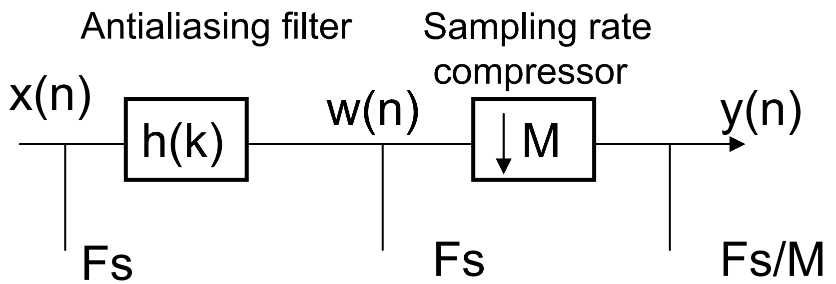
Multirate Signal Processing

Efficient technique for changing the sampling frequency of a digital signal.

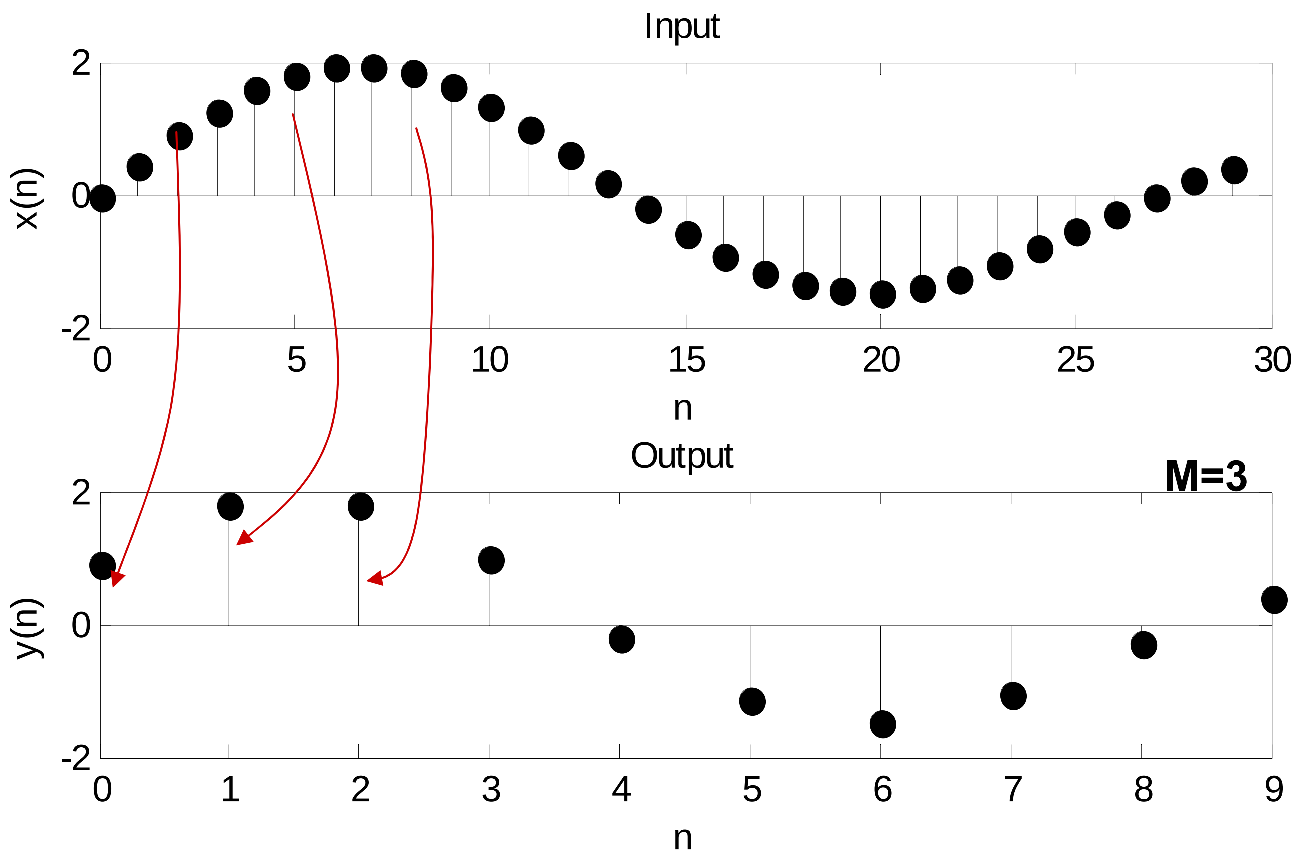
Sampling rate reduction-decimation by integre factor



$$y(n) = w(nM) = \sum_{k=-\infty}^{\infty} h(k)x(nM-k)$$

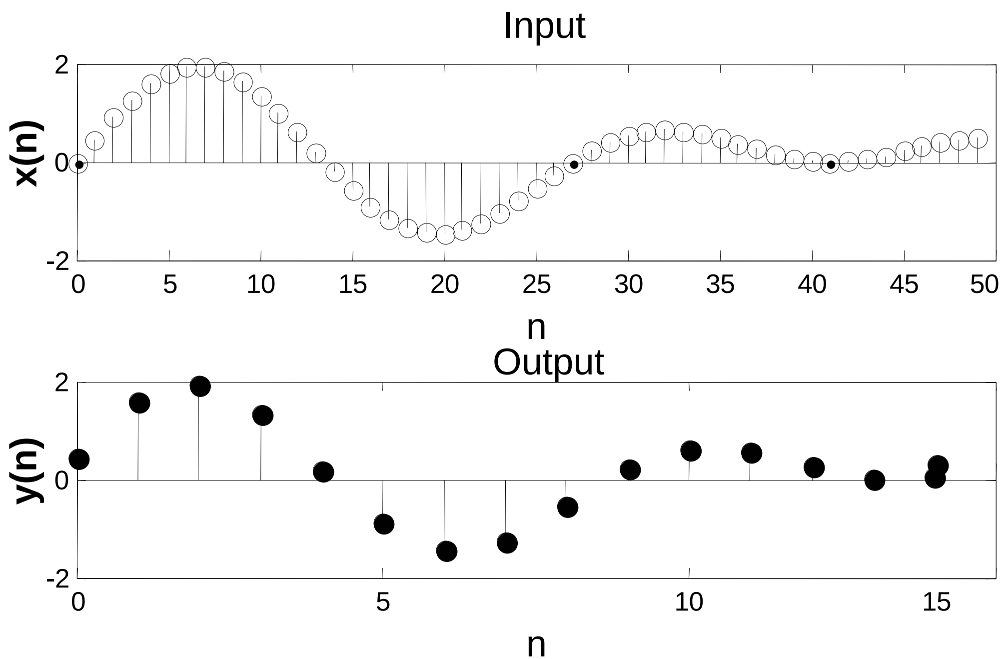
$$w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

