ECE 6560
Multirate Signal Processing
Decimation and Interpolation

Dr. Bradley J. Bazuin
Western Michigan University
College of Engineering and Applied Sciences
Department of Electrical and Computer Engineering
1903 W. Michigan Ave.
Kalamazoo MI, 49008-5329

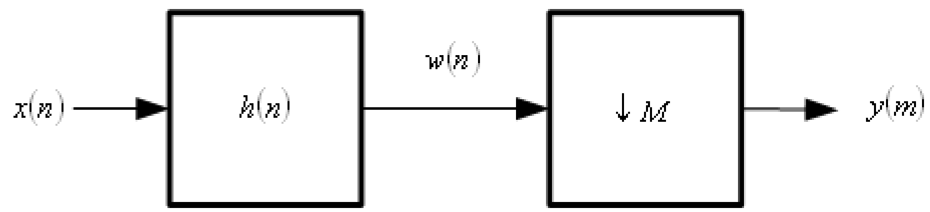
References

- Crochiere, R.E.; Rabiner, L.R.; , "Interpolation and decimation of digital signals—A tutorial review," Proceedings of the IEEE , vol.69, no.3, pp. 300- 331, March 1981.
  - http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01456237

- R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.

Filter-Decimator Implementation (1)

- Deriving a computational structure



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

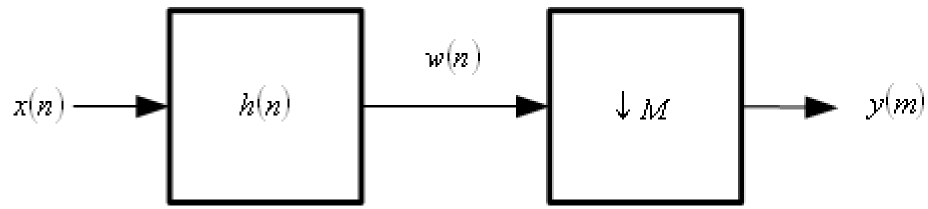
$$y(m) = w(m \cdot M) = \sum_{p=-\infty}^{\infty} h(p) \cdot x(m \cdot M - p)$$

$$p = r \cdot M + \rho$$

$$y(m) = w(m \cdot M) = \sum_{r=-\infty}^{\infty} \sum_{\rho=0}^{M-1} h(r \cdot M + \rho) \cdot x(m \cdot M - r \cdot M - \rho)$$

$$y(m) = \sum_{\rho=0}^{M-1} \left\{ \sum_{r=-\infty}^{\infty} h(r \cdot M + \rho) \cdot x((m-r) \cdot M - \rho) \right\}$$

Filter-Decimator Implementation (2)



$$y(m) = \sum_{\rho=0}^{M-1} \left\{ \sum_{r=-\infty}^{\infty} h(r \cdot M + \rho) \cdot x((m-r) \cdot M - \rho) \right\}$$

Implementation: (1) Generate polyphase elements
(2) Sum the polyphase elements

$$y_\rho(m) = \sum_{r=-\infty}^{\infty} h(r \cdot M + \rho) \cdot x((m-r) \cdot M - \rho) \qquad for \qquad \rho = 0 : M-1$$

$$y(m) = \sum_{\rho=0}^{M-1} y_\rho(m)$$

Filter-Decimator Implementation (3)

- Using a causal filter of length N=λM

Implementation: (1) Generate polyphase elements
(2) Sum the polyphase elements

$$y_\rho(m)=\sum_{r=0}^{\lambda-1} h(r\cdot M+\rho)\cdot x((m-r)\cdot M-\rho) \qquad for \qquad \rho=0:M-1$$

$$y(m)=\sum_{\rho=0}^{M-1} y_\rho(m)$$

Coefficient and Data Sets

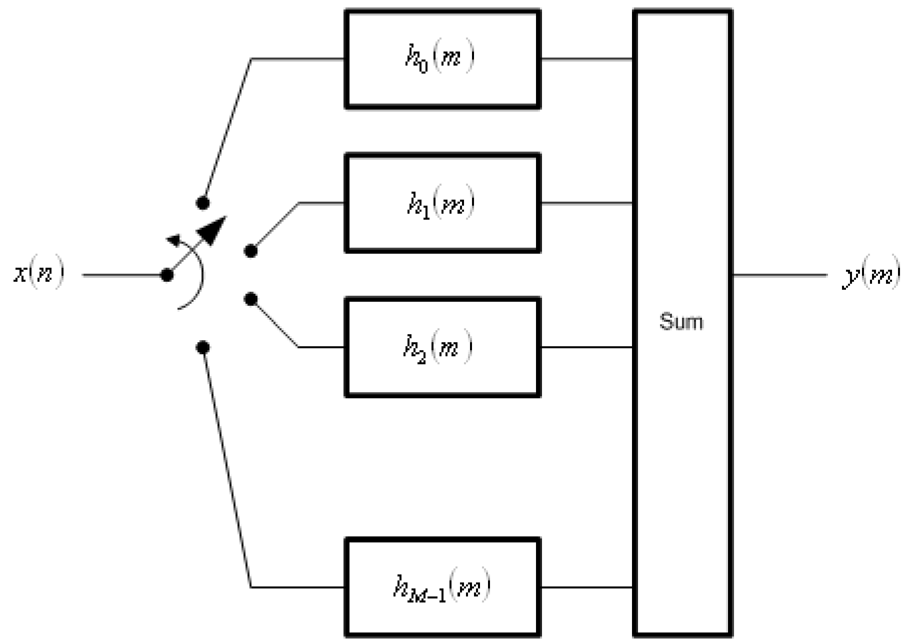
$$h_\rho(r)=h(r\cdot M+\rho) \qquad for \qquad r=0:\lambda-1$$

$$x_\rho((m-r)\cdot M)=x((m-r)\cdot M-\rho) \qquad for \qquad \rho=0:M-1$$

$$y_\rho(m)=\sum_{r=0}^{\lambda-1} h_\rho(r)\cdot x_\rho((m-r)\cdot M) \qquad for \qquad \rho=0:M-1$$

Polyphase Filter-Decimator

$$y(m) = \sum_{\rho=0}^{M-1} \left\{ \sum_{r=0}^{\lambda-1} h(r \cdot M + \rho) \cdot x((m-r) \cdot M - \rho) \right\}$$



Matrix Implementation

$$h_\rho(r) = h(r \cdot M + \rho) \qquad for \qquad r = 0 : \lambda - 1$$

$$x_\rho((m-r)\cdot M) = x((m-r)\cdot M - \rho) \qquad for \qquad \rho = 0 : M - 1$$

$$y_\rho(m) = \sum_{r=0}^{\lambda-1} h_\rho(r) \cdot x_\rho((m-r)\cdot M) \qquad for \qquad \rho = 0 : M - 1$$

$$y(m) = \sum_{\rho=0}^{M-1} y_\rho(m)$$

ρ rows
M columns

Newest sample

Locations based on previous page

$$\begin{bmatrix} y_0(m) \\ \\ y_\rho(m) \\ \\ y_{M-1}(m) \end{bmatrix} = Sum_{row}\left(\begin{bmatrix} x_{m-0} & \Lambda & x_{m-r\cdot M-0} & \Lambda & x_{m-(\lambda-1)\cdot M-0} \\ \mathrm{M} & \mathrm{O} & & & \mathrm{M} \\ x_{m-\rho} & & x_{m-r\cdot M-\rho} & & x_{m-(\lambda-1)\cdot M-\rho} \\ \mathrm{M} & & & \mathrm{O} & \mathrm{M} \\ x_{m-(M-1)} & \Lambda & x_{m-r\cdot M-(M-1)} & \Lambda & x_{m-(\lambda\cdot M-1)} \end{bmatrix} .* \begin{bmatrix} h_0 & \Lambda & h_{r\cdot M+0} & \Lambda & h_{(\lambda-1)\cdot M-0} \\ \mathrm{M} & \mathrm{O} & & & \mathrm{M} \\ h_\rho & & h_{r\cdot M+\rho} & & h_{(\lambda-1)\cdot M+\rho} \\ \mathrm{M} & & & \mathrm{O} & \mathrm{M} \\ h_{(M-1)} & \Lambda & h_{r\cdot M+(M-1)} & \Lambda & h_{\lambda\cdot M-1} \end{bmatrix}\right)$$

