NTNU – Trondheim
Norwegian University of Science and Technology

TTT4120 Digital Signal Processing Fall 2017

Multirate signal processing

Prof. Stefan Werner
stefan.werner@ntnu.no
Office B329

Lecture in course book*

- Proakis, Manolakis Digital Signal Processing, 4th Ed.
  - 11.1 Introduction
  - 11.2 Decimation by a factor D
  - 11.3 Interpolation by a factor I
  - 11.4 Sampling rate conversion by a rational factor I/D
  - 11.6 Multistage implementation of sampling rate conversion

A compressed overview of topics treated in the lecture, see "Flerhastighetssystemer" on Blackboard

*Level of detail is defined by lectures and problem sets

2

Contents and learning outcomes

- Multirate signal processing and sampling rate conversion
- Decimation by a factor *D*
- Interpolation by a factor *I*
- Rate conversion with a rational factor
- Multistage implementations

3

Multirate processing and rate conversion

- A multirate system is a digital system that operates on two or more sampling frequencies (or rates)
  - Interface between systems of different rates
  - Efficient realiztions of filter banks
  - Efficient realization of filters with sharp transition bands
  - Oversampled A/D- and D/A-converters

4

Multirate processing and rate conversion

- Two approaches to change sampling rate of a discrete-time signal

1. Reconstruct analog signal and resample at different rate

Original discrete signal: $x[n] = x_a(nT_1)$

Reconstructed analog signal:

$$x_a(t) = \sum_k x_a(nT_1) \frac{\sin[\pi(t-kT_1)/T_1]}{[\pi(t-kT_1)/T_1]}$$

Resampled analog signal:

$$x_a(nT_2) = \sum_k x_a(nT_1) \frac{\sin[\pi(nT_2-kT_1)/T_1]}{[\pi(nT_2-kT_1)/T_1]}$$

5

Multirate processing and rate conversion

2. Directly in digital domain:

$$x[n] \longrightarrow \boxed{\text{Discrete} - \text{time system}} \longrightarrow y[n]$$

Rate: $F_x = \frac{1}{T_x}$ Rate: $F_y = \frac{1}{T_y}$

- Avoids distortion due to non-ideal A/D- and D/A-conversion

6

Multirate processing and rate conversion

- Systems that change sampling rate
- System operating at different sampling rates

- Interpolation – increasing sampling rate
- Decimation – decreasing sampling rate

Decimation by a factor D

- Decimation – decreasing sampling rate
- Downsampling of highrate signal $x[n]$ into lowrate signal $y[m]$

$$x[n] \longrightarrow \boxed{\downarrow D} \longrightarrow y[m]$$

Rate: $F_x = \dfrac{1}{T_x}$ Rate: $F_y = \dfrac{F_x}{D} = \dfrac{1}{DT_x}$

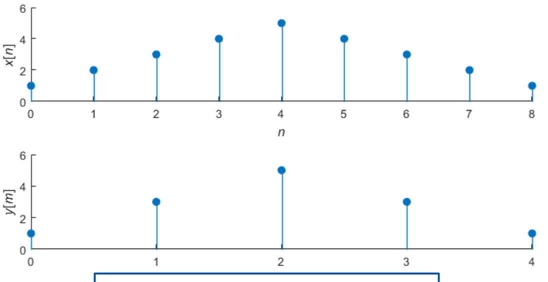
- Downsampled signal is obtained by selecting one out of D samples $x[n]$ and throwing away the other $(D-1)$ samples

$$y[m] = x[n]|_{n=mD} = x[mD], \quad n, m, D \in \{\text{integers}\}$$

Decimation by a factor D…

- Example: Using $D = 2$ and $x[n] = \{\underline{1}, 2, 3, 4, 5, 4, 3, 2, 1\}$

$$y[m] = x[mD] = \{\underline{1}, 3, 5, 3, 1\}$$



Matlab

```
x  = [1,2,3,4,5,4,3,2,1];
y  = downsample(x,2);
```

9

Decimation by a factor D…

- What happens in frequency domain?
- Relate the spectrum of downsampled signal $Y(f)$ to original spectrum $X(f)$

$$X(f) = F_x \sum_{k=-\infty}^{\infty} X_a([f-k]F_x) \text{ (Lecture 10)}$$

- I want to have the following spectrum

$$Y(f) = F_y \sum_{k=-\infty}^{\infty} X_a([f-k]F_y)$$

$$= \frac{F_x}{D} \sum_{k=-\infty}^{\infty} X_a\left(\left[\frac{f}{D} - \frac{k}{D}\right] F_x\right)$$