Time Domain Illustration

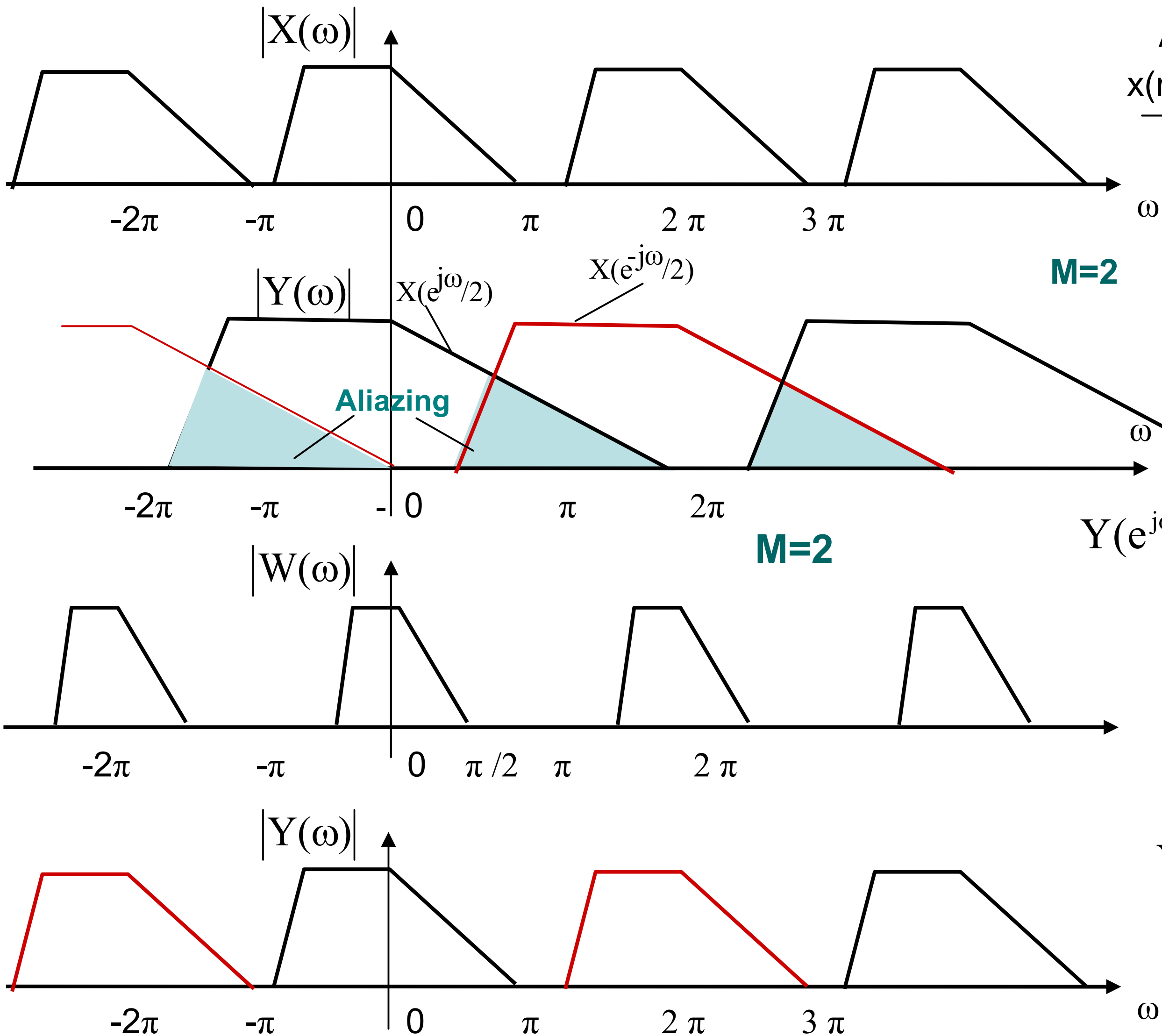


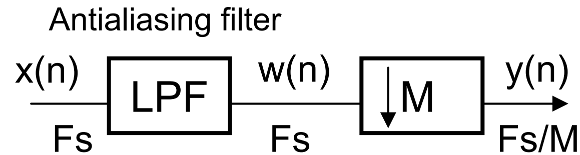
Decimation- Down-sampling

```
%Generate the input signal
N=50;M=3;
f1=0.043;
f2=0.031;
n=0:N-1;
x=sin(2*pi*f1*n)+sin(2*pi*f2*n);
%Generate the decimated signal
y=decimate(x,M);
% Plot trhe input and output signals
subplot(2,1,1)
stem(n,x(1:N)); title('Input')
xlabel('n'); ylabel('x(n)')
m=0:N/M-1
subplot(2,1,2)
stem(m,y(1:N/M));
title('Input');
xlabel('n'); ylabel('x(n)'
title('Output')
xlabel('n')
ylabel('y(n)')
```



Spectral Interpretation





$$W_M = e^{-j2\pi/M}$$

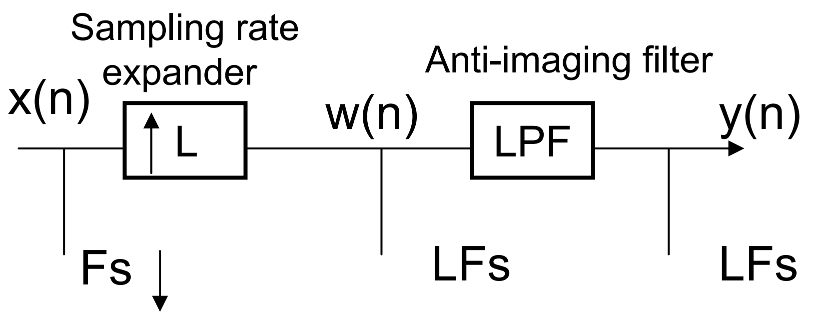
$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M}) W_M^{-k}$$

$$Y(e^{j\omega}) = \frac{1}{2}\left\{X(e^{j\omega/2}) + X(e^{-j\omega/2})\right\}$$

$$Y(-e^{j\omega}) = X(e^{j\frac{(\omega-2\pi)}{2}})$$

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j\frac{(\omega-2\pi k)}{M}})$$

