AUDIOVISUAL COMMUNICATIONS LCAV

Mathematical Foundations of Signal Processing

Module 6: Multirate Sequences and Systems

Benjamín Béjar
Mihailo Kolundžija
Reza Parhizkar
Martin Vetterli

October 13, 2014

Multirate Systems

Why Multirate?

Multirate is nothing but the combination of the downsampling, upsampling, and filtering operators, where the result of such a combination is strongly dependent on the order of execution of the operators.

A classical example of a multirate system is the changing of the sampling rate: Many everyday used devices implements a change of sampling rate.

TV

Mobile screen

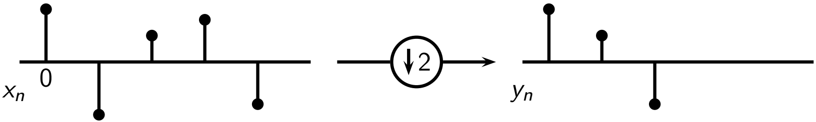
- CDs: 44.1kHz
- DVD: 48 kHz

Audio processing

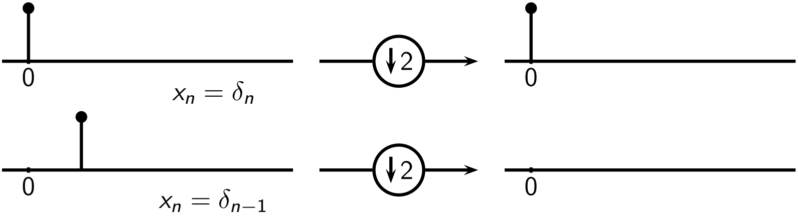
Downsampling

$N = 2$

$$x_n \longrightarrow \boxed{\downarrow 2} \longrightarrow y_n = x_{2n}$$



Downsampling: Origin of time



- What happens if $x_n = \delta_{n-2}$?
- Downsampling is shift variant **but** periodically shift variant
- Downsampling is lossy

Downsampling: Important Formulas

Some important equalities related to downsampling

By 2:

- Time domain

$$y_n = x_{2n}\,;$$

- z-domain

$$Y(z) \;=\; \frac{1}{2}\left[X(z^{1/2}) + X(-z^{1/2})\right]\;;$$

- Fourier domain

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

Downsampling: Important Formulas

By 2 (proof):

- z-domain

$$Y(z) \ = \ \sum_{k\in\mathbb{Z}} x_{2k} z^{-k} \ = \ \sum_{p=2k,k\in\mathbb{Z}} x_p z^{-\frac{p}{2}}.$$

  To keep only even terms we can use the fact that

$$z^k + (-z)^k = \begin{cases} 2z^k & k \text{ even} \\ 0 & \text{otherwise} \end{cases}$$

  Thus

$$Y(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} x_k (z^{\frac{1}{2}})^{-k} + \frac{1}{2}\sum_{k\in\mathbb{Z}} x_k (-z^{\frac{1}{2}})^{-k} = \frac{1}{2}[X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})].$$

- Fourier domain

  We simply have $z = e^{j\omega}$ and $e^{j(\omega-2\pi)/2} = e^{j\omega/2}e^{j2\pi/2} = -e^{j\omega/2}$.

Downsampling: Important Formulas

By N:

- Time domain

$$y_n = x_{Nn}\,;$$

- z-domain

$$Y(z) = \frac{1}{N}\sum_{n=0}^{N-1} X(W_N^n z^{1/N})\,, \text{ where } W_N^n = e^{-j(2\pi/N)n}\,;$$

- Fourier domain

$$Y(e^{j\omega}) = \frac{1}{N}\sum_{n=0}^{N-1} X(e^{j(\omega-2\pi n)/N})\,.$$

Downsampling: Important Formulas

By N (proof):

- z-domain

$$Y(z) = \sum_{k \in \mathbb{Z}} x_{Nk} z^{-k} = \sum_{m=kN, k \in \mathbb{Z}} x_m z^{-\frac{m}{N}} .$$

Using the identity for roots of unity: $\frac{1}{N} \sum_{n=0}^{N-1} W_N^{-nm} = \begin{cases} 1 & \text{when } m = kN \\ 0 & \text{otherwise} \end{cases}$.