$$= \frac{1}{D} \sum_{i=0}^{D-1} F_x \sum_{k=-\infty}^{\infty} X_a\left(\left[\frac{f}{D} - \frac{(i+kD)}{D}\right] F_x\right)$$

$$= \frac{1}{D} \sum_{i=0}^{D-1} X\left(\frac{f}{D} - \frac{i}{D}\right)$$

10

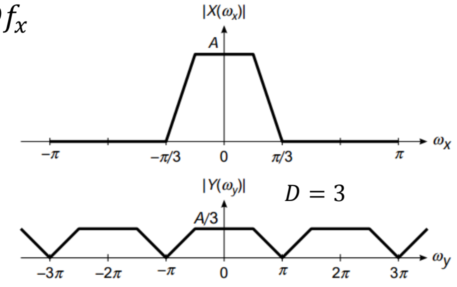
Decimation by a factor D…

- Spectrum of downsampled signal $Y(f)$ related to the original spectrum $X(f)$ by D scaled and shifted copies

$$Y(f) = \frac{1}{D}\sum_{i=0}^{D-1} X\left(\frac{f}{D} - \frac{i}{D}\right)$$

- Normalized frequency variables are related as

$$f_y = \frac{F}{F_y} = \frac{F}{F_x/D} = Df_x$$



11

Decimation by a factor D…

- Must avoid that downsampling causes aliasing
- Bandlimit the original signal related to $x[n]$ to $F_{x,\max} = F_x/2D$

1. Lowpass-filter signal with

$$H_D(f_x) = \begin{cases} 1, & |f_x| \leq 1/2D \\ 0, & \text{otherwise} \end{cases}$$

$$v[n] = \sum_{k=0}^{\infty} x[k]h_D[n-k]$$

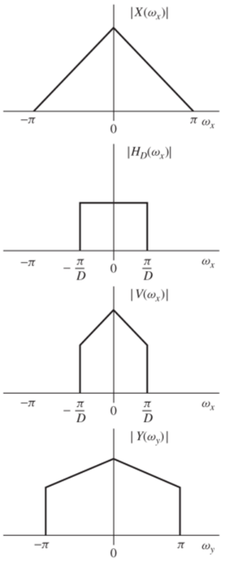
2. Decimation by a factor D

$$y[m] = v[Dm] = \sum_{k=0}^{\infty} x[k]h_D[Dm-k]$$

12

Decimation by a factor D…

- Example:



13

Decimation by a factor D…

$$x[n] \longrightarrow \boxed{h_D[n]} \longrightarrow \boxed{\downarrow D} \longrightarrow y[m]$$

- Filtering-view of decimation

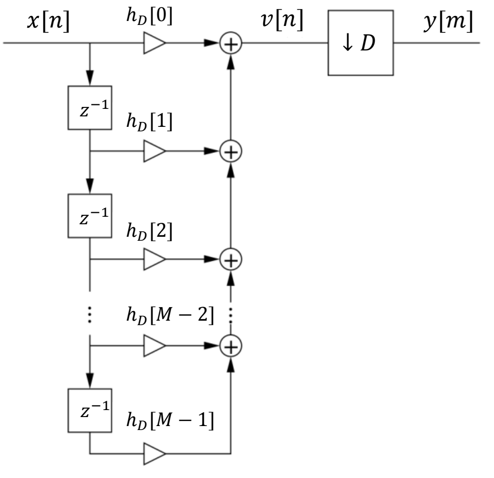
$$y[m] = v[Dm] = \sum_{k=0}^{\infty} x[k] h_D[Dm-k]$$

- The whole decimation process can be performed directly on $x[n]$
- Note that D new values of $x[n]$ are used for each output $y[m]$

14

Decimation by a factor D…

- Direct-form realization

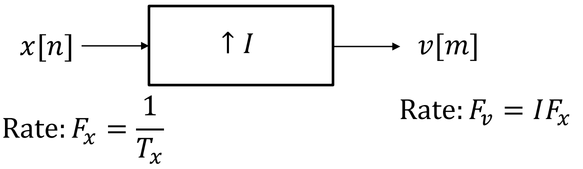


Decimation by a factor D…

- Some practical issues
  - Information is lost during decimation and filtering
  - Tradeoffs between efficiency (computational complexity) and information content
  - Tradeoff between bitrate and information content

Interpolation by a factor I

- Interpolation – increasing sampling rate
  - Interpolate $(I-1)$ new samples between successive samples
- Upsample $x[n]$ into sequence $v[m]$



