

## Multi-Rate Signal Processing

Digital Signal Processing

Notes

---

---

---

---

---

---

---

## Contents

Introduction

Decimation

Interpolation

Non-Integer Sample Rate Conversion

Multistage Sample Rate Conversion

Notes

---

---

---

---

---

---

---

## What is a Multirate Digital Signal Processing?

- A **digital signal processing system** that uses signals with **different sampling frequencies** is probably performing **multirate digital signal processing**.
- **Multirate digital signal processing** often uses **sample rate conversion** to convert from **one sampling frequency to another** *sampling frequency*.
- **Sample rate conversion** uses **decimation** to **decrease** the sampling rate, **interpolation** to **increase** the sampling rate.

Notes

---

---

---

---

---

---

---

# Sample Rate Conversion

Changing the **sampling frequency** in the **analog domain** requires:

- **digital to analog** conversion then
- **analog to digital** conversion at a **different sampling frequency**.

Both

- **Digital to analog conversion**
- **Analog to digital conversion**

introduce **errors** and **noise** into the signal.  
Therefore **sample rate conversion** is done in **digital domain** and uses a combination of:

- **Decimation,**
- **and Interpolation.**

Notes

---

---

---

---

---

---

---

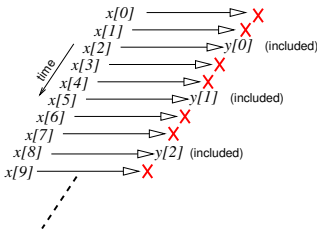
---

## Decimation (for Downsampling)

- **Decimation** removes samples from a signal.
- **Decimation** can therefore only **downsample** the signal by an **integer factor**:

$$\frac{f_s}{f_s^{new}} = D > 1 \quad \text{so that } f_s > f_s^{new}$$

where  $D$  is an integer,  $f_s$  is the old sampling rate (number of samples per second) and  $f_s^{new}$  is the new sampling rate.



Notes

---

---

---

---

---

---

---

---

## Anti-aliasing for Decimation

**Decimation** decreases the sampling rate.

- The **sampling theorem** states that the highest frequency in a signal should be less than **half the sampling frequency**.
- A digital **anti-aliasing filter** has to be applied to **remove frequencies** higher than:

$$f_{cf} = \frac{f_s^{new}}{2}$$

- So in digital frequency the cut-off frequency is:

$$\Omega_{cf} = \frac{\Omega_s^{new}}{2} = \frac{2\pi \frac{f_s^{new}}{f_s}}{2} = \pi \frac{f_s^{new}}{f_s} < \pi$$

as  $f_s^{new} < f_s$ .

Notes

---

---

---

---

---

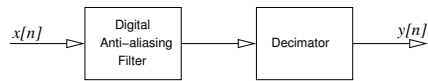
---

---

---

# Anti-aliasing for Decimation

This means that the signal has to be **filtered** in the **digital domain** before **decimation**:



Notes

---

---

---

---

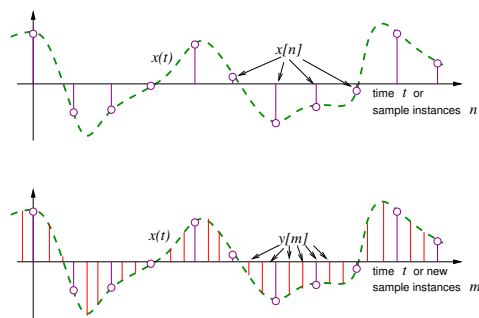
---

---

---

# Interpolation (for Upsampling)

- **Interpolation** increases the sampling frequency by estimating the value of the signal between samples.



Notes

---

---

---

---

---

---

---

# Interpolation

- The **new sampling frequency** is **greater** than the **old sampling frequency**:

$$f_s^{new} > f_s$$

where  $f_s$  is the old sampling frequency and  $f_s^{new}$  the new sampling frequency.

- Also, the new sampling frequency has to be an **integer multiple** of the original sampling frequency:

$$\frac{f_s^{new}}{f_s} = D > 1$$

where  $D$  is an **integer**.

Notes

---

---

---

---

---

---

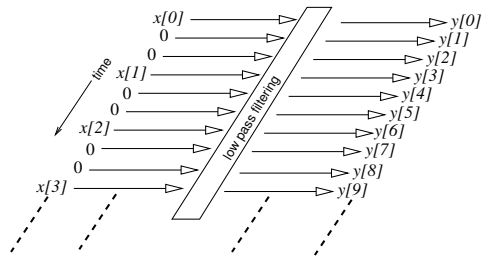
---

## Zero Filling Based Interpolation

A common interpolation approach is **zero filling based interpolation**.

