [n] represent 3 DIFFERENT signals, each sampled above Nyquist (but presumably not much above the Nyquist Rate.) The subscript k also represents a different subband since each signal is ultimately upsampled and placed in a different subband.

	$k=0$		$k=1$		$k=2$
$\ell=0:$	$x_0[n]$	$+$	$x_1[n]$	$+$	$x_2[n]$
$\ell=1:$	$x_0[n]$	$+$	$e^{j\frac{2\pi}{3}(1)(1)}\,x_1[n]$	$+$	$e^{j\frac{2\pi}{3}(2)(1)}\,x_2[n]$
$\ell=2:$	$x_0[n]$	$+$	$e^{j\frac{2\pi}{3}(1)(2)}\,x_1[n]$	$+$	$e^{j\frac{2\pi}{3}(2)(2)}\,x_2[n]$

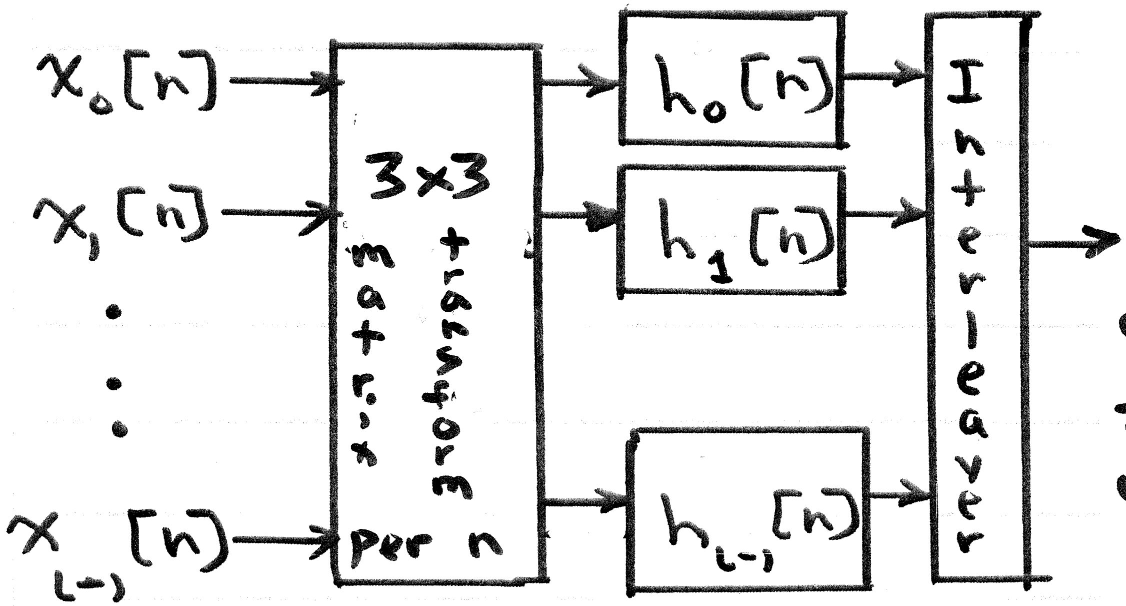


- again: k = signal index = subband index, $k=0,1,2$

ℓ = polyphase component index, $\ell = 0,1,2$

$\Rightarrow$ see matlab example Multiplex3Sigs.m

- The scalars on the polyphase components of the LP filter turn out to comprise 3×3 matrix