Samping Rate Increase: Interpolation-Upsampling

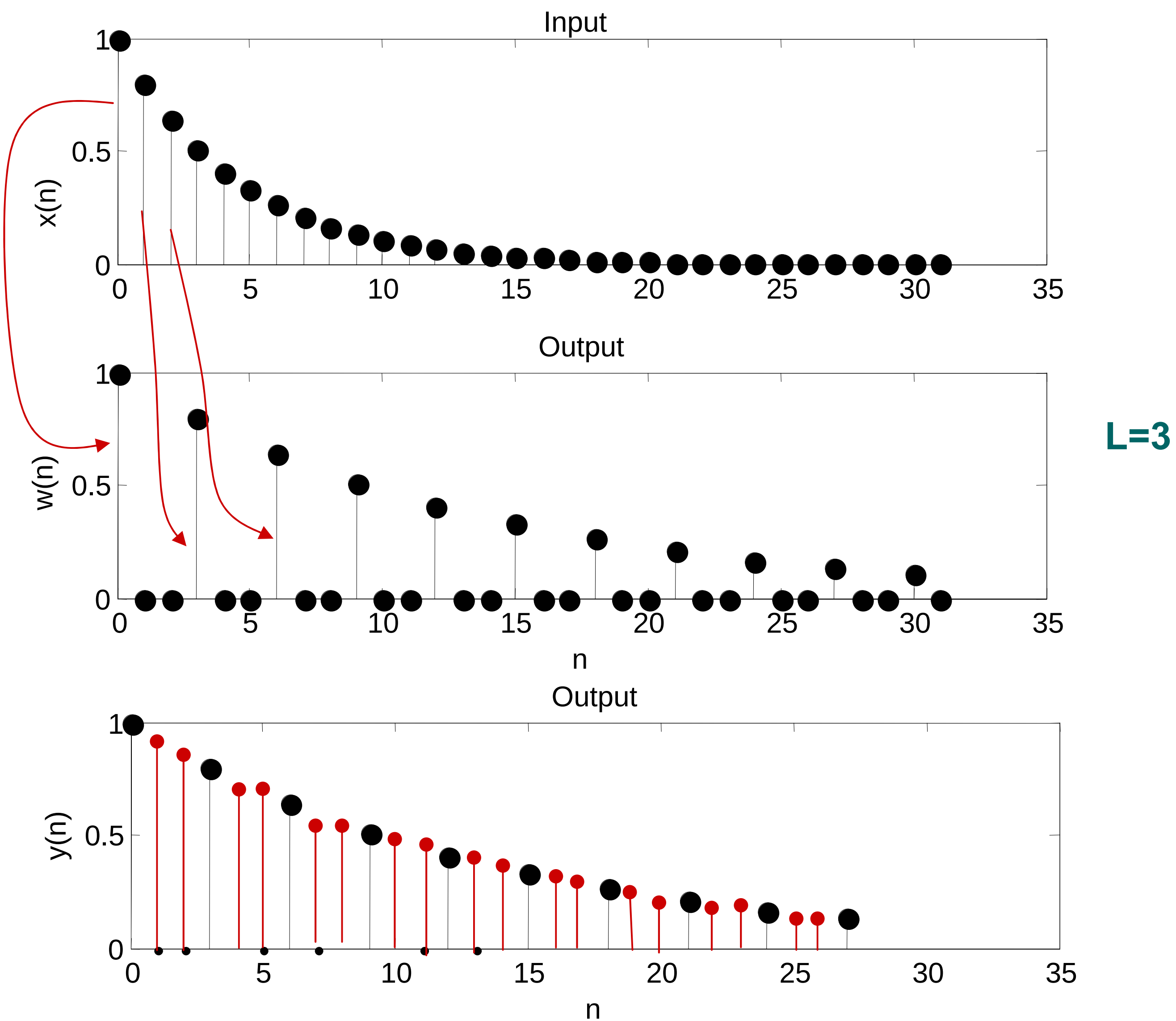


$$y(m) = \sum_{k=-\infty}^{\infty} h(k)w(m-k)$$

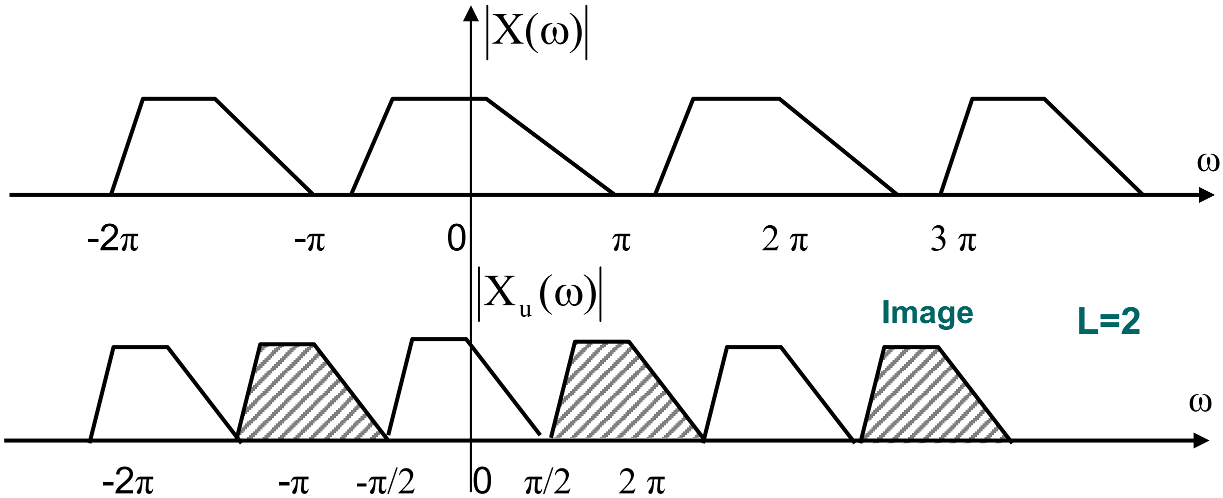
Where

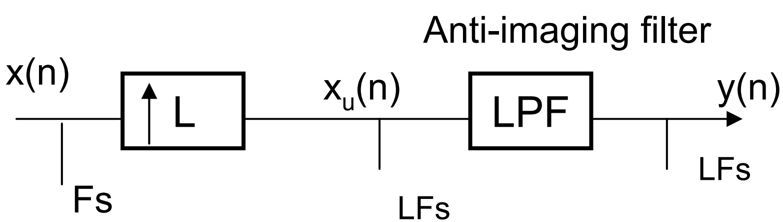
$$W(m) = \begin{cases} x(m/L) & m=0, \pm L, \pm 2L, \pm 3L, \ldots \\ 0) & \end{cases}$$

Time Domain Illustration



Spectral Interpretation





$$x_u(n) = \begin{cases} x(n/L) & n=0, \pm L, \pm 2L, \pm 3L, \ldots \\ 0 & \end{cases}$$

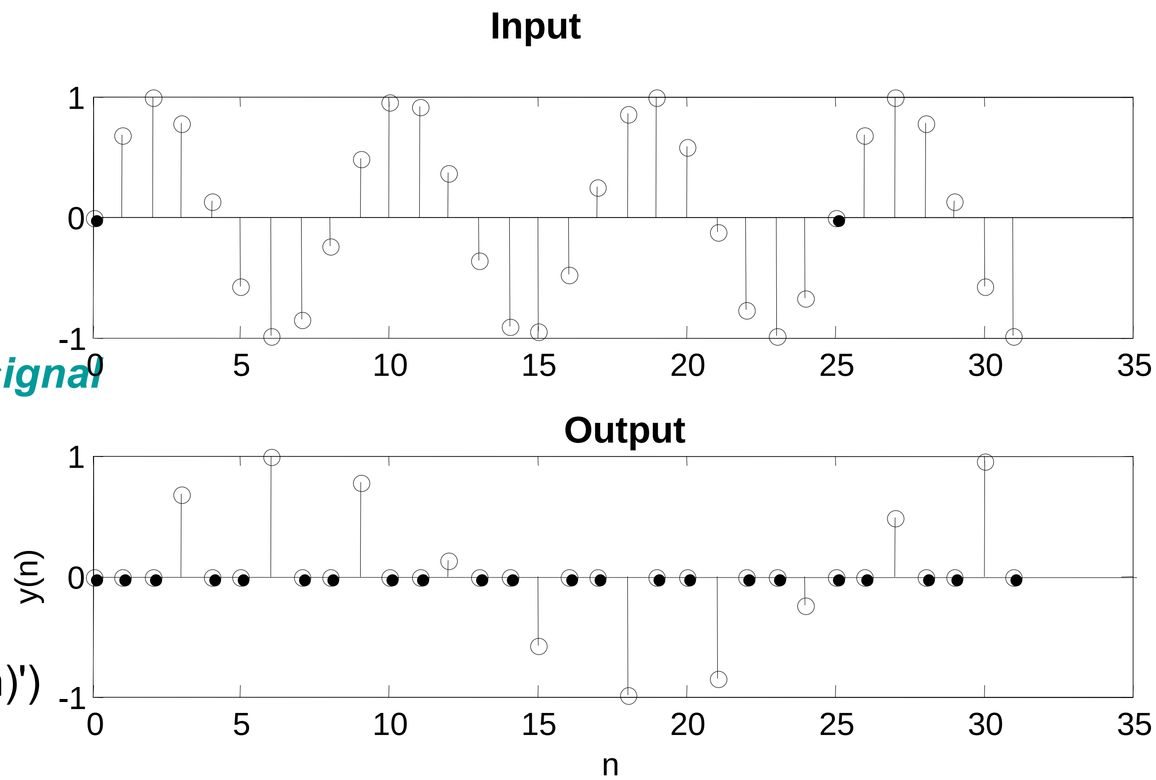
$$x_u(n) = \begin{cases} x(n/2) & n=0, \pm 2, \pm 4, \ldots \\ 0 & \end{cases}$$

$$X_u(z) = \sum_{n=-\infty}^{\infty} x_u(n) z^{-n} = \sum_{n=-\infty}^{\infty} x_u(n/2) z^{-n} = \sum_{n=-\infty}^{\infty} x_u(m) z^{-2m} = X(z^2)$$