- $v[m]$ obtained by adding $(I-1)$ zeros between samples of $x[n]$

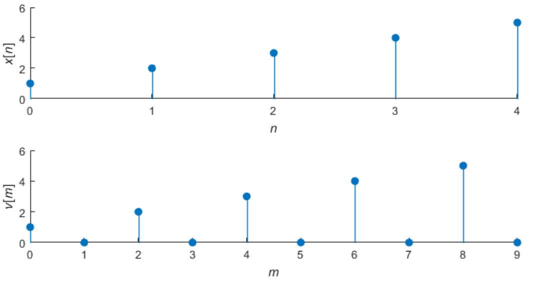
$$v[m] = \begin{cases} x\left[\frac{m}{I}\right], & m = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases}$$

17

Interpolation by a factor I…

- Example: Using $I = 2$ and $x[n] = \{\underline{1}, 2, 3, 4, 5\}$

$$v[m] = \{\underline{1}, 0, 2, 0, 3, 0, 4, 0, 5\}$$



Matlab

```
x  = [1,2,3,4,5];
y  = upsample(x,2);
```

18

Interpolation by a factor I…

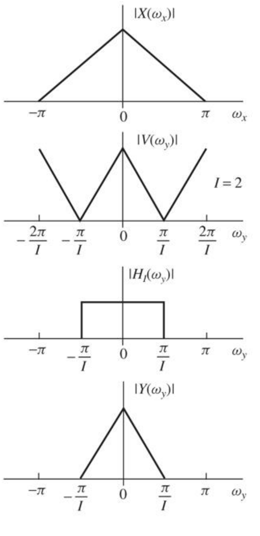
- Given $x[n] = x_a(nT_x)$ and $v[m]$: how to obtain $y[m] = x_a(mT_y)$?
- In frequency domain, we have the relation

$$V(\omega) = \sum_{n=-\infty}^{\infty} v[n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega nI}$$
$$= X(\omega I)$$

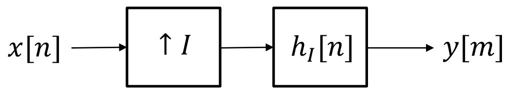
- Thus we just need to pass $v[n]$ through a lowpass filter

$$H_I(f_y) = \begin{cases} 1, & |f_y| \leq 1/2I \\ 0, & \text{otherwise} \end{cases}$$

Interpolation by a factor I…



Interpolation by a factor I…



- Filtering view of interpolation

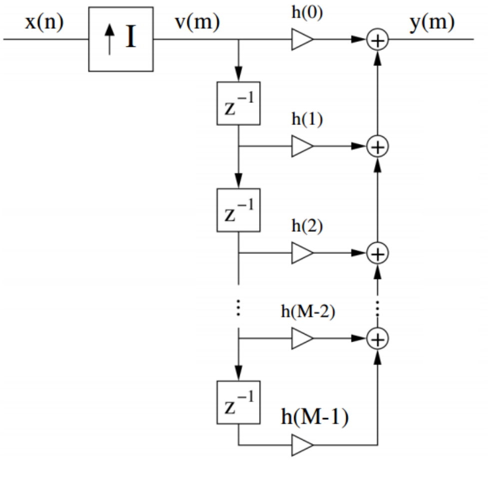
$$y[m] = \sum_{k=0}^{\infty} x[k] h_I[m - kI]$$

- The whole decimation process can be performed directly on $x[n]$

21

Interpolation by a factor I…

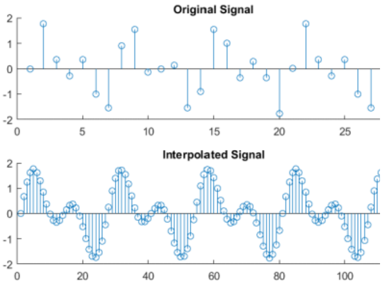
- Direct-form realization

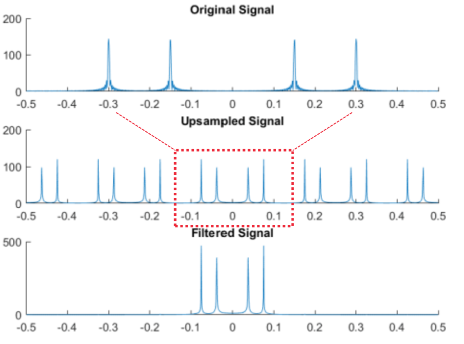


22

Interpolation by a factor *I*…

- Example: Signal $x(t) = \sin(2\pi \cdot 30t) + \sin(2\pi \cdot 60t)$ is samples at $F_s = 200$ Hz resulting in $x[n]$. Interpolate sequence $x[n]$ to obtain $x_a(nt/800)$