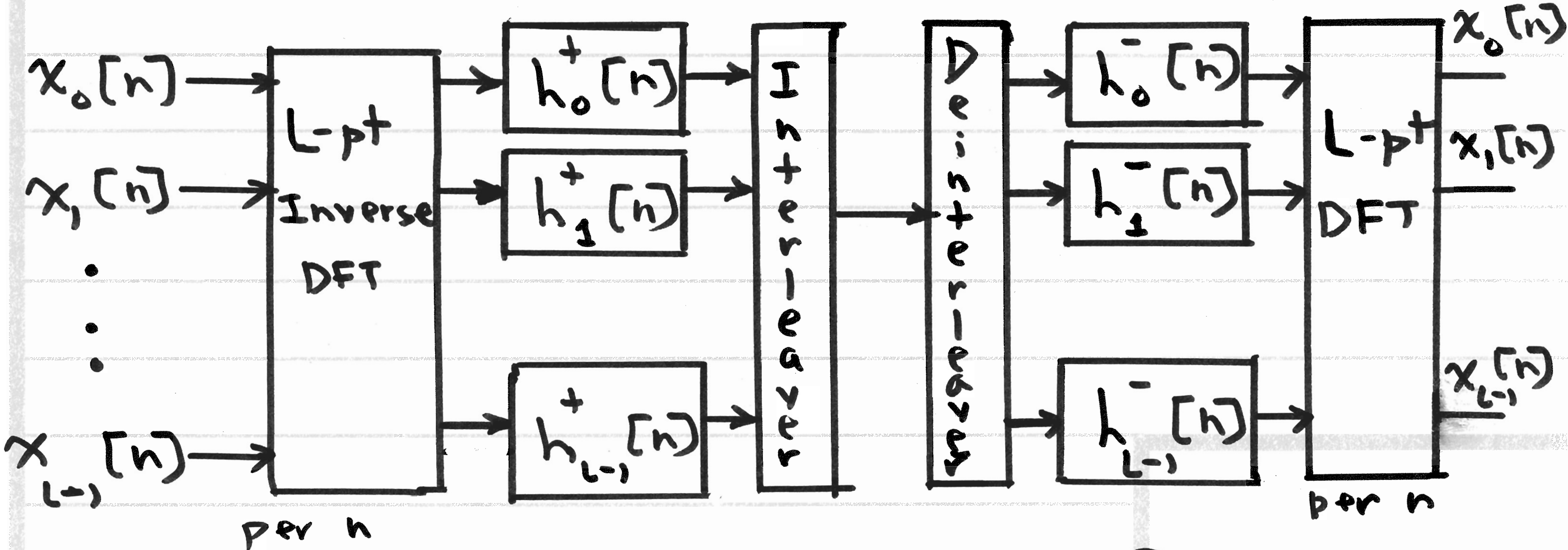


sum of L signals, each upsampled × L and placed in non-overlapping adjacent frequency bands of width $\frac{2\pi}{L}$

- In ideal case where $h_{LP}[n] = L\,\frac{\sin(\frac{\pi}{L}n)}{\pi n}$

$$H_\ell(w) = e^{j\frac{\ell}{L}w} \quad \text{for } -\pi < w < \pi$$

- The scalars on the polyphase components of the LP filter turn out to comprise either an L-pt DFT or an L-pt Inverse DFT