$$|W(z)| = X(z^L)$$

Up-Sampling

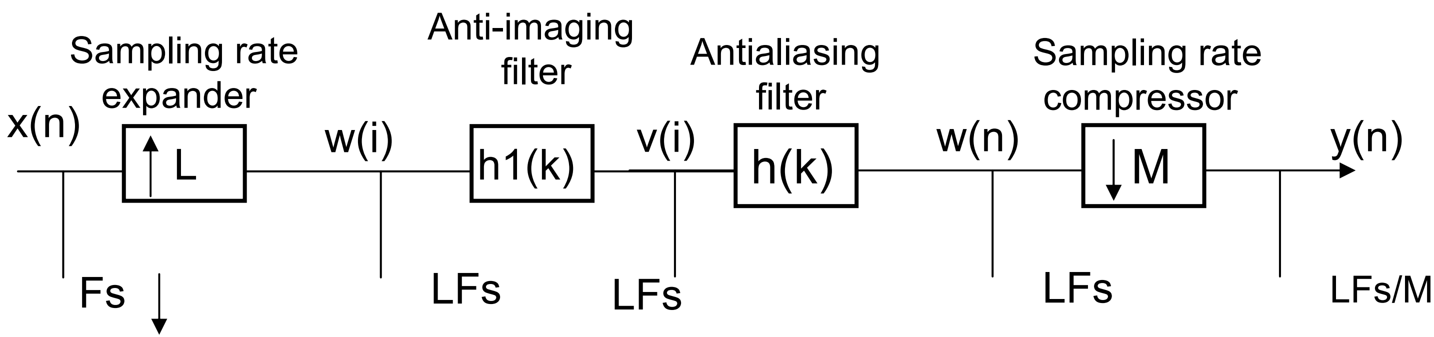
% Upsampling factor
L=3;N=32;
% Generation the input
f=0.12;
n=0:N-1;
x=sin(2*pi*f*n);
%Generation the up-sampled signal
y=zeros(1,L*length(x));
y([1:L:length(y)])=x;
% Plot the input and output
subplot(2,1,1)
stem(n,x)
title(‘Input'); xlabel('n'); ylabel(‘x(n)')
subplot(2,1,2)
stem(n,y(1:length(x)));
title('Output'); xlabel('n'); ylabel('y(n)')



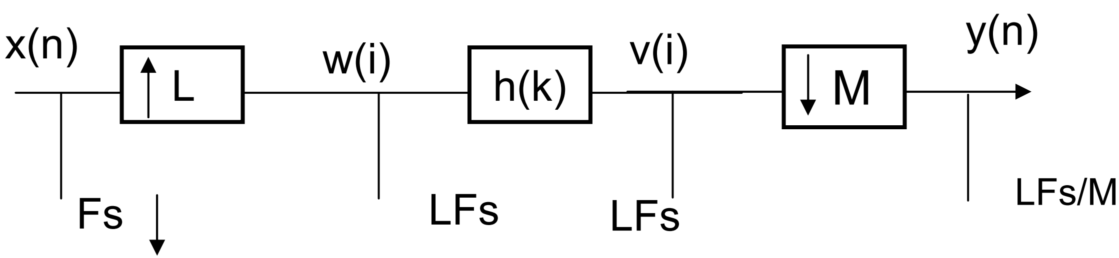
Sampling Rate Conversion by Non-integre Factor

Transfer data from CD at a rate 44.1 kHz to a Digital audio tape (DAT) at a trate 48 kHz
48/44.1is anon integrer.

If L=160 and M=147 we increase CD data rate to 7056(44.1x160) and then reduce it by (7056/147) to 48 kHz

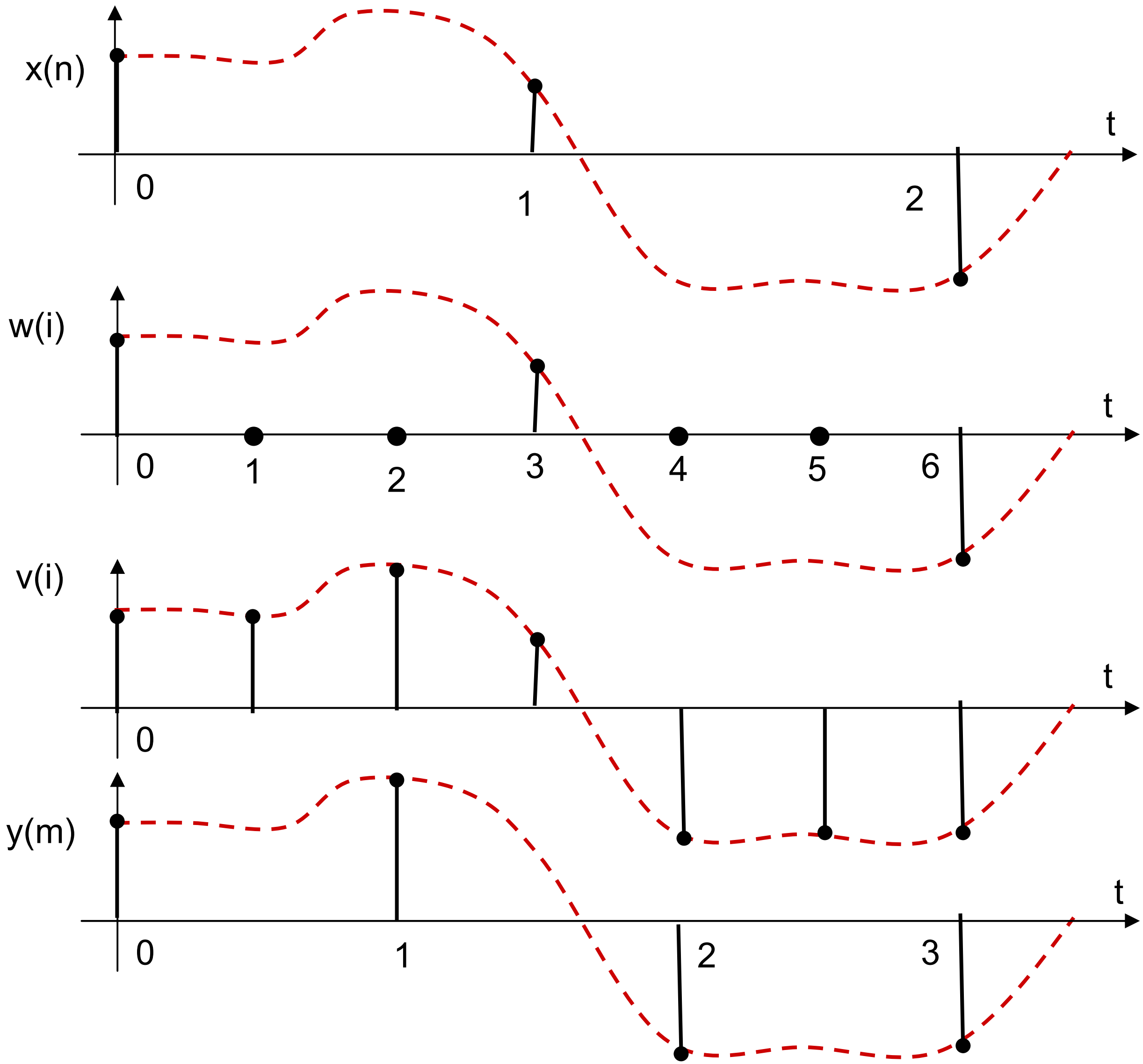


If M > L, resulting operation is a decimation
If M < L, resulting operation is a interpolation