We obtain:

$$Y(z) = \frac{1}{N} \sum_{m=kN} x_m \left(\sum_{n=0}^{N-1} W_N^{-nm} z^{-\frac{m}{N}} \right) = \frac{1}{N} \sum_{m \in \mathbb{Z}} \sum_{n=0}^{N-1} x_m \left(W_N^n z^{\frac{1}{N}} \right)^{-m}$$

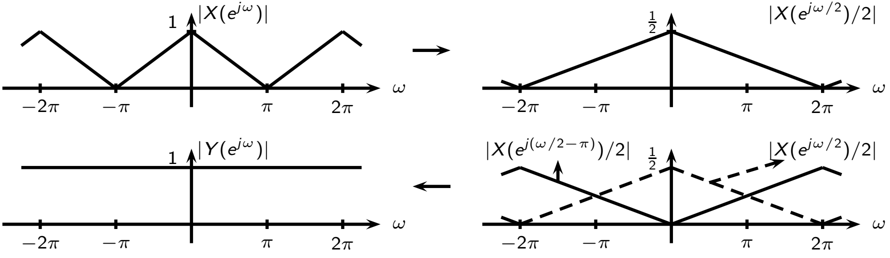
$$= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m \in \mathbb{Z}} x_m \left(W_N^n z^{\frac{1}{N}} \right)^{-m} = \frac{1}{N} \sum_{n=0}^{N-1} X\left(W_N^n z^{\frac{1}{N}} \right) .$$

- Fourier domain

We simply have $z = e^{j\omega}$.

Downsampling: Spectrum

By 2:



- Shrink in time $\rightarrow$ Expand in frequency
- Loss of information in signal reflected by overlapping spectrum (aliasing)

By N:

- $(N-1)$ shifted copies overlap with the central one and create aliasing
- Copies are N times wider than the initial spectrum

Downsampling: Operators

By 2:

- Start with identity, take out every other row: keep even rows only (remember: origin of time is important)

$$
\begin{bmatrix} \vdots \\ y_{-1} \\ \boxed{y_0} \\ y_1 \\ y_2 \\ \vdots \end{bmatrix}
=
\underbrace{\begin{bmatrix}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\cdots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & \boxed{1} & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{bmatrix}}_{D_2}
\begin{bmatrix} \vdots \\ x_{-2} \\ x_{-1} \\ \boxed{x_0} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix}
=
\begin{bmatrix} \vdots \\ x_{-2} \\ \boxed{x_0} \\ x_2 \\ x_4 \\ \vdots \end{bmatrix}
$$

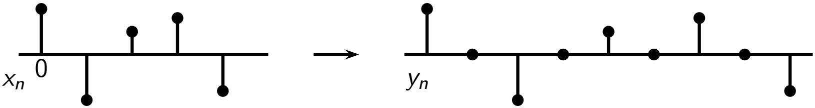
$$y = D_2 \, x$$

By N:

- D_N: Remove $(N-1)$ rows from identity (keep the origin)

Upsampling

- By 2: $x_n \longrightarrow \boxed{\uparrow 2} \longrightarrow y_n$



- By N: Introduce $N-1$ zeros in-between every two samples

Upsampling: Formulas

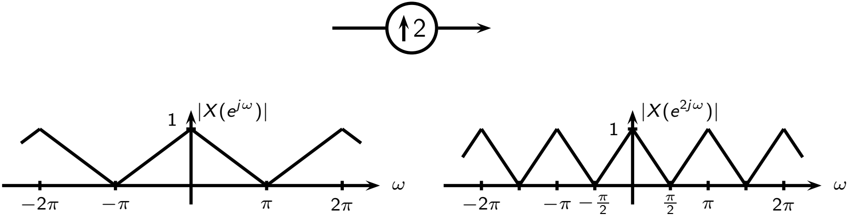
By 2:

- Time domain : $y_n = \begin{cases} x_{\frac{n}{2}} & n = 2k \\ 0 & \text{otherwise} \end{cases}$
- z-domain : $Y(z) \ = \ X(z^2)$
- Fourier domain : $Y(e^{j\omega}) \ = \ X(e^{j2\omega})$

By N:

- Time domain : $y_n = \begin{cases} x_{\frac{n}{N}} & n = Nk \\ 0 & \text{otherwise} \end{cases}$
- z-domain : $Y(z) \ = \ X(z^N)$
- Fourier domain : $Y(e^{j\omega}) \ = \ X(e^{jN\omega})$

Upsampling: Spectrum



- Expand in time $\rightarrow$ shrink in frequency
- Spectrum shrinks by N
- Lossy or not ?

Upsampling: Operator

By 2:

- Introduce an all-zero row in between every pair of rows of the identity matrix

$$\begin{bmatrix} \vdots \\ y_{-2} \\ y_{-1} \\ \boxed{y_0} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 1 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & \boxed{1} & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 1 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}}_{U_2} \begin{bmatrix} \vdots \\ x_{-1} \\ \boxed{x_0} \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ x_{-1} \\ 0 \\ \boxed{x_0} \\ 0 \\ x_1 \\ 0 \\ x_2 \\ \vdots \end{bmatrix}$$

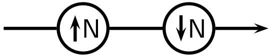
$$y = U_2\, x$$

By N:

- U_N: Introduce (N-1) zero rows in the identity

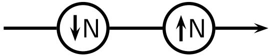
Are downsampling and upsampling friends?