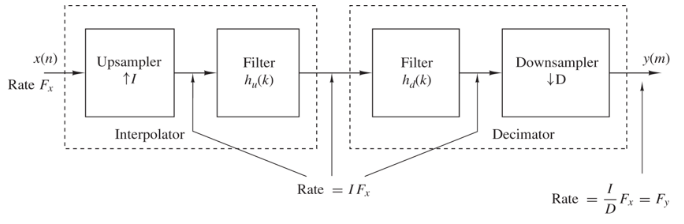
Interpolation by a factor *I*…

Matlab

```
t = 0:0.001:.029; Nfft = 1024;

x = sin(2*pi*30*t) + sin(2*pi*60*t);
y = interp(x,4);
subplot(211); stem(x);
subplot(212); stem(y);

K = (-Nfft/2:Nfft/2-1)/Nfft;
X = fftshift(fft(x,Nfft));
V = fftshift(fft(upsample(x,4),Nfft));
Y = fftshift(fft(y,Nfft));

figure,
subplot(311), plot(K,abs(X)),
title('Original Signal');
subplot(312), plot(K,abs(V)),
title('Upsampled Signal');
subplot(313), plot(K,abs(Y)),
title('Filtered Signal');
```

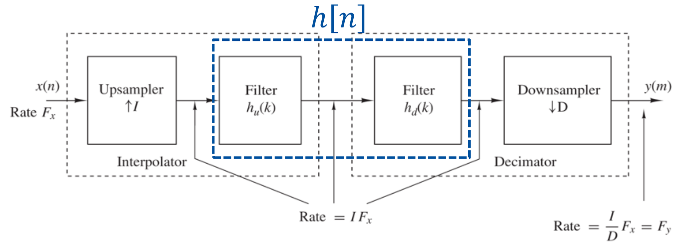
Rate conversion by a rational factor

- Treated special cases:
  - Decimation (downsampling) by a factor D
  - Interpolation (upsampling) by a factor I
- What if we would like to change the rate from 48kHz to 32kHz?
- Combine interpolation and decimation



25

Rate conversion by a rational factor…



- Frequency response of the combined filter $h[n]$

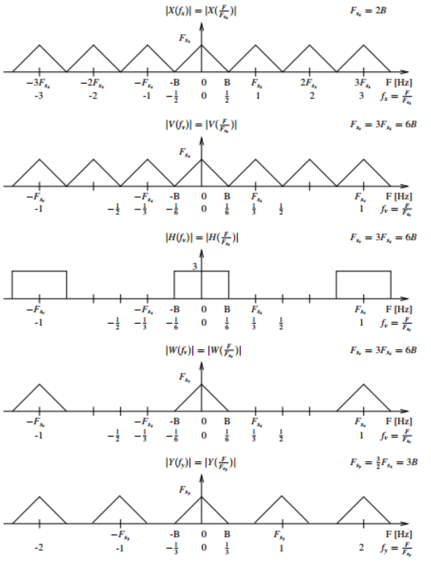
$$H(f_v) = \begin{cases} 1, & |f_v| \leq \frac{1}{2\max(I,D)} \\ 0, & \text{otherwise} \end{cases}$$

26

Fixed-point representation…

- Rate conversion:

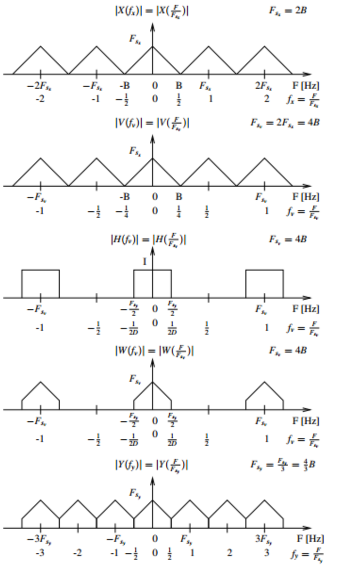
  $I/D = 3/2$



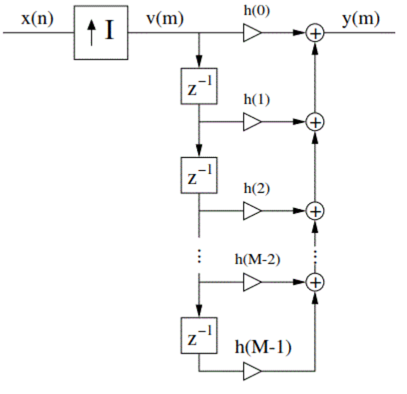
Fixed-point representation…

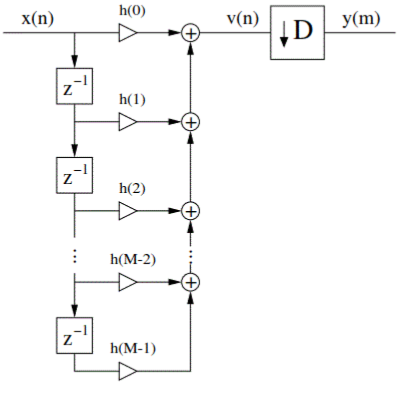
- Rate conversion:

  $I/D = 2/3$



Efficient implementation structures

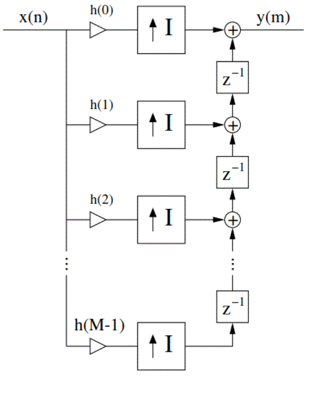


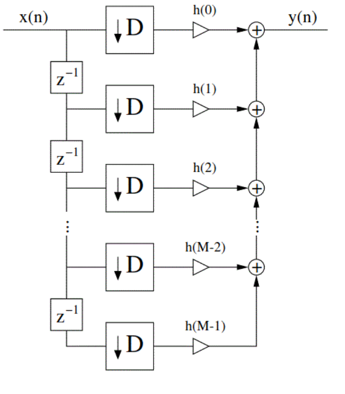


- Interpolation filter: only every Ith sample non-zero
- Decimation filter: only every Dth sample used

29

Efficient implementation structures…



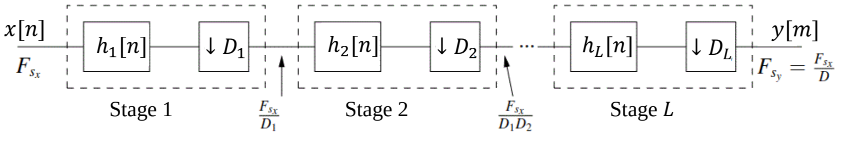


- Interpolation filter: multiplications with non-zero samples only
- Decimation filter: multiplications with used samples only

30

Multistage implementation

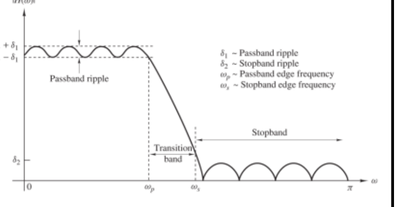
- Large interpolation- or decimation factors give stringent filter speciation
- Can be avoided by using multistage implementation
- Example: Decimation with $D = D_1 \cdot D_2 \cdots D_L$ implemented as



- Filter length can be reduced due to relaxed requirements on the width of transition region
- Note that passband ripple must be reduced by a factor of L

31

Multistage implementation…



- Subband coding
  - Filterbank of BP filters
  - Critical sampling in each band
- Audioband signal at $F_s = 8000$ Hz
- Isolate frequency components below 80 Hz with a filter that has passband, $0 - 75$ Hz
  - $f_p = 75/8000$, $f_s = 80/8000$
- Ripple specification: $\delta_1 = 10^{-2}$, $\delta_2 = 10^{-4}$
- Filter order (firpm): 5022

  $\Rightarrow 5022 \cdot 8000 \approx 40.176 \cdot 10^6$ mult/sample
- Instead use two-stage decimation: $D_1 = 25$ and $D_2 = 2$

32

Multistage implementation…

Requirements for two-stage implementation

- Stage 1: $F_s = 8000/25 = 320$ Hz
  - $f_{p1} = 75/8000$ Hz
  - $f_{p2} = (320 - 80)/320$ Hz (allow aliasing in band that will be filtered away)
  - $\delta_{11} = \frac{\delta_1}{2} = 0.5 \cdot 10^{-2}$, $\delta_{21} = 10^{-4}$
  - Filter order (firpm): 164
- Stage 2: $F_s = 320/2 = 160$ Hz
  - $f_{p1} = 75/320$ Hz, $f_{p2} = 80/320$ Hz
  - $\delta_{11} = \frac{\delta_1}{2} = 0.5 \cdot 10^{-2}$, $\delta_{21} = 10^{-4}$
  - Filter order (firpm): 216

Multistage implementation…

- Total amount of multiplications

$$164 \cdot 8000 + 216 \cdot 320 = 1.381 \cdot 10^6 \text{ mult/s}$$

$\Longrightarrow$ Less than 4% of the full rate-solution

Summary

- Today we discussed:
  - Multirate
- Next:
  - Exam