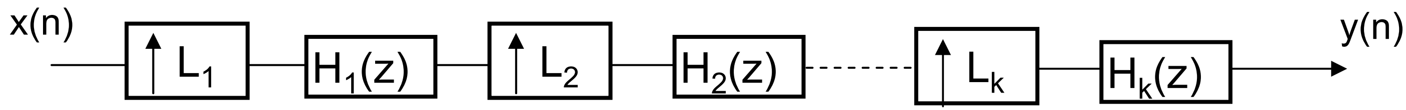
Interpolation by a Rational Factor

L=3; M=2; L/M=1.5

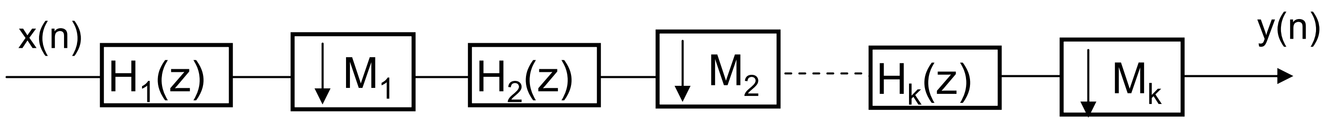


Multistage Design

$L = L_1 L_2 . . . L_k$

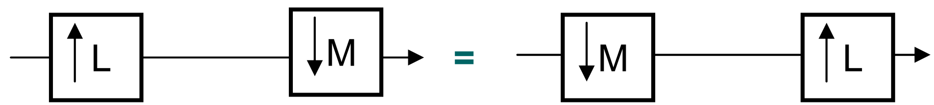


$M = M_1 M_2 . . . M_k$

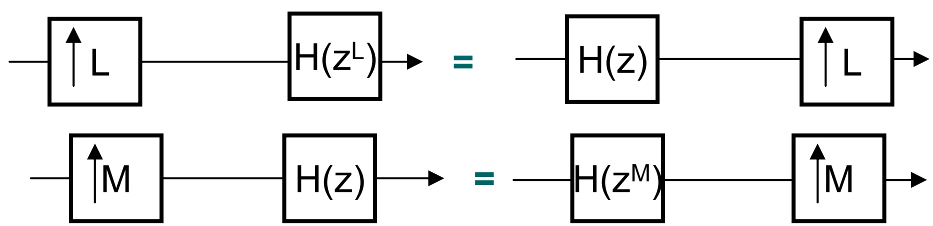


Cascade Equivalences

İs valid if M and L are relatively prime



Equivalences with filters



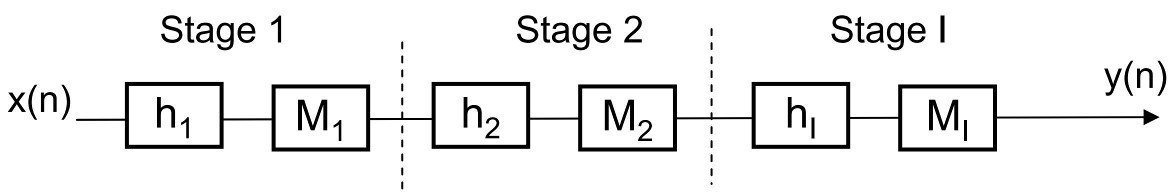
FIlter Requirements

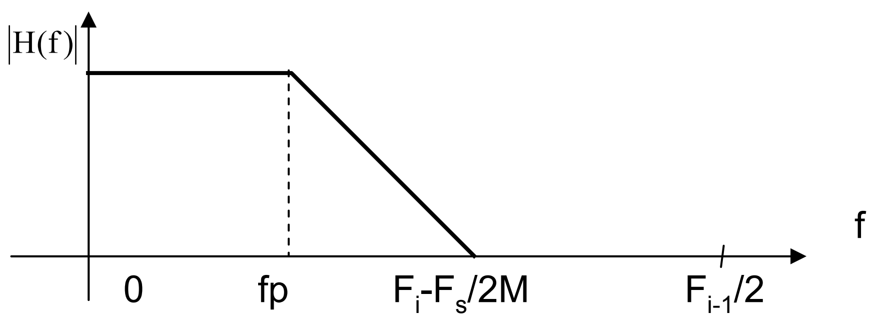
For multistage decimator:

Passband	$0 \leq f \geq fp$
Stopband	$(Fi\text{-}Fs/2M) \leq f \geq F_{i-1}/2$, $i=1, 2, ...I$
Pasband deviation	$\delta p/I$
Stopband deviation	δs
Filter length	$N \approx \frac{D\infty(\delta_p, \delta_s)}{\Delta f_i} - f(\delta_p, \delta_s)\Delta\Delta_i + 1$

The output sampling frequency for each stage i is given by $\mathbf{F_i=(F_i-1)/M_i}$

$\mathbf{F_i}$, $\mathbf{N_i}$ **and** $\mathbf{\Delta f_i}$ are , the output sampling frequency, length and transition width for i^{th} stage





Optimum Number of Stages

1. Specify the overall filter parameters
2. For each value of I obtain all possible set of integrer decimation factors of M
3. For each set of decimation factor determinethe MPS and TSR
4. Select the decimation factor giving minimum MPS and TSR

$$\mathrm{MPS} = \sum_{i=1}^{I} N_i F_i \qquad \mathrm{TSR} = \sum_{i=1}^{I} N_i$$

For optimum MPS and TSR the optimum decimation factor satisfy: $M_1 > M_2 > M3 \ldots\ldots M_I$

For I=2 (two stages)

$$M_{1opt} = \frac{2M}{2 - \Delta f + (2M\Delta f)^{1/2}}$$

$$M_{2opt} = \frac{M}{M_{1opt}}$$

Design of Practical Multirate System

1. Specify the overall AAF and AIF requirement
2. Determine the optimum number of stages
3. Determine the decimationand or interpolation factors foreach stage
4. Design an appropriate filter for each stage

1. Filter specification

1.1 Antialiasing filter AAF for decimation

Passband	0 ≤ f ≥fp
Stopband	Fs/2M ≤ f ≥Fs/2
Pasband deviation	δp
Stopband deviation	δs

Where fp ishighest frequency of original signal

1.2 Anti-imaging filter AIF for interpolation

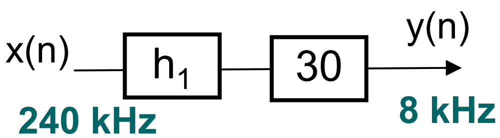
Passband	0 ≤ f ≥fp
Stopband	Fs/2 ≤ f ≥LFs/2
Pasband deviation	δp
Stopband deviation	δs

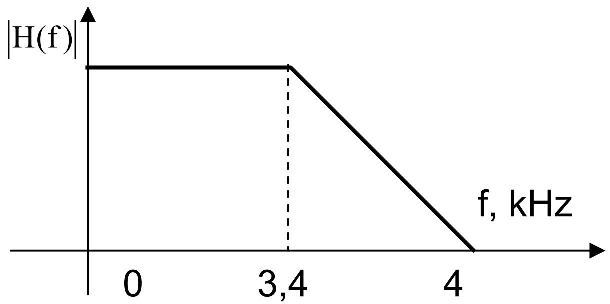
Problem

Design a 2-stage decimator that downsamples audio signal a factor 30 and satisfy the following specifications:

Input sampling frequency	Fs=240 kHz
Highest frequency of original data	3.4 kHz
Pasband deviation	$\delta p=0.05$
Stopband deviation	$\delta s=0.01$
Filter length	$N=\frac{-10\log(\delta_p\delta_s)-13}{14.6\Delta f}+1$