- $U_N = D_N^T$
- $D_N U_N = I$



no loss

- $U_N D_N \neq I$



lossy

- $P = U_2 D_2$

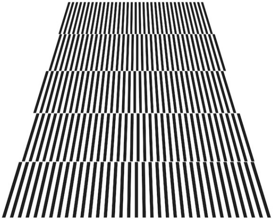
  is an orthogonal projection onto subspace of all even indexed samples

Filtering, downsampling, & upsampling

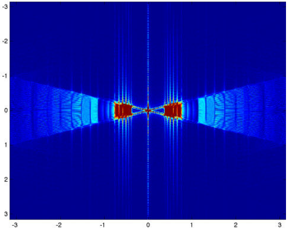
Filtering & downsampling

Should we filter before or after downsampling? Let's consider the following setup:

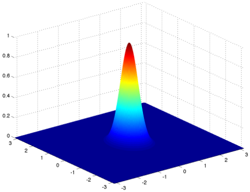
Original



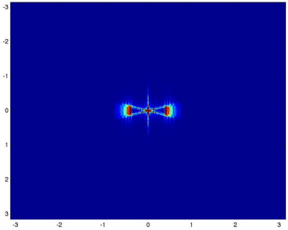
Spectrum original



Low pass filter

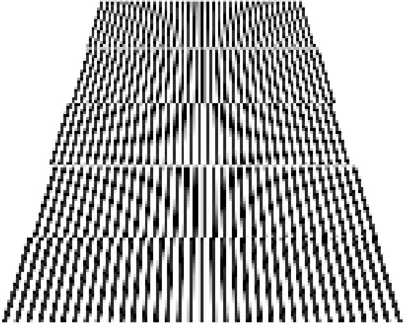


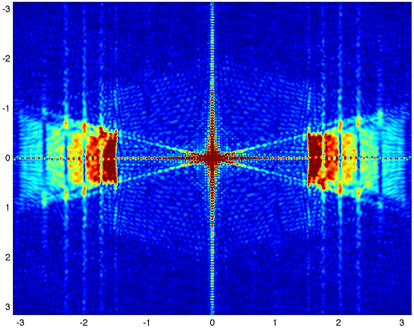
Spectrum original filtered



Filtering & downsampling

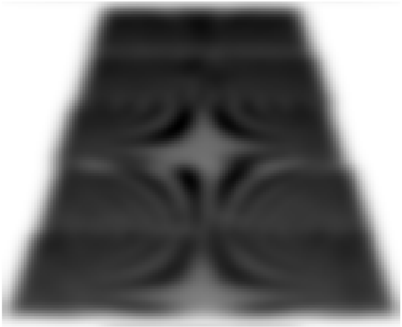
Original downsampled by 4

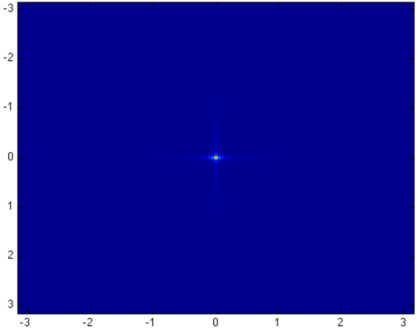




Filtering & downsampling

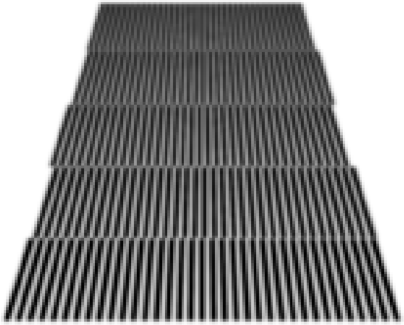
Filtering after downsampling by 4

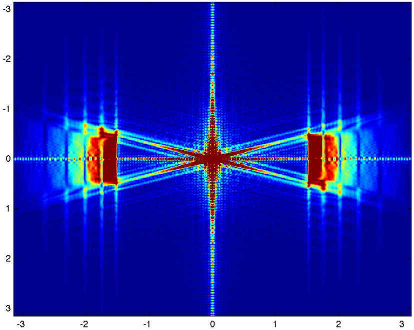




Filtering & downsampling

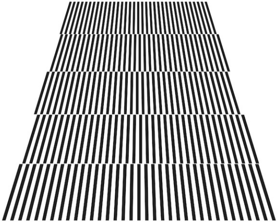
Filtering before downsampling by 4



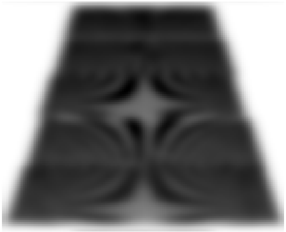


Filtering & downsampling

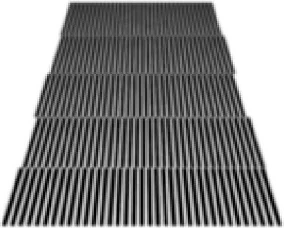
Filtering before or after downsampling?