Oldest sample

$$y(m) = \sum_{\rho=0}^{M-1} y_\rho(m)$$

Matlab Implementation

Example Matlab Code Execution for y(m) given h and x(m-(lambda*M-1):m-0)

```
Xmatrix = flipud(fliprl(reshape(x,M,lambda);
Hmatrix= reshape(h,M,lambda);
Yrho = sum(Xmatrix.*Hmatrix,2);
Y = sum(Yrho);
```

Updating Xmatrix

```
Xmatrix(:,2:lambda) = Xmatrix(:,1:lambda-1);
Xmatrix(:,1) = flipud(x(1:M)^T);
```

Filter and Decimate

- Input at 20 kHz, with an output data rate only at 400 Hz.
  - Compute the 360 tap filter once every 50 input data cycles, but the data locations must be maintained.
  - What if we took the 360 tap filter an partitioned it into 50 sampled filters?

$$\phi_i = h(i : 50 : end)$$

  - Each filter would get one new input every 50 clock cycles
  - These is a polyphase implementation

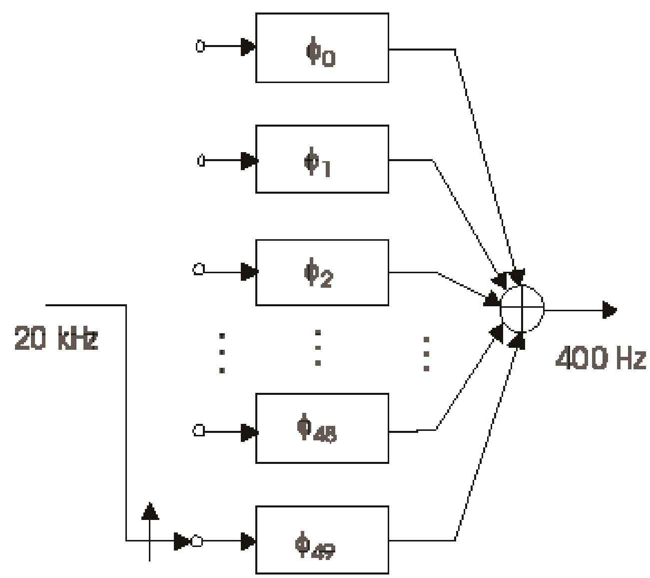


Figure 5.3 50-to-1 Polyphase Partition and Down Sampling of Low-pass Filter

MATLAB Polyphase Filter Decimator

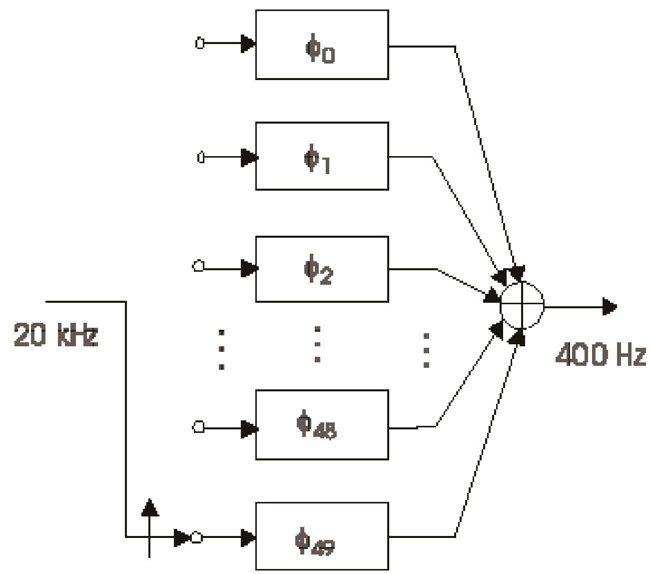
- Chap5_2.m

```
lambda = length(h1)/polytaps;
M = polytaps;

Pfilter1 = reshape(h1,M,lambda);

xarray = zeros(M,lambda);
xshift = [zeros(lambda-1,1) eye(lambda-1) ;
          zeros(1,lambda)];

for ii = 1:numblocks
   xarray = xarray * xshift;
   Tindex = 1+((ii-1)*M:(ii*M-1))';
   xarray(:,1) = flipud(TestSig(Tindex));
   yvect(:,ii) = sum(((xarray)) .* Pfilter1,2);
end
yout1 = sum(yvect).';
```



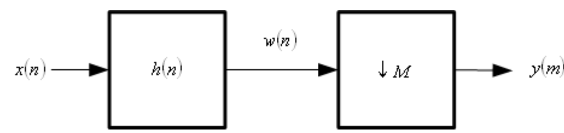
The Z-Transform of w, x and y

$$w'(n)=\begin{cases} w(n), & n=0,\pm M,\pm 2M \Lambda \\ 0, & otherwise \end{cases}$$

Note: w'(n) is not decimated

$$w'(n)=w(n)\left[\frac{1}{M}\sum_{l=0}^{M-1}\exp\left(j2\pi\cdot\frac{l\cdot n}{M}\right)\right],\quad -\infty<n<\infty$$

The discrete Fourier series of an impulse train with period M



$$Y(z)=\sum_{m=-\infty}^{\infty} y(m)\cdot z^{-m}$$

$$Y(z)=\sum_{m=-\infty}^{\infty} w'(mM)\cdot z^{-m}=\sum_{m=-\infty}^{\infty} w'(m)\cdot z^{-m/M}$$

$$Y(z)=\sum_{m=-\infty}^{\infty} w(m)\left[\frac{1}{M}\sum_{l=0}^{M-1}\exp\left(j2\pi\cdot\frac{l\cdot m}{M}\right)\right]\cdot z^{-m/M}$$

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Studying the Z-Transform (2)

$$Y(z)=\sum_{m=-\infty}^{\infty} w(m)\left[\frac{1}{M}\sum_{l=0}^{M-1}\exp\left(j2\pi\cdot\frac{l\cdot m}{M}\right)\right]\cdot z^{-m/M}$$

$$Y(z)=\frac{1}{M}\sum_{l=0}^{M-1}\sum_{m=-\infty}^{\infty} w(m)\cdot\exp\left(j2\pi\cdot\frac{l\cdot m}{M}\right)\cdot z^{-m/M}$$

$$Y(z)=\frac{1}{M}\sum_{l=0}^{M-1} W\left(\exp\left(-j2\pi\cdot\frac{l}{M}\right)\cdot z^{-1/M}\right)$$