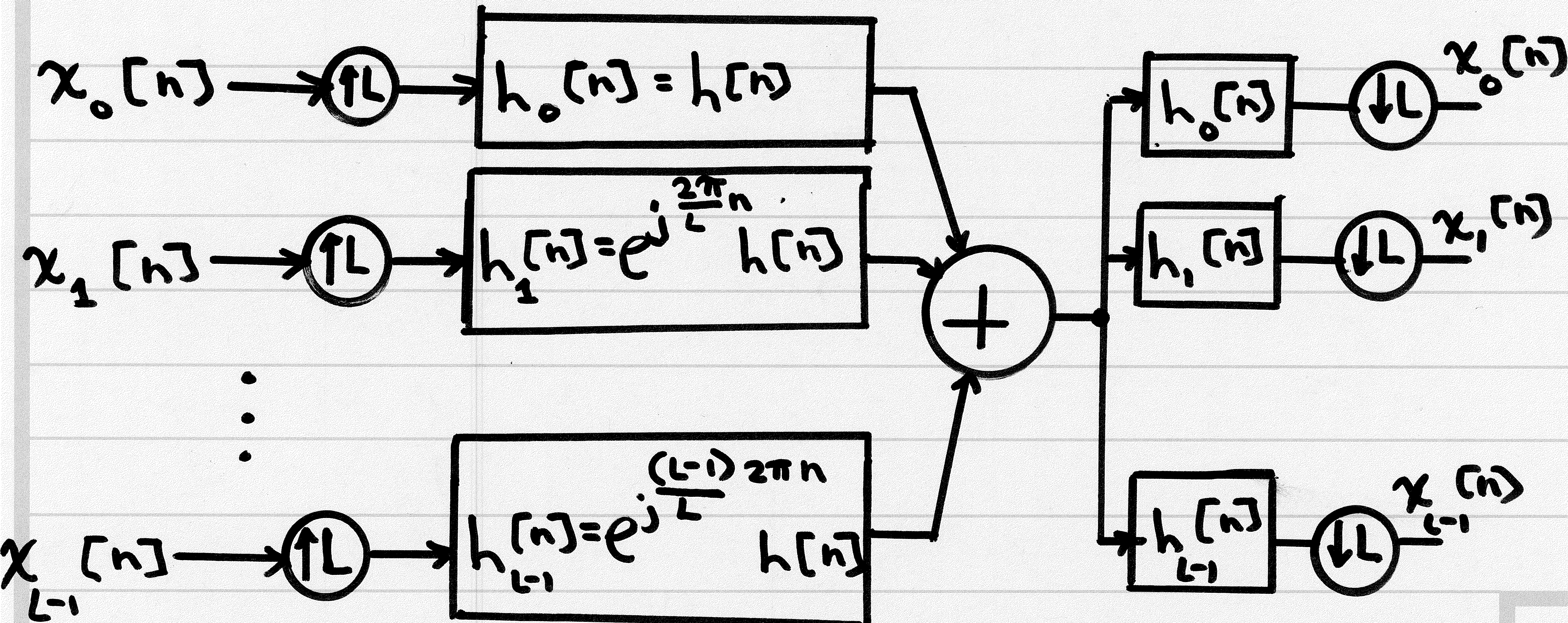


- In ideal case where $h_{LP}[n] = L\dfrac{\sin(\frac{\pi}{L}n)}{\pi n}$

$$H_\ell^-(\omega) = e^{-j\frac{\ell}{L}\omega} \quad \text{for } -\pi < \omega < \pi$$

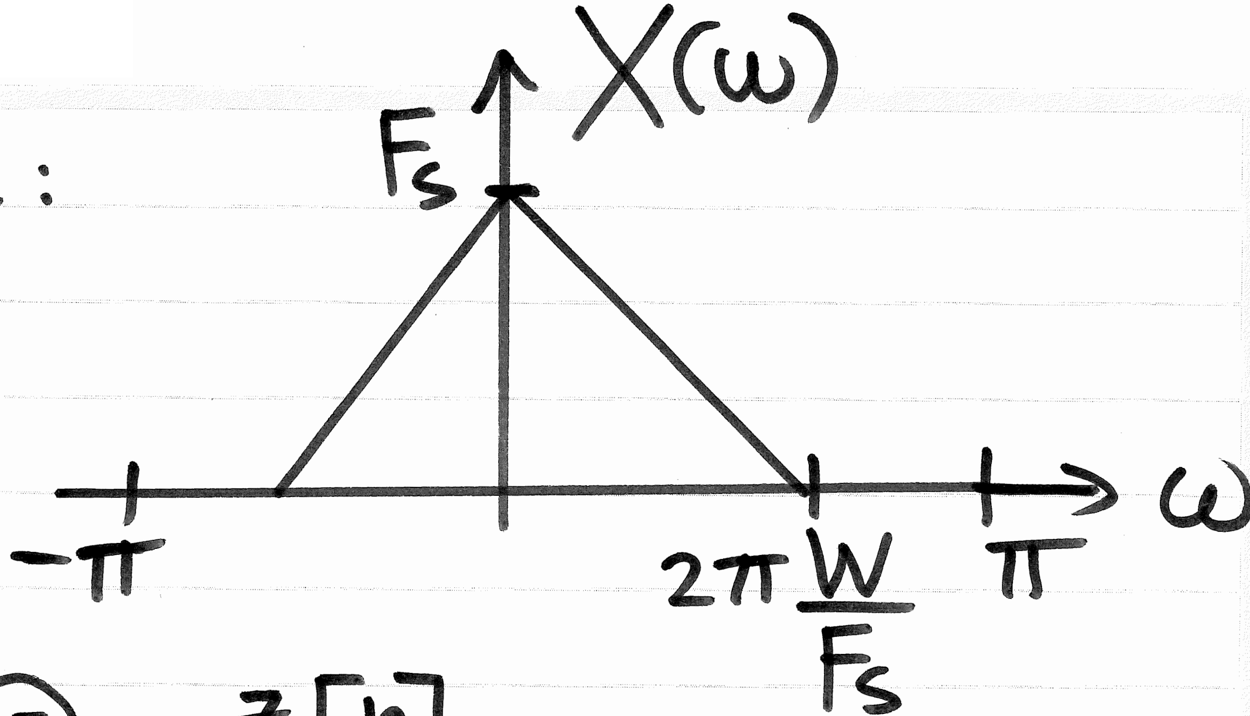
See Multiplex3SigsAlt.m

Previous page is efficient implementation of:



Recall:

• **Example:**



$x[n] \longrightarrow \uparrow 5 \longrightarrow z[n]$