Original



After



Before

Filtering & downsampling

Filtering before downsampling

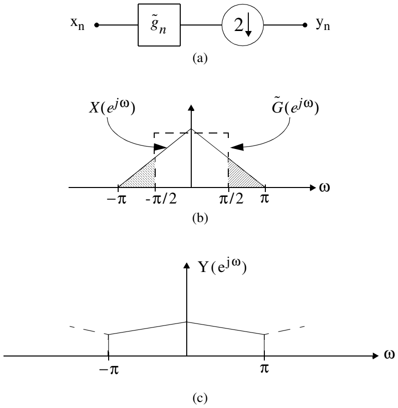
Let's see what happens in mathematical terms (consider a downsampling by 2).

$$y = \begin{bmatrix} \vdots \\ y_{-1} \\ \boxed{y_0} \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & \widetilde{g}_1 & \widetilde{g}_0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & \widetilde{g}_3 & \widetilde{g}_2 & \widetilde{g}_1 & \boxed{\widetilde{g}_0} & 0 & 0 & \cdots \\ \cdots & 0 & 0 & \widetilde{g}_3 & \widetilde{g}_2 & \widetilde{g}_1 & \widetilde{g}_0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & \widetilde{g}_3 & \widetilde{g}_2 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}}_{D_2 \widetilde{G}} \begin{bmatrix} \vdots \\ x_{-3} \\ x_{-2} \\ x_{-1} \\ \boxed{x_0} \\ x_1 \\ x_2 \\ \vdots \end{bmatrix} = D_2 \widetilde{G}\, x.$$

Filtering & downsampling

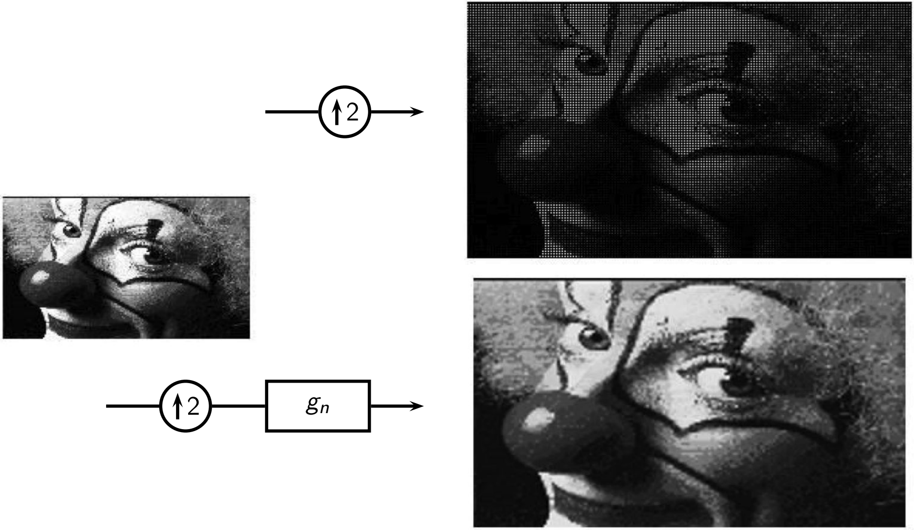
Filtering before downsampling

Low pass filtering enables to reduce bandwidth to $[-\pi/2, \pi/2]$ to avoid aliasing when downsampling



Filtering & upsampling

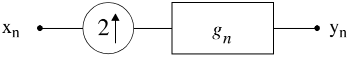
Filtering after upsampling



Filtering & upsampling

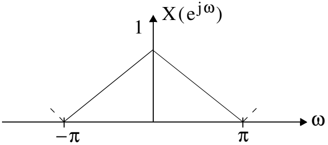
Filtering after upsampling

Filtering after upsampling enables to keep the bandwidth in $[-\pi/2, \pi/2]$



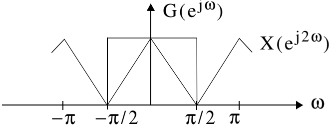
(a)

It amounts to interpolating the sequence y_n so as to fill in the zeros



(b)

$$
y = \begin{bmatrix} \vdots \\ y_{-2} \\ y_{-1} \\ \boxed{y_0} \\ y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \\ \cdots & g_2 & g_0 & 0 & 0 & \cdots \\ \cdots & g_3 & g_1 & 0 & 0 & \cdots \\ \cdots & 0 & g_2 & \boxed{g_0} & 0 & \cdots \\ \cdots & 0 & g_3 & g_1 & 0 & \cdots \\ \cdots & 0 & 0 & g_2 & g_0 & \cdots \\ \cdots & 0 & 0 & g_3 & g_1 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}}_{GU_2} \begin{bmatrix} \vdots \\ x_{-2} \\ x_{-1} \\ \boxed{x_0} \\ x_1 \\ \vdots \end{bmatrix}
$$



(c)