Since

$$W(z)=H(z)\cdot X(z)$$

$$Y(z)=\frac{1}{M}\sum_{l=0}^{M-1} H\left(\exp\left(-j2\pi\cdot\frac{l}{M}\right)\cdot z^{1/M}\right)\cdot X\left(\exp\left(-j2\pi\cdot\frac{l}{M}\right)\cdot z^{1/M}\right)$$

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Z-Transform in terms of the Frequency

$$z = \exp\left(jw'\right) \quad where\, w' = 2\pi \cdot f \cdot T'$$

$$Y\left(e^{jw'}\right) = \frac{1}{M}\sum_{l=0}^{M-1} H\left(\exp\left(\frac{-j2\pi \cdot l}{M}\right) \cdot \exp\left(j\frac{w'}{M}\right)\right) \cdot X\left(\exp\left(\frac{-j2\pi \cdot l}{M}\right) \cdot \exp\left(j\frac{w'}{M}\right)\right)$$

$$Y\left(e^{jw'}\right) = \frac{1}{M}\sum_{l=0}^{M-1} H\left(\exp\left(j\frac{w' - 2\pi \cdot l}{M}\right)\right) \cdot X\left(\exp\left(\frac{w' - 2\pi \cdot l}{M}\right)\right)$$

$$Y\left(e^{jw'}\right) = \frac{1}{M}\left\{H\left(e^{jw'}\right) \cdot X\left(e^{jw'}\right) + H\left(e^{j\left(w' - 2\pi/M\right)}\right) \cdot X\left(e^{j\left(w' - 2\pi/M\right)}\right) + \Lambda\right\}$$

Note: The Nyquist zone are included/aliased based on the filter response. For a low pass filter signal or filter …

$$Y\left(e^{jw'}\right) = \frac{1}{M} H\left(e^{jw'}\right) \cdot X\left(e^{jw'}\right) = \frac{1}{M} X\left(e^{jw'}\right)$$

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Bandpass Filter

- What would happen if a bandpass filters were used instead of a low pass filter?

$$Y\left(e^{jw'}\right) = \frac{1}{M}\left\{H\left(e^{jw'}\right)\cdot X\left(e^{jw'}\right) + H\left(e^{j\left(w' - 2\pi/M\right)}\right)\cdot X\left(e^{j\left(w' - 2\pi/M\right)}\right) + \Lambda\right\}$$

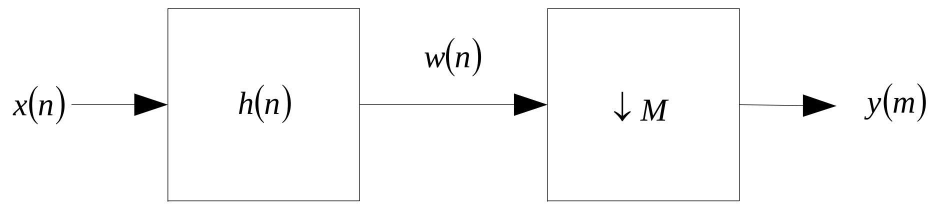
- It should work the same way ….

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Nyquist Zones of a Polyphase Filter

- Chap6_1.m and Chap6_2.m
  - Magnitude and phase plots of non-decimated polyphase filter elements.
  - When a complex filter/complex signal mixing is performed, the phase plots are only equal in the Nyquist band of the perfectly centered complex mixing signal.
    - All others should provide phase sums that cancel signal in the Nyquist band.

Crochiere and Rabiner
Filter Decimation Summarized (1)



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

$$y(m) = w(mM)$$

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

Assume a causal FIR filter of length λM:

$$h(k) = \begin{cases} 0, & for\ k < 0 \\ h(k), & for\ 0 \le k \le \lambda M - 1 \\ 0, & for\ \lambda M - 1 < k \end{cases}$$

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Filter Decimation Summarized (2)

$$y(m)=\sum_{k=0}^{\lambda M-1} h(k)\cdot x(mM-k)$$

Let $k = rM + \rho$

$$y(m)=\sum_{r=0}^{\lambda-1}\sum_{\rho=0}^{M-1} h(rM+\rho)\cdot x(mM-rM-\rho)$$

$$y(m)=\sum_{\rho=0}^{M-1}\sum_{r=0}^{\lambda-1} h(rM+\rho)\cdot x((m-r)M-\rho)$$

$$y(m)=\sum_{\rho=0}^{M-1}\left\{\sum_{r=0}^{\lambda-1} h_\rho(r)\cdot x_\rho(m-r)\right\}$$

$$h_\rho(r)=h(rM+\rho) \qquad x_\rho(m-r)=x((m-r)M-\rho)$$

The summation of M λ-tap filters.

The computation is performed once every M input samples.

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Vector Interpretation

$$y(m)=\sum_{k=0}^{\lambda M-1} h(k)\cdot x(mM-k) \qquad y(m)=\sum_{\rho=0}^{M-1}\sum_{r=0}^{\lambda-1} h_\rho(r)\cdot x_\rho(m-r)$$

$$h_\rho(r)=h(rM+\rho) \qquad x_\rho(m-r)=x((m-r)M-\rho)$$

x has λ columns of M rows, with columns numbered left-to-right in time as r increases and bottom-to-top as "–ρ" increases.

h has λ columns of M rows, with columns numbered left-to-right as r increases and top-to-bottom as ρ increases. (x & h must convolve)

To perform point-wise multiplication, if x is stored left-to-right and top-to-bottom, fliplr h and then flipud and it will line up correctly for point-wise multiplication and summation. (This can be done to x or h to convolve).

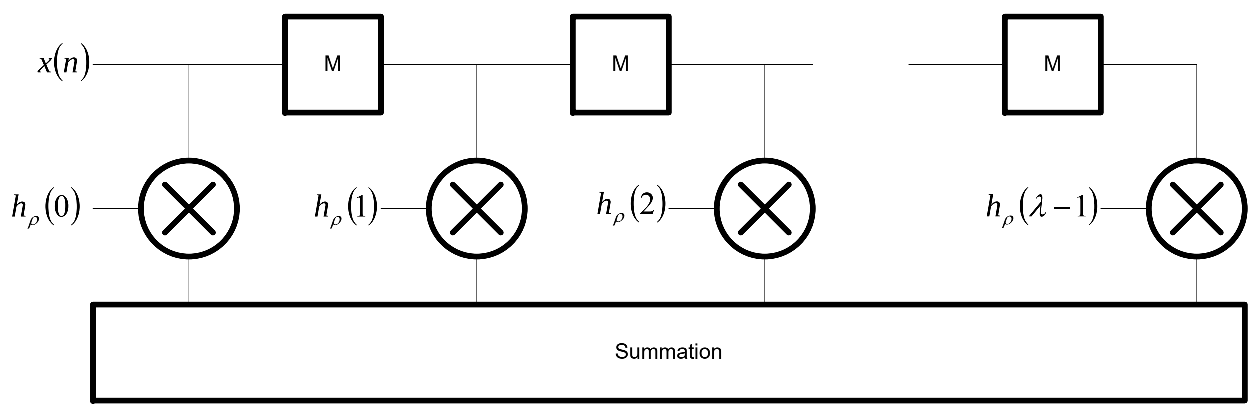
This appears as a "matrix" convolution of the sample vectors.