1. One-stage realization

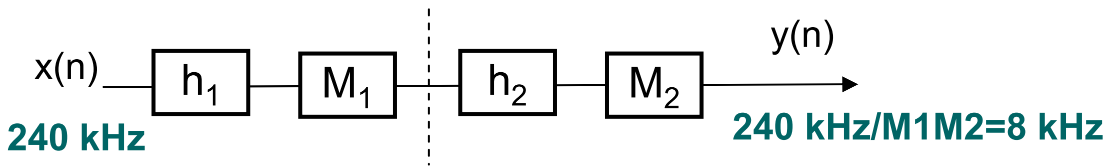




Fs=240; M=8;Fi=8;fp=3.4 kHz; fs=8-240/2x30=4 kHz

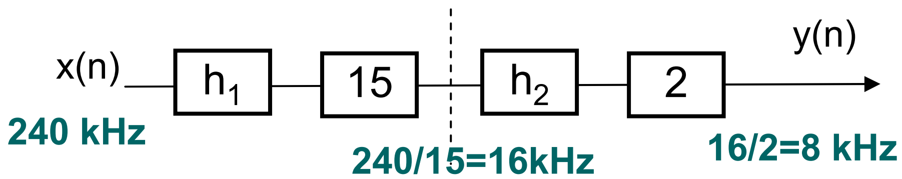
Δf=(4-3.4)/240 =2.5x10-3; N=549; MPS=N*Fi=1368x10^{3}; TSR=N=549

2. Two-stages realization



Integre factors that allow to ge decimation factor 30 are: **15x2; 10x3; 6x5**

1. M1=15; M2=2

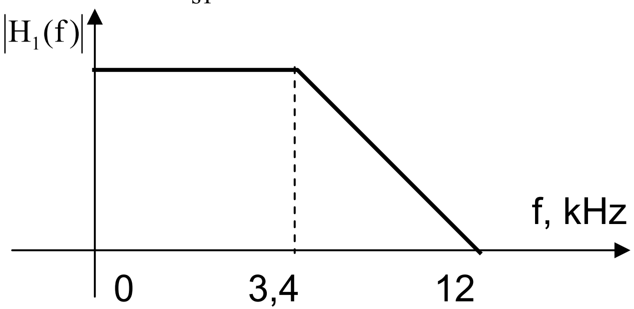


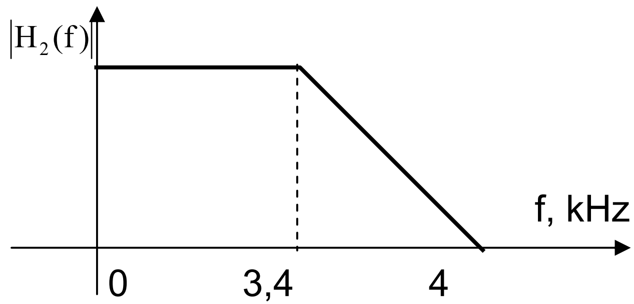
For h1 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=16-240/2x30=12 kHz**

$$\Delta f = \frac{f_{s1} - fp}{Fs} = \frac{12 - 3.4}{240} = 0.025; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s1} = 0.01; \; N1 = 45$$

For h2 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=8-240/2x30=4 kHz**

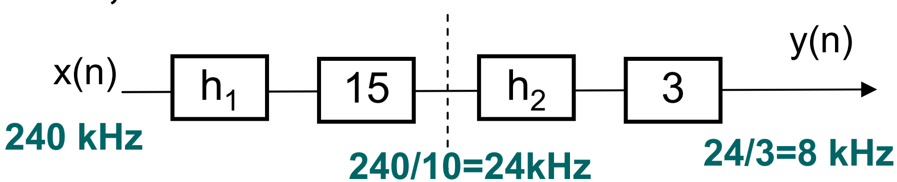
$$\Delta f = \frac{f_{s2} - f_p}{F_{s1}} = 0.0375; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s2} = 0.01; \quad N1 = 43$$





MPS=(45x16+43x8)10³=1064x10³ ; TSR=N1+N2=45+43=88

2. M1=10; M2=3

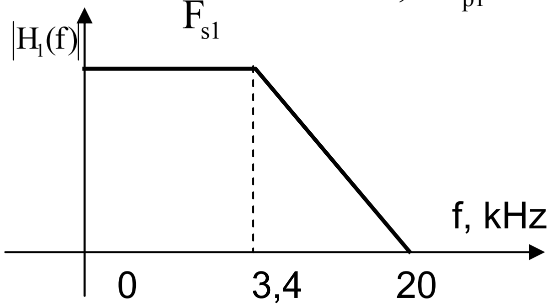


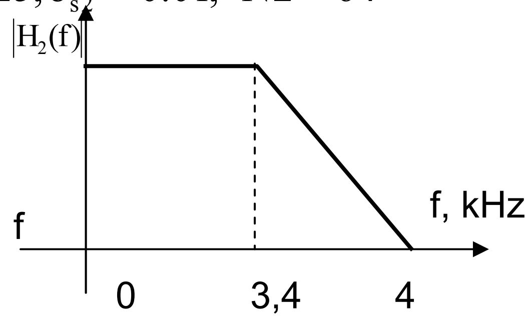
For h1 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=24-240/2x30=20 kHz**

$$\Delta f = \frac{f_{s1} - fp}{Fs} = \frac{20 - 3.4}{240} = 0.0691; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s1} = 0.01; \quad N1 \approx 24$$

For h2 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=8-4=4 kHz**

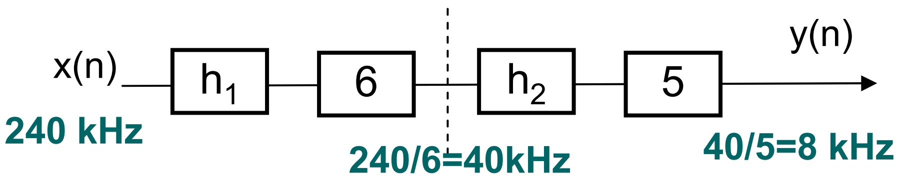
$$\Delta f = \frac{f_{s2} - f_p}{F_{s1}} = 0.025; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s2} = 0.01; \quad N2 = 64$$





MPS=(24x24+64x8)10^3=1088x10^3 ; TSR=N1+N2=64+24=88

3. M1=6; M2=5

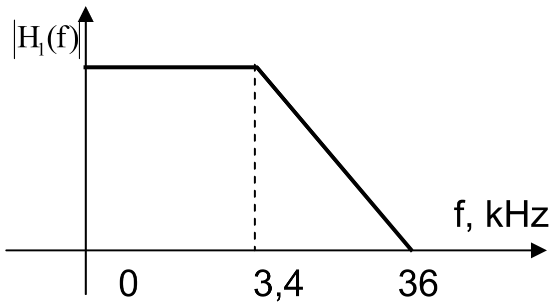


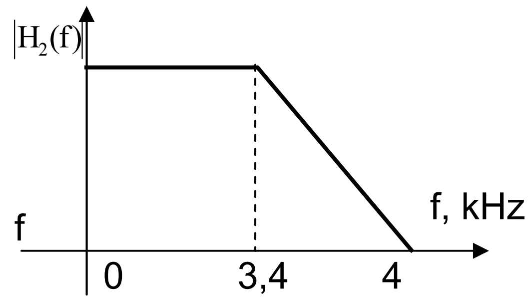
For h1 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=40-240/2x30=36 kHz**

$$\Delta f = \frac{f_{s1} - fp}{Fs} = \frac{36 - 3.4}{240} = 0.1358; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s1} = 0.01; \quad N1 \approx 13$$

For h2 filter bandedge frequencies: **fp=3.4 kHz** and **fs= Fi-Fs/2M=8-4=4 kHz**

$$\Delta f = \frac{f_{s2} - f_p}{F_{s1}} = 0.015; \quad \delta_{p1} = \frac{0.05}{2} = 0.025; \delta_{s2} = 0.01; \quad N2 = 106$$





MPS=(13x40+106x8)10^3=1368x10^3 ; TSR=13+106=119

Decimation Factors	Length, N	MPS	TSR
M1=15;M2=2	**N1=45,N2=43**	**1064x10^3**	**88**
M1=10;M2=3	N1=24,N2=64	1088x10^3	88
M1=6;M2=5	N1=13,N2=106	1368x10^3	119
M=30	549	4392x10^3	549