Downsampling, upsampling & filtering

When $\widetilde{g}_n^* = g_{-n}$, $D_2\widetilde{G}$ and GU_2 are adjoint

$$(D_2\widetilde{G})^* = \begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & \widetilde{g}_{-2}^* & \widetilde{g}_0^* & \widetilde{g}_2^* & \widetilde{g}_4^* & \widetilde{g}_6^* & \cdots \\ \cdots & \widetilde{g}_{-3}^* & \widetilde{g}_{-1}^* & \widetilde{g}_1^* & \widetilde{g}_3^* & \widetilde{g}_5^* & \cdots \\ \cdots & \widetilde{g}_{-4}^* & \widetilde{g}_{-2}^* & \boxed{\widetilde{g}_0^*} & \widetilde{g}_2^* & \widetilde{g}_4^* & \cdots \\ \cdots & \widetilde{g}_{-5}^* & \widetilde{g}_{-3}^* & \widetilde{g}_{-1}^* & \widetilde{g}_1^* & \widetilde{g}_3^* & \cdots \\ \cdots & \widetilde{g}_{-6}^* & \widetilde{g}_{-4}^* & \widetilde{g}_{-2}^* & \widetilde{g}_0^* & \widetilde{g}_2^* & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

$$= \begin{bmatrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & g_2 & g_0 & g_{-2} & g_{-4} & g_{-6} & \cdots \\ \cdots & g_3 & g_1 & g_{-1} & g_{-3} & g_{-5} & \cdots \\ \cdots & g_4 & g_2 & \boxed{g_0} & g_{-2} & g_{-4} & \cdots \\ \cdots & g_5 & g_3 & g_1 & g_{-1} & g_{-3} & \cdots \\ \cdots & g_6 & g_4 & g_2 & g_0 & g_{-2} & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} = GU_2$$

Multirate Identities

Filters with impulse responses orthogonal to even shifts

Consider a filter with impulse response orthogonal to its even shifts, *that is,*

$$\langle g_n, g_{n-2k} \rangle_n = \delta_k \,,$$

which corresponds to the deterministic autocorrelation of g downsampled by 2.

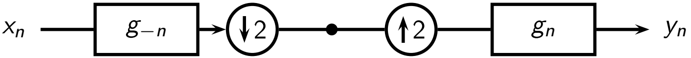
We have

$$D_2 G^* G U_2 = I \,.$$

In particular

$$G U_2 D_2 G^* = P \,,$$

is an orthogonal projection onto the subspace spanned by g_n and its even shifts.



Filters with impulse responses orthogonal to even shifts

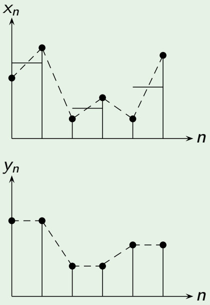
Example (Haar smoothing operator)

- Filter: two point average (normalized to unit norm), or Haar lowpass

$$g_n = \frac{1}{\sqrt{2}}[\delta_n + \delta_{n-1}]$$

- Then

$$
\begin{aligned}
P &= GU_2 \cdot D_2 G^T \\
&= \frac{1}{2}\begin{bmatrix} \ddots & \vdots & \vdots & \ddots \\ \cdots & 1 & 0 & \cdots \\ \cdots & 1 & 0 & \cdots \\ \cdots & 0 & 1 & \cdots \\ \cdots & 0 & 1 & \cdots \\ \ddots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \cdots & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & 1 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \cdots & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & 1 & \cdots \\ \cdots & 0 & 0 & 1 & 1 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}
\end{aligned}
$$



Filters with impulse responses orthogonal to even shifts

Quadrature mirror formula

The orthogonality to even translates $\langle g_n, g_{n-2k}\rangle_n = \delta_k$ gives

$$\frac{1}{2}\Big[G(z^{1/2})G(z^{-1/2}) + G(-z^{1/2})G(-z^{-1/2})\Big] = 1\,,$$

that in the frequency domain reads

$$|G(e^{j\omega})|^2 + |G(e^{j(\omega+\pi)})|^2 = 2\,,$$

the latter being called the *quadrature mirror formula* or *power complementarity*

Filters with impulse responses orthogonal to even shifts

Quadrature mirror formula

Let's see where it comes from ...

The orthogonality to even translates $\langle g_n, g_{n-2k} \rangle_n = \delta_k$ can be seen as a deterministic autocorrelation of g

$$a_k = \langle g_n, g_{n-k} \rangle_n \,,$$

downsampled by 2, *that is,* , $a_{2k} = \delta_k$.