From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Polyphase Implementation (1)

Delay Line Based

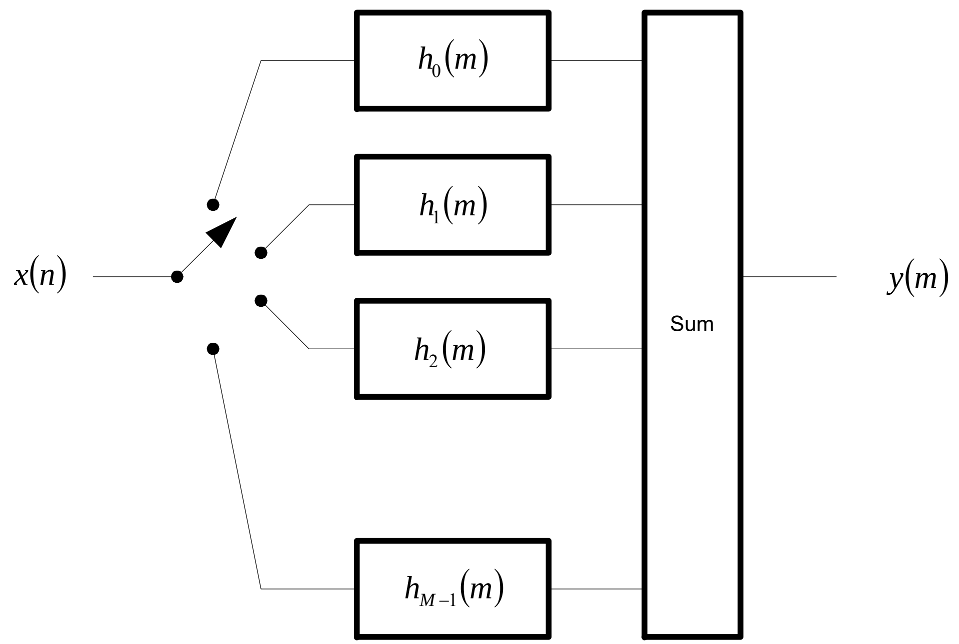
Block M
delays



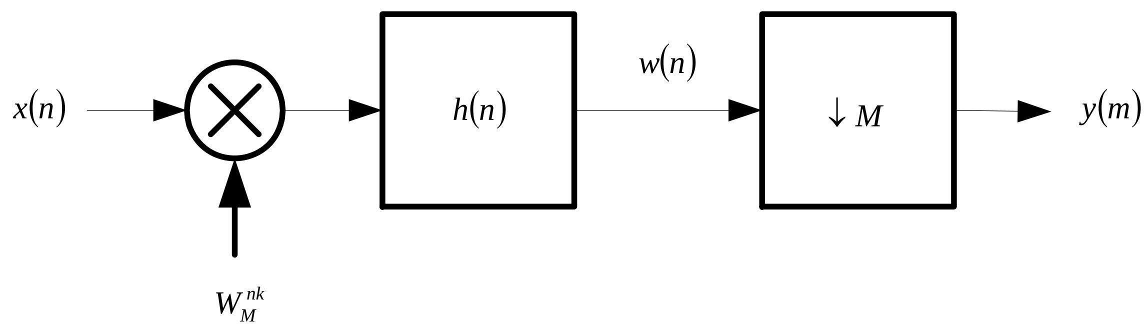
ρ coefficients
applied
sequentially

Polyphase Implementation (2)

Commutator Base



What about Mixing Prior to Filter Decimation



$$y_k(m) = \sum_{n=-\infty}^{\infty} h(n) \cdot x(mM - n) \cdot \exp\left(-j2\pi \cdot \frac{(mM - n) \cdot k}{N} \right)$$

$$y_k(m) = \sum_{n=-\infty}^{\infty} h(n) \cdot x(mM - n) \cdot \exp\left(-j2\pi \cdot m \cdot k \cdot \frac{M}{N} \right) \cdot \exp\left(j2\pi \cdot \frac{n \cdot k}{N} \right)$$

Let $M = N$

$$y_k(m) = \sum_{n=-\infty}^{\infty} \left\{ h(n) \cdot \exp\left(j2\pi \cdot \frac{n \cdot k}{N} \right) \right\} \cdot x(mM - n)$$

Note equivalence to a complex filter

Mixing Continued

Let $\quad n = rM + \rho$

$$y_k(m) = \sum_{r=0}^{\lambda-1} \sum_{\rho=0}^{M-1} h(rM+\rho) \cdot x(mM - rM + \rho) \cdot \exp\left(j2\pi \cdot \frac{(rM+\rho) \cdot k}{M} \right)$$

$$y_k(m) = \sum_{r=0}^{\lambda-1} \sum_{\rho=0}^{M-1} h_\rho(r) \cdot \exp\left(j2\pi \cdot \frac{\rho \cdot k}{M} \right) \cdot x_\rho(m-r)$$

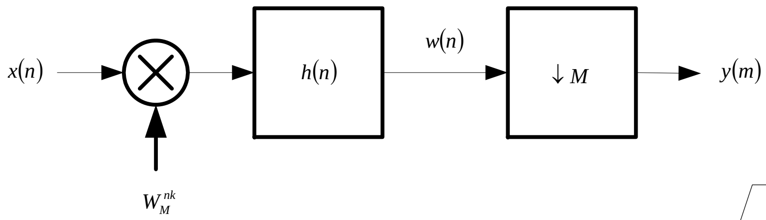
$$y_k(m) = \sum_{\rho=0}^{M-1} \exp\left(j2\pi \cdot \frac{\rho \cdot k}{M} \right) \cdot \sum_{r=0}^{\lambda-1} h_\rho(r) \cdot x_\rho(m-r)$$

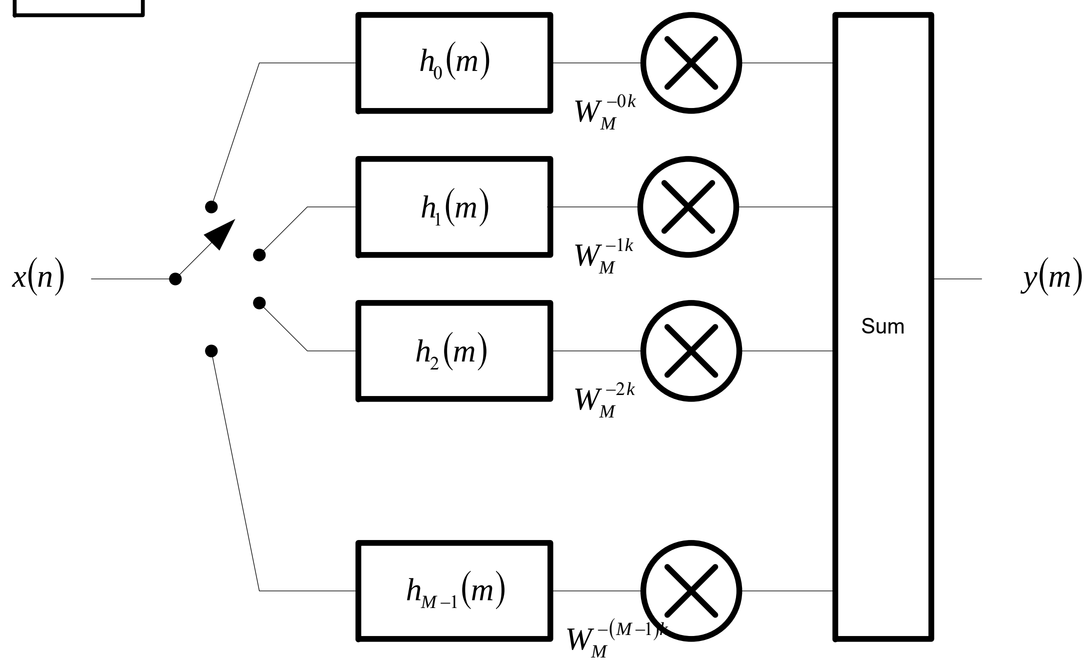
Note, k for one frequency, use an IFFT for all frequencies in k.