Optimum ← (M1=15;M2=2)

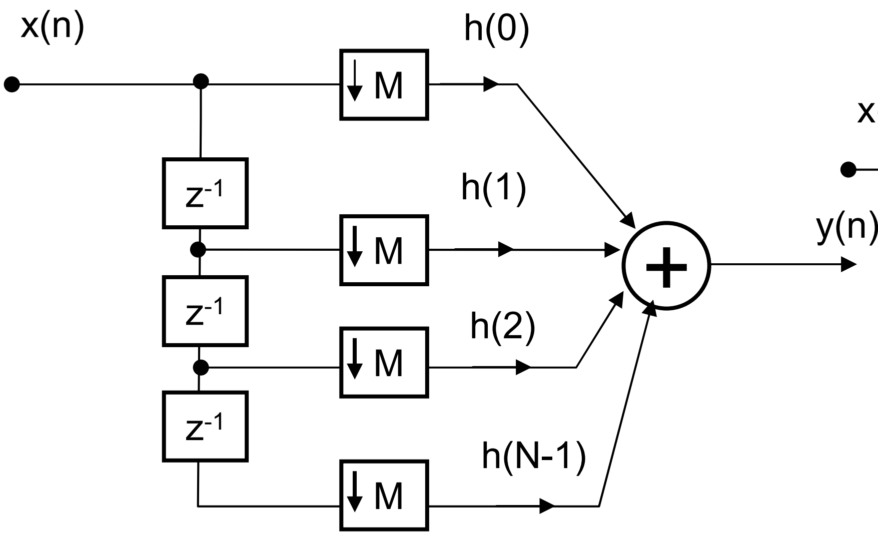
Implementation of Decimamtion and Interpolation

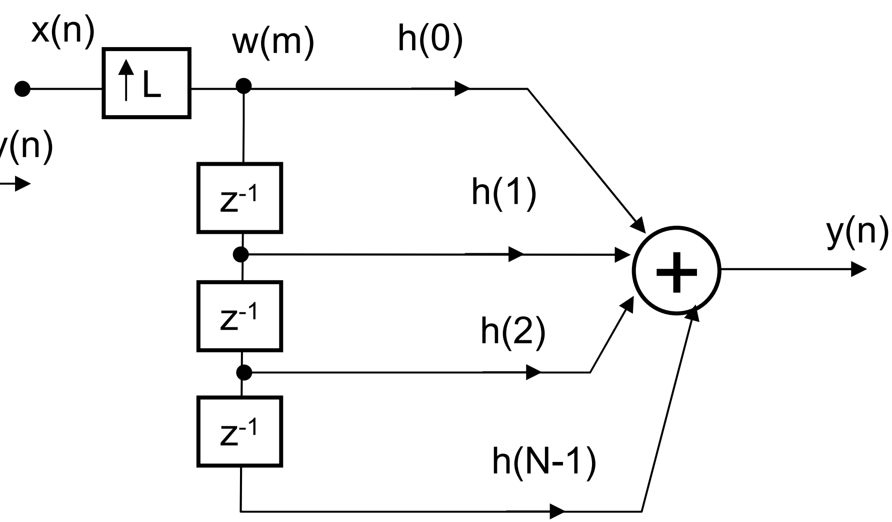
$$y(m) = w(mM) = \sum_{k=-\infty}^{\infty} h(k)x(mM - k)$$

$$y(m) = \sum_{k=-\infty}^{\infty} h(k)w(m-k)$$

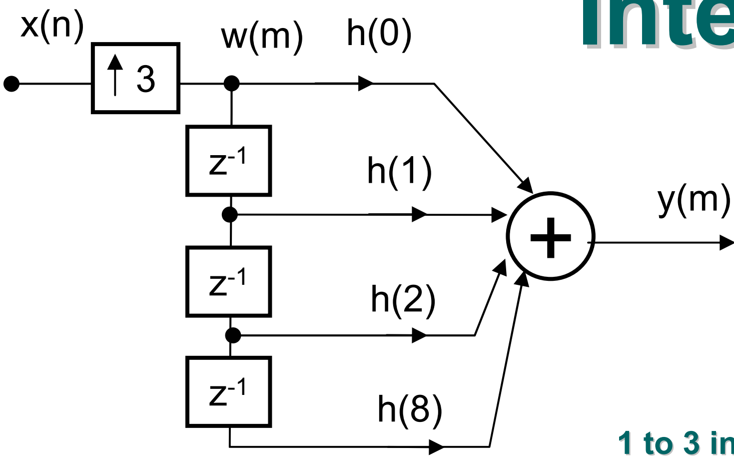
$$w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

$$W(m) = \begin{cases} x(m/L) & m=0, \pm L, \pm 2L, \pm 3L, \ldots \\ 0) & \end{cases}$$





Polyphase Implementation of Interpolators



$y(0)= x(0)\,h(0)$
$y(1)= x(0)h(1)$
$y(2)=x(0)h(2)$

$Y(3)=x(1)\,h(0)$
$Y(4)=x(1)h(1)+x(0)h(4)$
$Y(5)= x(1)h(2)+x(0)h(5)$

$Y(6)= x(2)h(0)+x(1)h(3)+x(0)h(6)$
$Y(7)= x(2)h(1)+x(1)h(4)+x(0)h(7)$
$Y(8)= x(2)h(2)+x(1)h(5)+x(0)h(8)$

$y(9)=x(3)h(0)+x(2)h(3)+x(1)h(6)$
$Y(10)=x(3)h(1)+x(2)h(4)+x(1)h(7)$
$Y(11)= x(3)h(2)+x(2)h(5)+x(1)h(8)$

1 to 3 interpolation using polyphase filter

h \ n	0	-	-	1	-	-	2	-	-	3	-	-
h(0)	x(0)	0	0	x(1)	0	0	x(2)	0	0	x(3)	0	0
h(1)	0	x(0)	0	0	x(1)		0	x(2)	0	0	x(3)	0
h(2)	0	0	x(0)	0	0	x(1)	0	0	x(2)	0	0	x(3)
h(3)	0	0	0	x(0)	0	0	x(1)	0	0	x(2)	0	0
h(4)	0	0	0	0	x(0)	0	0	x(1)	0	0	x(2)	0
h(5)	0	0	0	0	0	x(0)	0	0	x(1)	0	0	x(2)
h(6)	0	0	0	0	0	0	x(0)	0	0	x(1)	0	0
h(7)	0	0	0	0	0	0	0	x(0)	0	0	x(1)	0
h(8)	0	0	0	0	0	0	0		x(0)	0	0	x(1)
y(m)	**y(0)**	**y(1)**	**y(2)**	**y(3)**	**y(4)**	**y(5)**	**y(6)**	**y(7)**	**y(8)**	**y(9)**	**y(10)**	**y(11)**