Assuming now a real g, we have

$$A(z) = G(z)G(z^{-1}) \,.$$

Keeping only the even terms can be accomplished by adding $A(z)$ and $A(-z)$ and dividing by 2:

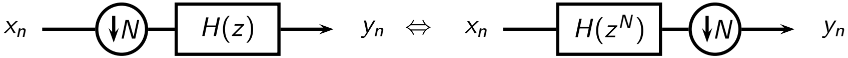
$$\frac{1}{2}(A(z) + A(-z)) \;=\; \frac{1}{2}(G(z)G(z^{-1}) + G(-z)G(-z^{-1})) \;=\; 1 \,,$$

which on the unit circle leads to

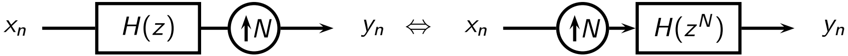
$$|G(e^{j\omega})|^2 + |G(e^{j(\omega+\pi)})|^2 = 2$$

Interchange of Multirate Operations and Filtering

- Interchange of filtering and downsampling

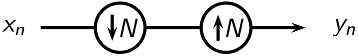


- Interchange of filtering and upsampling

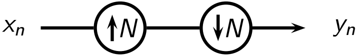


Commutativity of Multirate Operations

- Commutativity

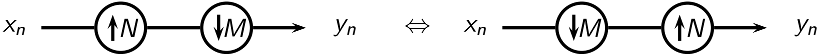


$U_N D_N = P$ projection



$D_N U_N = I$ identity

Upsampling and downsampling for the same integer do not commute.



Upsampling and downsampling for the different integers N and M, respectively, do commute when $gcd(N, M) = 1$.

Polyphase representations

Polyphase representation

Multirate processing brings a major twist to signal processing: shift invariance is replaced by periodic shift variance, represented by block-Toeplitz matrices.

The polyphase representation is a tool dealing with such periodic shift variance. It is a key method to transform single-input single-output linear periodically shift varying systems into multiple-input multiple-output linear shift invariant systems.

Polyphase representation: $N = 2$

Forward and inverse polyphase transforms

- Forward polyphase transform

  Decompose the signal into odd and even parts

$$X(z) = \sum_{n\in\mathbb{Z}} x_n z^{-n} = \sum_{n\in\mathbb{Z}} x_{2n} z^{-2n} + \sum_{n\in\mathbb{Z}} x_{2n+1} z^{-2n-1} = X_0(z^2) + z^{-1} X_1(z^2)\,,$$

  where

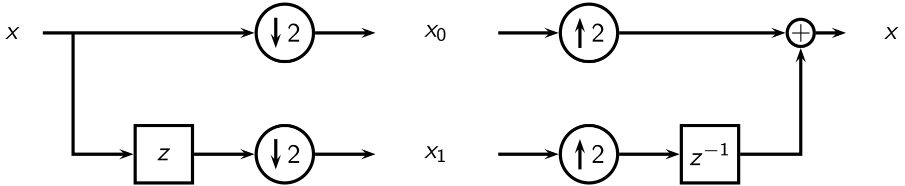
$$X_0(z) = \sum_{n\in\mathbb{Z}} x_{2n} z^{-n}, X_1(z) = \sum_{n\in\mathbb{Z}} x_{2n+1} z^{-n}\,.$$

- Inverse polyphase transform

  Take polyphase components and reconstruct X

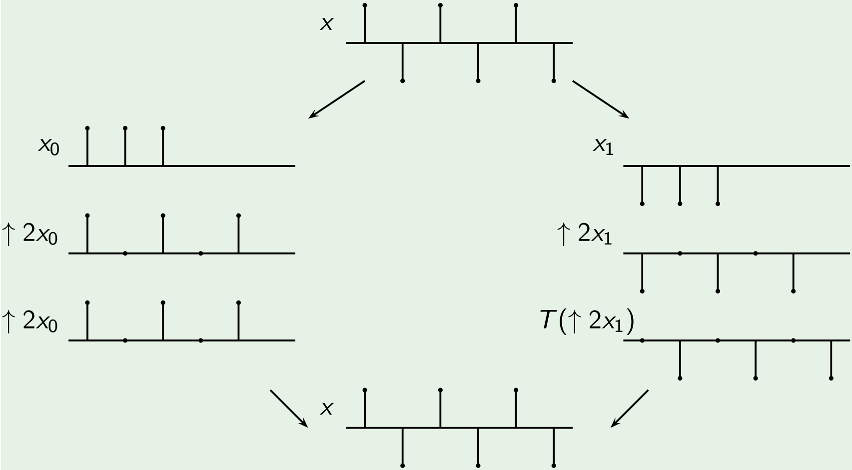
Polyphase representation: $N = 2$

Forward and inverse polyphase transforms



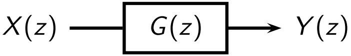
Polyphase representation: $N = 2$

Example

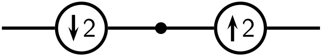


Filtering in Polyphase Domain

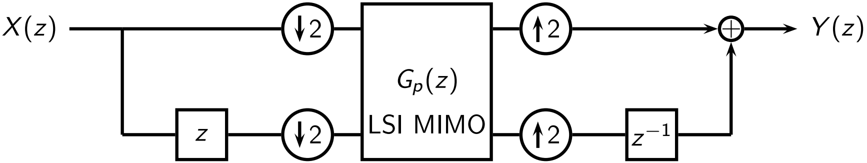
- Linear Shit-Invariant - LSI SISO systems



- Multirate leads to Linear Periodically Shift-Varying - LPSV SISO systems



- Polyphase takes SISO linear periodically shift varying system to MIMO linear shift invariant system