The complex weighting and summation of M λ-tap filters.

The computation is performed once every M input samples.

Polyphase Implementation

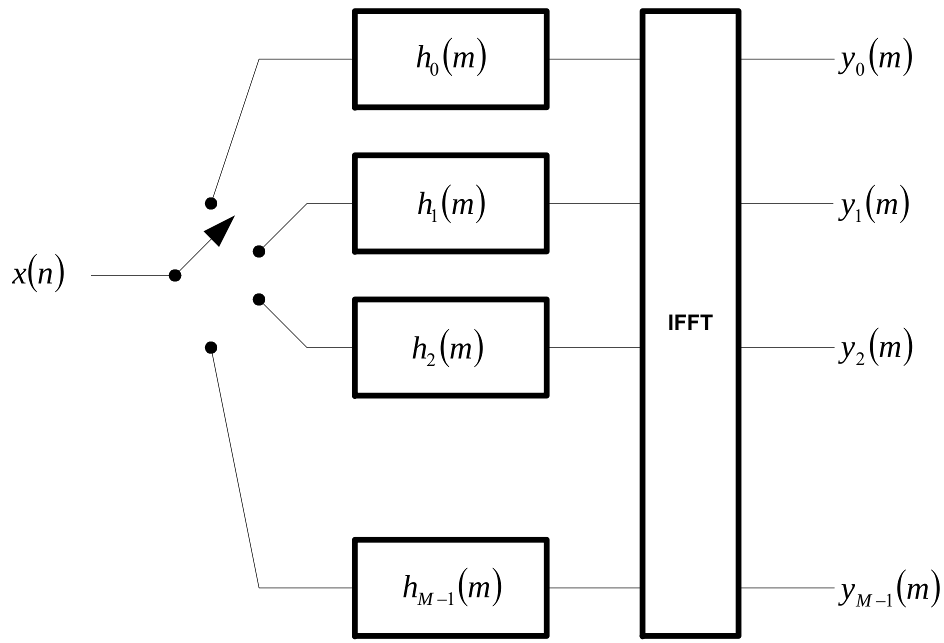




$$y_k(m) = \sum_{\rho=0}^{M-1} \sum_{r=0}^{\lambda-1} h_\rho(r) \cdot \exp\left(j2\pi \cdot \frac{\rho \cdot k}{M} \right) \cdot x_\rho(m-r)$$

A Single Frequency Digital Channel Output.

Polyphase Implementation

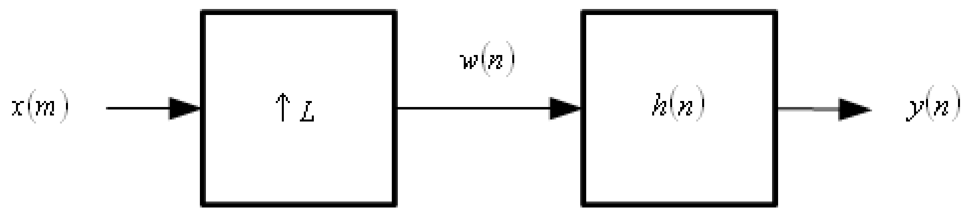


$$y_k(m) = \sum_{\rho=0}^{M-1} \exp\left(j2\pi \cdot \frac{\rho \cdot k}{M} \right) \cdot \sum_{r=0}^{\lambda-1} h_\rho(r) \cdot x_\rho(m-r)$$

A Multichannel "Digital Channelizer"
or "Analysis" Implementation

Interpolate-Filter (1)

- Deriving a computational structure



$$w(n) = \begin{cases} x\left(\frac{n}{L}\right), & \frac{n}{L} = m = 0, \pm 1, \pm 2 \Lambda \\ 0, & otherwise \end{cases}$$

$$y(n) = \sum_{p=-\infty}^{\infty} h(p) \cdot w(n-p)$$

$$y(n) = \sum_{p=-\infty}^{\infty} h(p) \cdot x\left(\frac{n-p}{L}\right)$$

$$p = r \cdot M + \rho \qquad\qquad n = s \cdot M + p$$

Interpolate-Filter (2)

$$y(n) = \sum_{p=-\infty}^{\infty} h(p) \cdot x\left(\frac{n-p}{L}\right)$$

$$p = r \cdot L + \rho$$

$$y(n) = \sum_{\rho=0}^{L-1} \sum_{r=-\infty}^{\infty} h(r \cdot L + \rho) \cdot x\left(\frac{n - r \cdot L - \rho}{L}\right)$$

$$n = s \cdot L + p$$

$$y(s \cdot L + p) = \sum_{\rho=0}^{L-1} \sum_{r=-\infty}^{\infty} h(r \cdot L + \rho) \cdot x\left(\frac{s \cdot L + p - r \cdot L - \rho}{L}\right)$$

$$y(s \cdot L + p) = \sum_{\rho=0}^{L-1} \sum_{r=-\infty}^{\infty} h(r \cdot L + \rho) \cdot x\left((s - r) + \frac{p - \rho}{L}\right)$$

Note that the rho summation only exists for p = ρ making y exist as signal outputs from the r summation

$$y_\rho(s \cdot L) = \sum_{r=-\infty}^{\infty} h(r \cdot L + \rho) \cdot x(s - r)$$

Interpolate-Filter (3)

- Using a causal filter of length λL

$$h(k) = \begin{cases} 0, & for\ k < 0 \\ h(k), & for\ 0 \le k \le \lambda L - 1 \\ 0, & for\ \lambda L - 1 < k \end{cases}$$

Implementation: (1) Generate polyphase elements

(2) Output is individual elements

$$y_\rho(s \cdot L) = \sum_{r=0}^{L-1} h(r \cdot L + \rho) \cdot x(s - r) \qquad for \qquad \rho = 0 : L - 1$$

$$y(s \cdot L + 0 : s \cdot L + \rho) = y_\rho(s \cdot L)$$

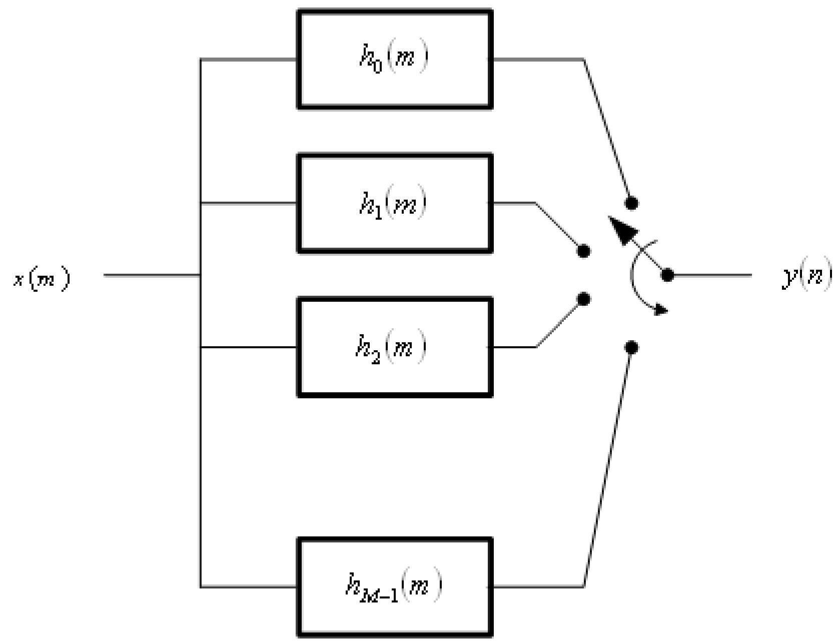
Coefficient sets

$$h_\rho(r) = h(r \cdot L + \rho) \qquad for \qquad r = 0 : \lambda - 1$$

$$y_\rho(s \cdot L) = \sum_{r=0}^{L-1} h_\rho(r) \cdot x(s - r) \qquad for \qquad \rho = 0 : L - 1$$