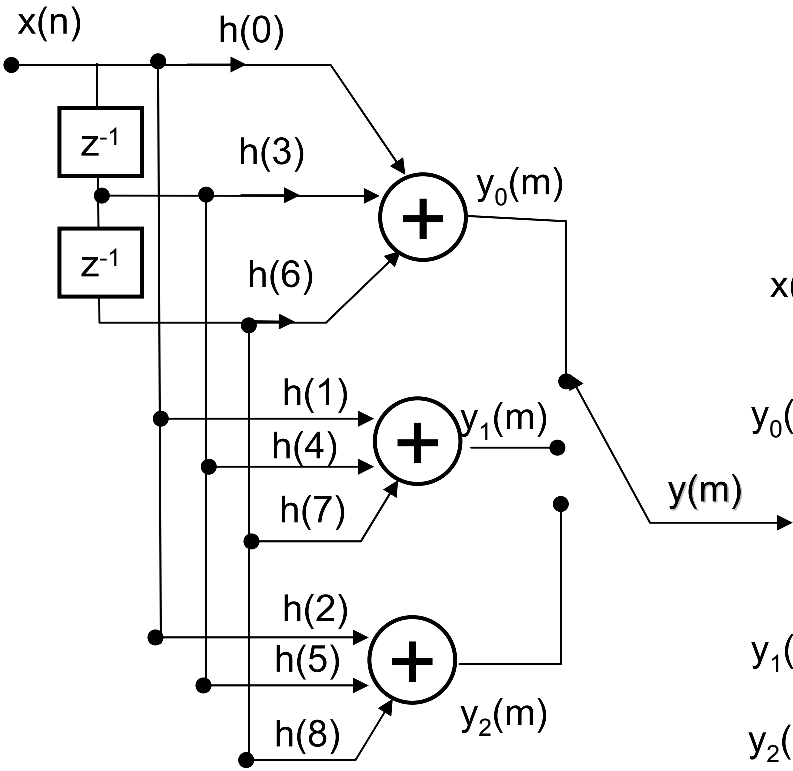
Polyphase Filter Implementation

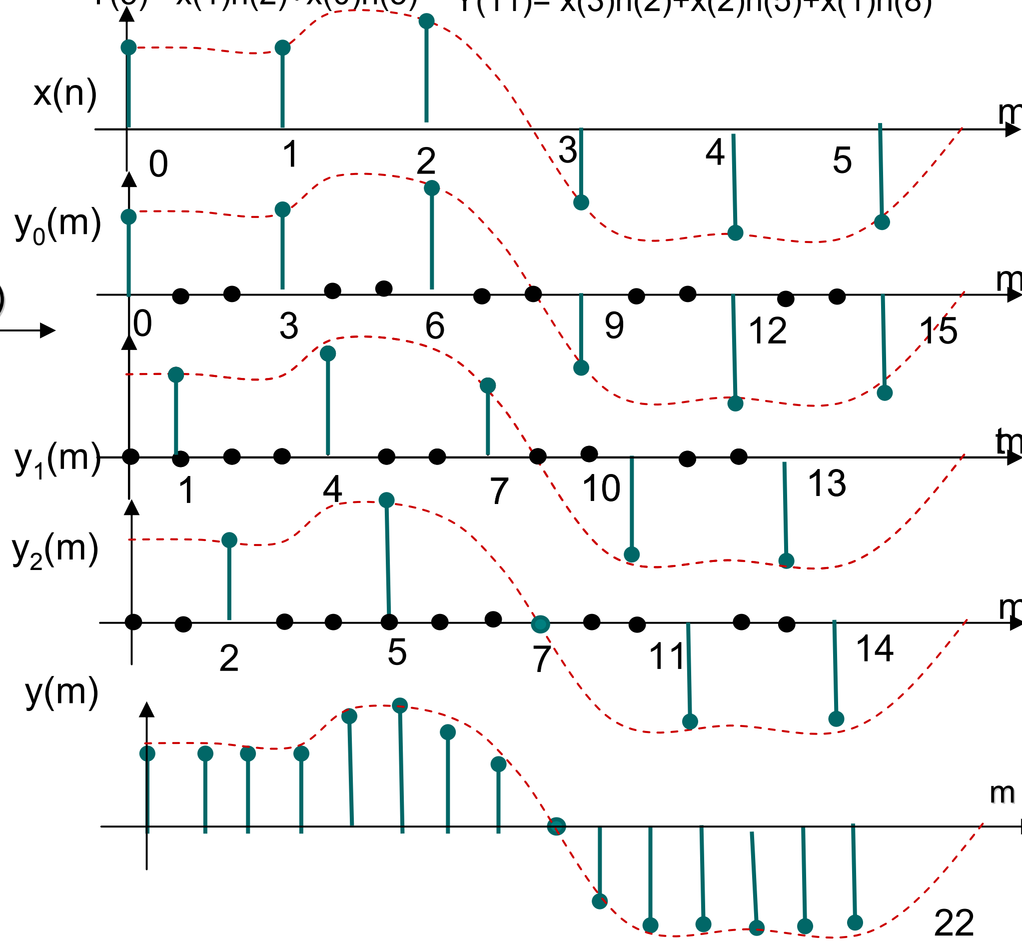
y(0)= x(0) h(0)
y(1)= x(0)h(1)
y(2)=x(0)h(2)

Y(3)=x(1) h(0)
Y(4)=x(1)h(1)+x(0)h(4)
Y(5)= x(1)h(2)+x(0)h(5)

Y(6)= x(2)h(0)+x(1)h(3)+x(0)h(6)
Y(7)= x(2)h(1)+x(1)h(4)+x(0)h(7)
Y(8)= x(2)h(2)+x(1)h(5)+x(0)h(8)

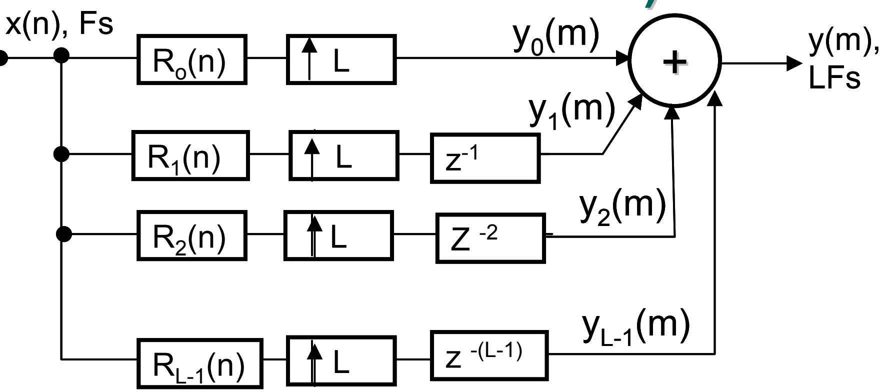
y(9)=x(3)h(0)+x(2)h(3)+x(1)h(6)
Y(10)=x(3)h(1)+x(2)h(4)+x(1)h(7)
Y(11)= x(3)h(2)+x(2)h(5)+x(1)h(8)





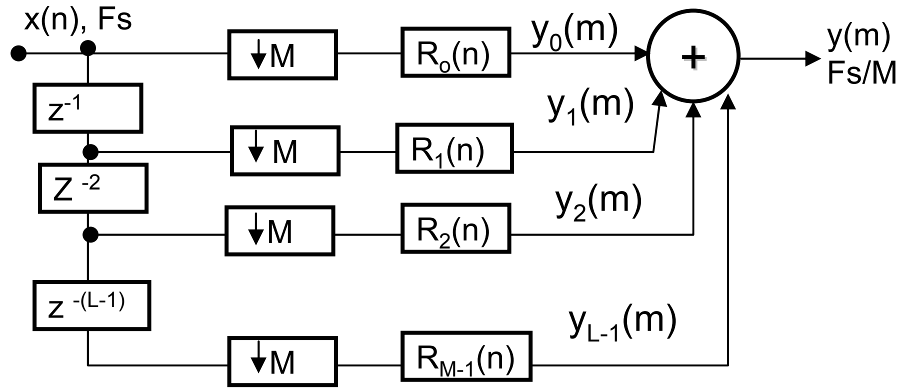
Models of the Polyphase Filters

a) General Models



The coefficients for interpolation:

$Rk(n)=h(k+nL), \quad k=0, 1, \ldots L-1;\ n=0, 1, \ldots (N/L)-1;$



The coefficients for decimation:

$Rk(n)=h(k+nM), \quad k=0, 1, \ldots M-1;\ n=0, 1, \ldots (N/M)$

b) Commutative Models