Filtering in Polyphase Domain

Given a filter (LSI system) with impulse response g_n, we have

$$g_{0,n} = g_{2n} \qquad G_0(z) = \sum_{n\in\mathbb{Z}} g_{2n} z^{-n}\,,$$

$$g_{1,n} = g_{2n+1} \qquad G_1(z) = \sum_{n\in\mathbb{Z}} g_{2n+1} z^{-n}\,,$$

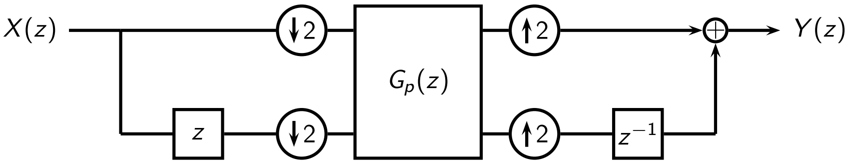
obtaining

$$G(z) = G_0(z^2) + z^{-1} G_1(z^2)\,.$$

Such a decomposition can be described vie the polyphase representation, mapping $G(z)$ into

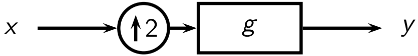
$$G_p(z) = \begin{bmatrix} G_0(z) & z^{-1}G_1(z) \\ G_1(z) & G_0(z) \end{bmatrix}$$

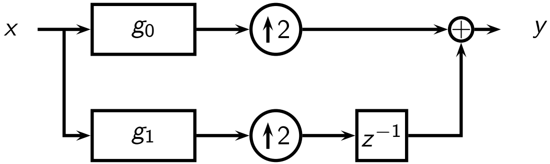
Filtering in Polyphase Domain



$$
\begin{aligned}
Y(z) &= \begin{bmatrix} 1 & z^{-1} \end{bmatrix} G_p(z^2) \begin{bmatrix} X_0(z^2) \\ X_1(z^2) \end{bmatrix} \\
&= \begin{bmatrix} 1 & z^{-1} \end{bmatrix} \begin{bmatrix} G_0(z^2) & z^{-2} G_1(z^2) \\ G_1(z^2) & G_0(z^2) \end{bmatrix} \begin{bmatrix} X_0(z^2) \\ X_1(z^2) \end{bmatrix} \\
&= \begin{bmatrix} 1 & z^{-1} \end{bmatrix} \begin{bmatrix} G_0(z^2) X_0(z^2) + z^{-2} G_1(z^2) X_1(z^2) \\ G_1(z^2) X_0(z^2) + G_0(z^2) X_1(z^2) \end{bmatrix} \\
&= G_0(z^2) X_0(z^2) + z^{-2} G_1(z^2) X_1(z^2) + z^{-1} G_1(z^2) X_0(z^2) + z^{-1} G_0(z^2) X_1(z^2) \\
&= [G_0(z^2) + z^{-1} G_1(z^2)][X_0(z^2) + z^{-1} X_1(z^2)] = G(z) X(z) \,.
\end{aligned}
$$

Polyphase representation of upsampling + filtering



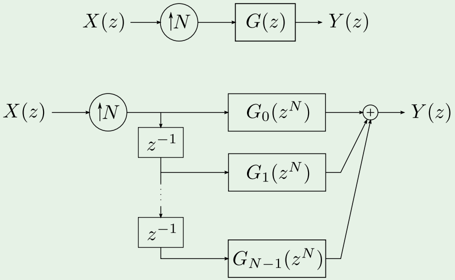


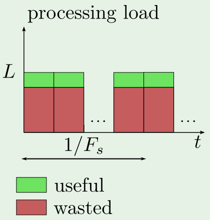
$$Y(z) \;=\; G_0(z^2)X(z^2) + z^{-1}G_1(z^2)X(z^2) \;=\; Y_0(z^2) + z^{-1}Y_1(z^2)$$

Polyphase representation of filtering + upsampling by N

Polyphase implementation of upsampling FIR filters

- Upsampling followed by filtering results in many multiplications by zero ($N-1$ out of N filter inputs are zero)

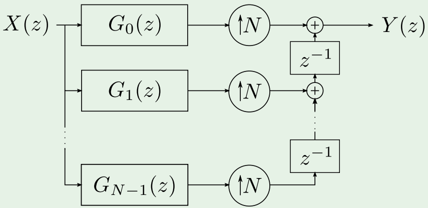


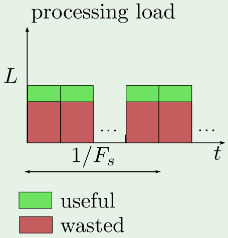


Polyphase representation of filtering + upsampling by N

Polyphase implementation of upsampling FIR filters

- Interchanging filtering and downsampling moves the processing to the lower rate F_s (as opposed to NF_s)
- Naive implementation
  - Compute the outputs of all polyphase filters
  - Add delayed upsampled outputs of all polyphase filters
  - Processing load decreases, but still adding many zeros (every output sampled obtained from only one non-zero usamplers' outputs)