Polyphase Interpolator-Filter

$$y(s \cdot L + \rho) = \sum_{r=0}^{L-1} h_{\rho}(r) \cdot x(s - r)$$



Matrix Implementation

$$y\left(s \cdot L + \rho\right) = \sum_{r=0}^{L-1} h_{\rho}\left(r\right) \cdot x\left(s - r\right)$$

Oldest sample

Newest sample

$$\begin{bmatrix} y_{m+0} \\ \vdots \\ y_{m+\rho} \\ \vdots \\ y_{m+(L-1)} \end{bmatrix} = \begin{bmatrix} h_0 & \cdots & h_{r \cdot M+0} & \cdots & h_{(\lambda-1)\cdot M-0} \\ \vdots & \ddots & & & \vdots \\ h_{\rho} & & h_{r \cdot M+\rho} & & h_{(\lambda-1)\cdot M+\rho} \\ \vdots & & & \ddots & \vdots \\ h_{(M-1)} & \cdots & h_{r \cdot M+(M-1)} & \cdots & h_{\lambda \cdot M-1} \end{bmatrix} * \begin{bmatrix} x_{m-0} \\ \vdots \\ x_{m-s} \\ \vdots \\ x_{m-(\lambda-1)} \end{bmatrix}$$

Newest sample

Oldest sample

Rows in x
(shift registers)

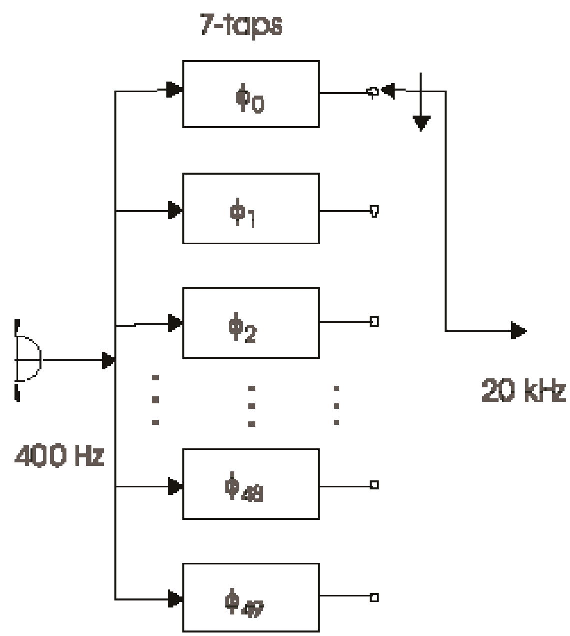
MATLAB Polyphase Interpolate Filter

- Chap5_3.m

```
xivector = zeros(lambda,1);
xshift2 = [zeros(1,lambda);
        eye(lambda-1) zeros(lambda-1,1) ] ;

for ii = 1:numblocks
   xivector = xshift2 * xivector;
   Tindex = 1+((ii-1)*M:(ii*M-1))';
   xivector(1,1) = yout1(ii);
   ymatrix(:,ii) = Pfilter1 * xivector;
end

yout2 = M*reshape((ymatrix),num_samples,1);
```



Vector Interpretation

$$y(n) = \sum_{k=0}^{\lambda L-1} h(k) \cdot x\left(\frac{n-k}{L}\right) \qquad y(sL+t) = \sum_{r=0}^{\lambda-1} h_t(r) \cdot x(s-r)$$

x has λ rows of length 1 columns, with rows numbered top-to-bottom in time as r increases (negative time).

h has λ columns of L rows, with columns numbered left-to-right as r increases and top-to-bottom as t increases.

To perform vector-matrix multiplication, if h is stored left-to-right and top-to-bottom it will line up correctly for vector multiplication with x which is inverted in time. (Note: if x is flipud, then fliplr, flipud h and flipud the output vector).

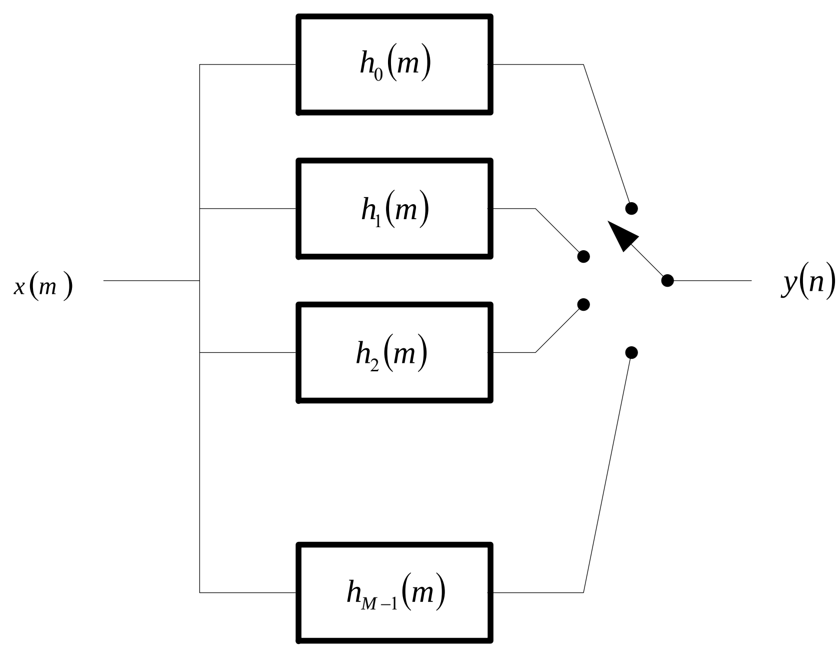
Extract the output y from the column based result top-to-bottom.

This appears as a “matrix” multiplication of the sample vectors.

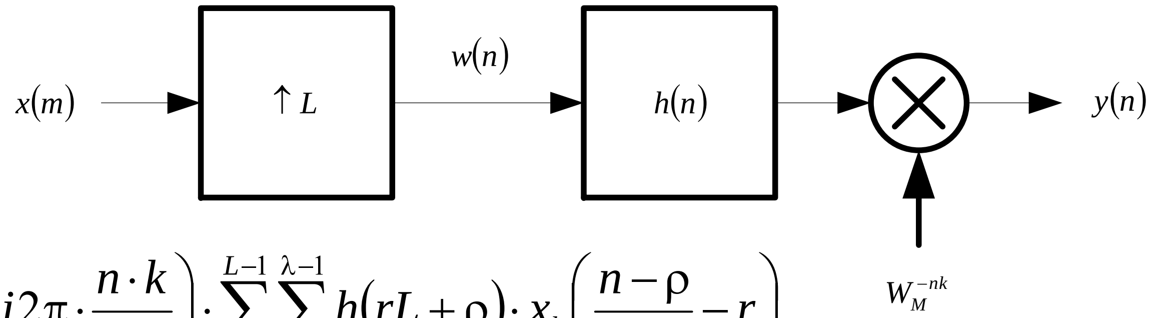
From: R.E. Crochiere and L.R. Rabiner, "Multirate Digital Signal Processing," Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983. Chapter 2, p. 31-39.

Interpolation-Filter Polyphase Implementation

Commutator Base



What about Mixing (Up-conversion) After The Interpolation Filter



$$y(n) = \cdot \exp\left(j2\pi \cdot \frac{n \cdot k}{N}\right) \cdot \sum_{\rho=0}^{L-1} \sum_{r=0}^{\lambda-1} h(rL+\rho) \cdot x_k\left(\frac{n-\rho}{L} - r\right)$$

Let $n = sL + t$

$$y(sL+t) = \exp\left(j2\pi \cdot \frac{(sL+t) \cdot k}{N}\right) \cdot \sum_{\rho=0}^{L-1} \sum_{r=0}^{\lambda-1} h(rL+\rho) \cdot x_k\left(\frac{sL+t-\rho}{L} - r\right)$$

$$y(sL+t) = \exp\left(j2\pi \cdot \frac{sL \cdot k}{N}\right) \cdot \exp\left(j2\pi \cdot \frac{t \cdot k}{N}\right) \cdot \sum_{r=0}^{\lambda-1} h(rL+t) \cdot x_k(s-r)$$

Let $L = N$

$$y(sL+t) = \sum_{r=0}^{\lambda-1} \left[h(rL+t) \cdot \exp\left(j2\pi \cdot \frac{t \cdot k}{N}\right)\right] \cdot x_k(s-r)$$

Complex filter equivalence

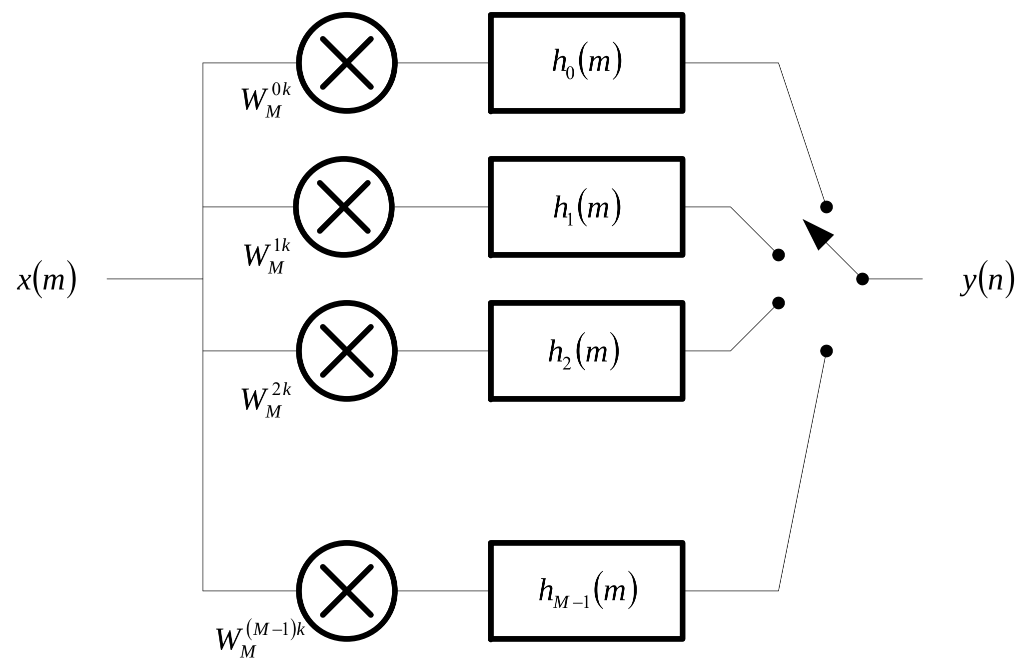
Mixing Polyphase Implementation

$$y(sL+t) = \sum_{r=0}^{\lambda-1} h(rL+t) \cdot \left[x_k(s-r) \cdot \exp\left(j2\pi \cdot \frac{t \cdot k}{N} \right) \right]$$

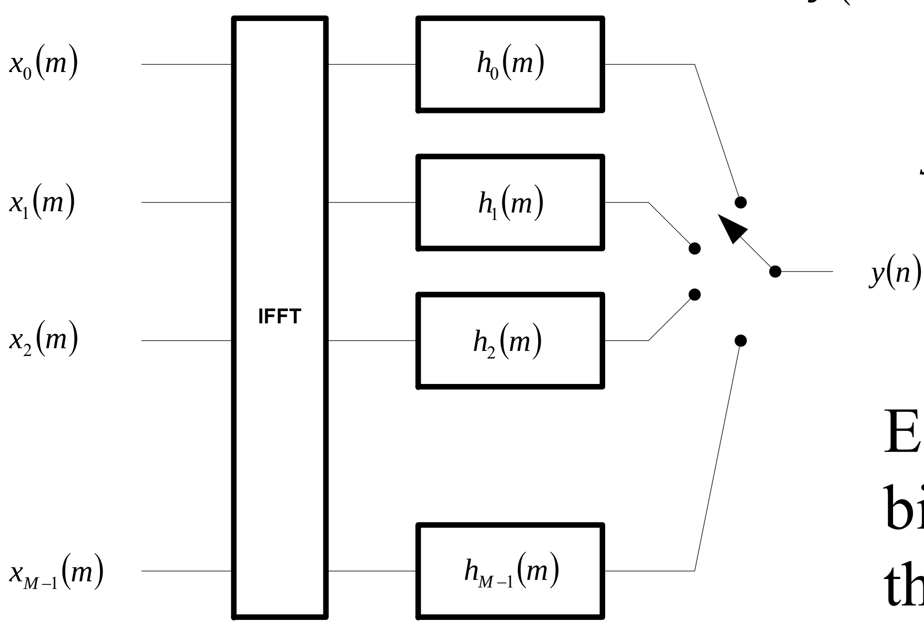
$$y(sL+t) = \sum_{r=0}^{\lambda-1} h_t^{BPF(k)}(r) \cdot x_k(s-r)$$

x_k placed in Nyquist region k.

Filter becomes a complex bandpass filter.



Multiple Input Signal Implementation



$$y(sL+t) = \sum_{k=0}^{N-1} \sum_{r=0}^{\lambda-1} h(rL+t) \cdot \left[x_k(s-r) \cdot \exp\left(j2\pi \cdot \frac{t \cdot k}{N} \right) \right]$$

$$y(sL+t) = \sum_{r=0}^{\lambda-1} h_t(r) \cdot \sum_{k=0}^{N-1} x_k(s-r) \cdot \exp\left(j2\pi \cdot \frac{t \cdot k}{N} \right)$$

Each x is mixed to a different bin/band and summed with all the other k prior to polyphase interpolation.