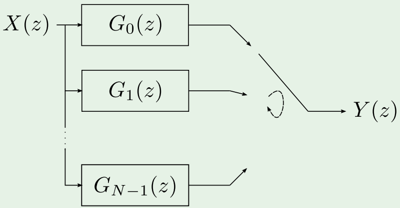


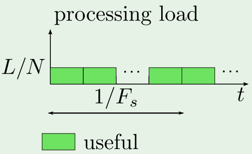


Polyphase representation of filtering + upsampling by N

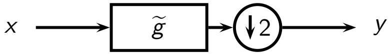
Polyphase implementation of upsampling FIR filters

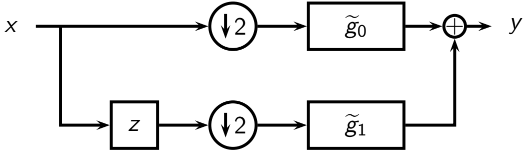
- For every output sample, only one polyphase component is computed
- Filter delay lines loaded with a new input sample every N output samples
- The processing load decreased by a factor of N





Polyphase representation of filtering + downsampling





$$\widetilde{g}_{0,n} = \widetilde{g}_{2n} \quad \stackrel{\text{ZT}}{\longleftrightarrow} \quad \widetilde{G}_0(z) = \sum_{n\in\mathbb{Z}} \widetilde{g}_{2n} z^{-n},$$

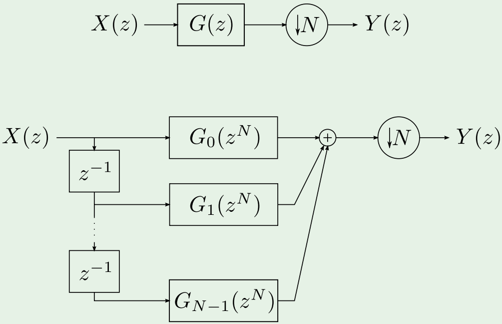
$$\widetilde{g}_{1,n} = \widetilde{g}_{2n-1} \quad \stackrel{\text{ZT}}{\longleftrightarrow} \quad \widetilde{G}_1(z) = \sum_{n\in\mathbb{Z}} \widetilde{g}_{2n-1} z^{-n},$$

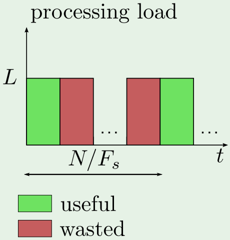
$$\widetilde{G}(z) = \widetilde{G}_0(z^2) + z\widetilde{G}_1(z^2).$$

Polyphase representation of filtering + downsampling by N

Polyphase implementation of downsampling FIR filters

- Filtering followed by downsampling discards $N-1$ every N samples and wastes processing time

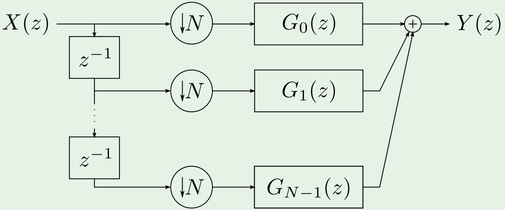


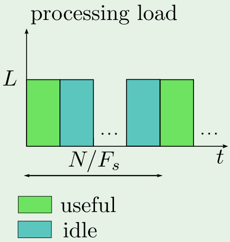


Polyphase representation of filtering + downsampling by N

Polyphase implementation of downsampling FIR filters

- Interchanging filtering and downsampling moves the processing to the lower rate F_s/N
- Naive implementation
  - Pre-load filters
  - Compute the output sample after all the polyphase filters have been loaded (every N input samples)
  - Processing load decreased, but done in the same bursts every N samples

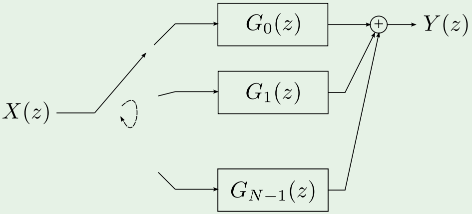


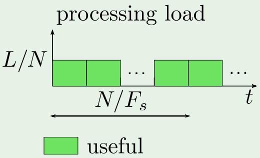


Polyphase representation of filtering + downsampling by N

Polyphase implementation of downsampling FIR filters

- Commutator runs at F_s
- One polyphase component computed per input sample and stored in accumulator
- After completing the last polyphase component, the output is emitted and accumulator reset
- One reduces the processing load by a factor of N without having idle cycles





Polyphase representation: any N

- Take indices modulo N
- N polyphase components
- $X(z) = \sum_{i=0}^{N-1} z^{-i} X_i(z^N)$ with

  $$X_i(z) = \sum_{n \in \mathbb{Z}} x_{Nn+i} z^{-n} \quad i = 0, \ldots, N-1$$