"Cascaded" Elements

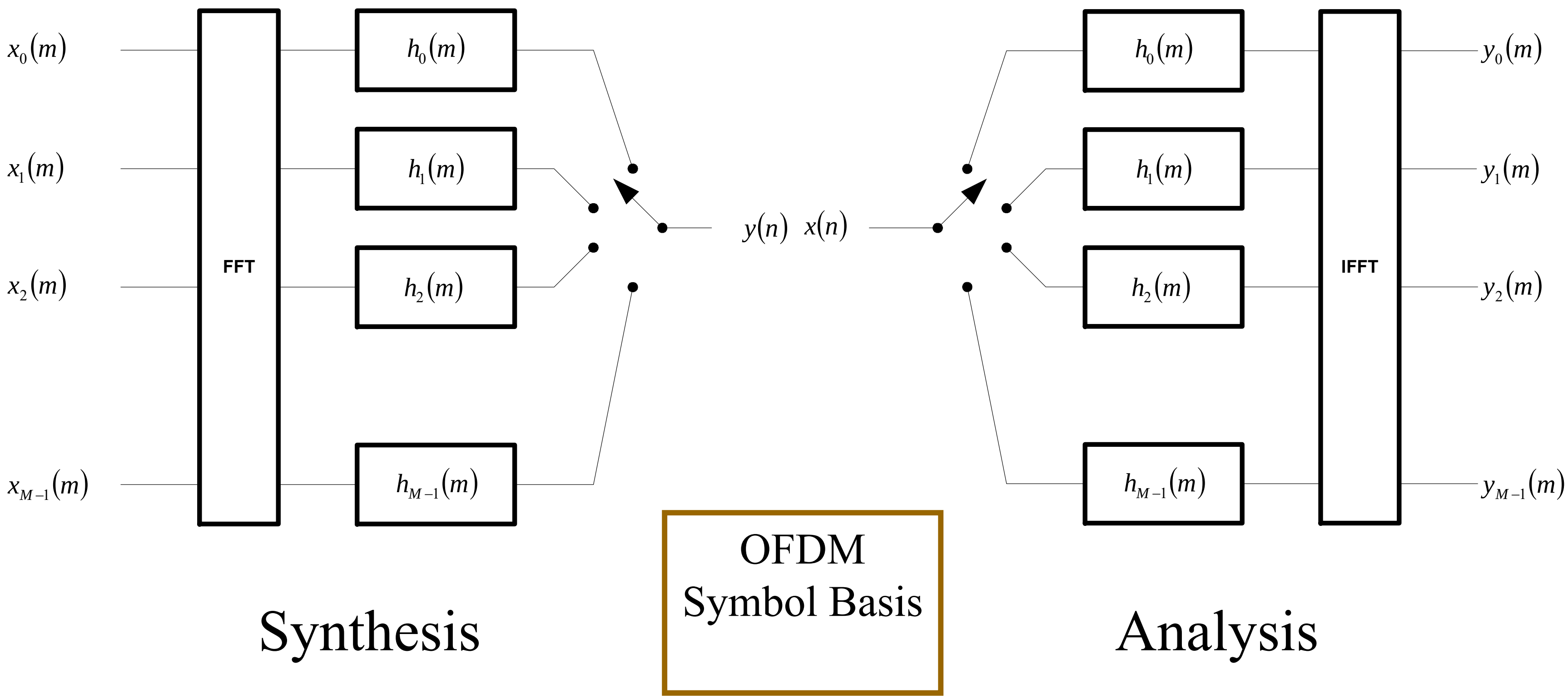
- FFT interpolation-filtering
  - Nyquist rate TDM samples to FDM frequency
  - Complex TDM symbols to FDM output (similar to OFDM symbol generation)

- Filter-decimation with FFT
  - FDM frequencies to narrowband TDM at the Nyquist rate!
  - FDM symbols to Complex TDM output (similar to OFDM symbol reception)

FDM Generation and Processing

Forming Wideband and Reforming Narrowband

Filters narrower than the Nyquist regions are used for generating FDM waveforms.
A guard band between adjacent frequencies is typically used

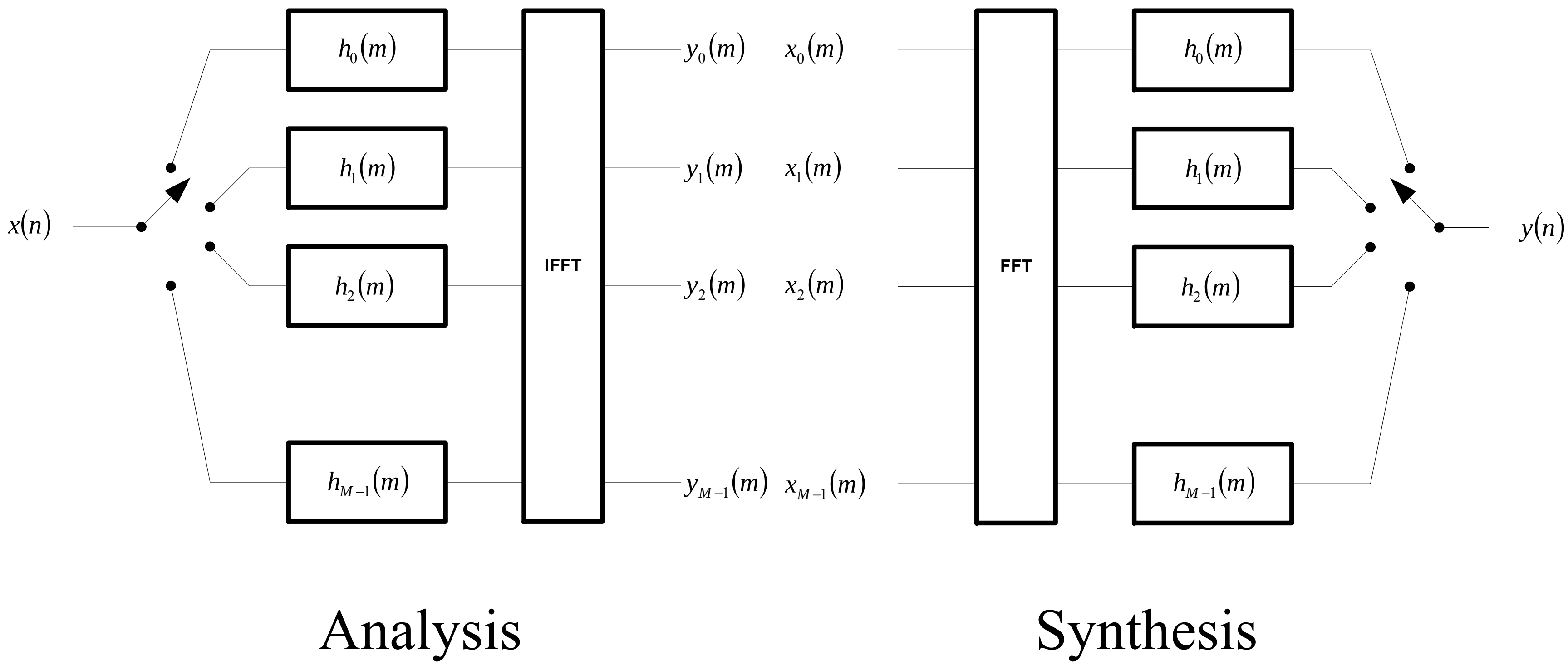


Quadrature Mirror Filter Processing

Observing Narrowband and Reforming Wideband

Significant filter restriction are required if the output is required to approximate the input!

Quadrature Mirror Filter Definition and Requirements



QMF Applications

- Frequency domain filtering or equalization
- Time-Spectral Analysis with reconstruction

- Arbitrarily take signals apart and then reconstruct them
  - Partial-Band Synthesis to one or more arbitrary bandwidths (universal base station receiver)
  - Partial-Band Analysis with frequency domain summation and full-band synthesis (universal base station transmitter)
  - Applications: cellular telephone base stations, satellite relay stations, etc.