There are two stages:

1. **zero filling**
2. **low pass filtering**



Notes

---

---

---

---

---

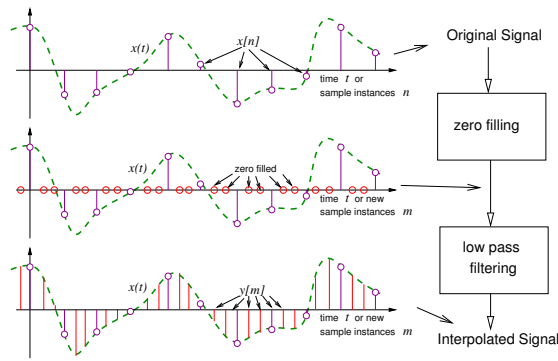
---

---

---

## Zero Filling Based Interpolation

**Example:** Interpolating by  $\times 3$  (two zero samples are inserted between each original sample).



Notes

---

---

---

---

---

---

---

---

## Non-Integer Sample Rate Conversion

Both:

- **Decimation** (for downsampling):

$$\frac{f_s}{f_s^{\text{new}}} = D$$

- **and Interpolation** (for upsampling):

$$\frac{f_s^{\text{new}}}{f_s} = D$$

where  $D$  is an integer, can only **change** the **sampling frequency** to an **integer** of the **original frequency**.

Notes

---

---

---

---

---

---

---

---

# Non-Integer Sample Rate Conversion

Example:

- A CD player stores music at 44.1kHz.
  - A professional music recording device processes audio at 48kHz.
  - Transfer of the music to or from the CD player and the professional audio device using:
    - decimation only or
    - interpolation only
- are **not possible** because:

$$\frac{48e3}{44.1e3} = 1.0884$$

which is not an integer.

Notes

---

---

---

---

---

---

---

# Non-Integer Sample Rate Conversion

Solution!

Combine decimation and interpolation to get **non-integer sample rate conversion**.  
Similar to finding a **common denominator** in fractions...

1. Find **common (integer) factor** of the two sample rates,  $L$
2. **Interpolate (upsample)** by this **common factor**  $L$
3. **Decimate (downsample)** to the **new sample rate**  $f_s^{new}$  by downsampling by an integer factor  $M$ .

The sample rate conversion is then:

$$\frac{L}{M} = \frac{f_s^{new}}{f_s}$$

Notes

---

---

---

---

---

---

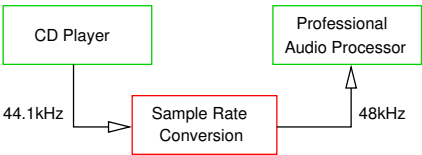
---

# Non-Integer Sample Rate Conversion

Example

Example:

Get audio from 44.1kHz sampled source (CD player) and transfer to professional audio processor requiring 48kHz sample rate.  
**A.** This process requires upsampling to 48kHz from 44.1kHz.



Notes

---

---

---

---

---

---

---

# Non-Integer Sample Rate Conversion

Example cont'd.

- 1. **Worst case common factor:**  $L = 48\text{kHz}$  to give  $f_s \times 48\text{kHz} = 2116.8\text{MHz}$ .  
**Better alternative is**  $L = 160$  to give  $L \times 44.1\text{kHz} = 7056\text{kHz}$
- 2. So **interpolate** by factor  $L$  by **inserting 159 zeros** for each sample in 44.1kHz CD player signal then **low pass filtering**.
- 3. Then **decimate** to 48kHz by **removing 146 samples** in every 147 ( $= L \times 44.1\text{kHz}/48\text{kHz}$ ) from the **upsampled signal** (after applying **anti-aliasing low pass filter**).

The resulting sample rate conversion is:

$$\frac{L}{M} = \frac{160}{147} = 1.088$$

which is the same as

$$\frac{f_s^{\text{new}}}{f_s} = \frac{48\text{kHz}}{44.1\text{kHz}} = 1.088.$$

Notes

---

---

---

---

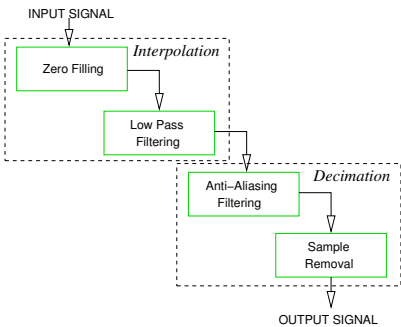
---

---

---

# Optimising Non-Integer Sample Rate Conversion

There are  $\times 2$  **low pass filters** (low pass filtering and anti-aliasing filtering) for **non-integer sample rate conversion**:



Notes

---

---

---

---

---

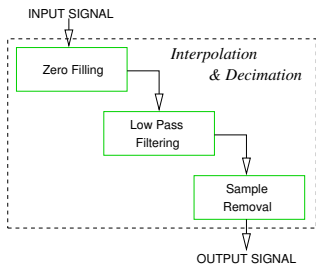
---

---

# Optimising Non-Integer Sample Rate Conversion

The **interpolation low pass filter** and the **anti-aliasing filter** for the **decimation** stage can be **combined**

with a cut-off frequency equal to the lower of the two filters' cut-off frequencies.



Notes

---

---

---

---

---

---

---

Multistage Sample Rate Conversion

Problem!

In real world applications sample rate conversion converts a **sampling frequency** to **another sampling frequency** that is:

- Very much greater ( $f_s^{new} \gg f_s$ ) or
  - Very much smaller ( $f_s^{new} \ll f_s$ )
- than the **original signal sampling frequency**.

But what is wrong with this?

This is best explained by an example.

Notes

Multistage Sample Rate Conversion: Problem

**Q.** A signal  $x[n]$ , sampled at 4.096kHz has to be decimated to 128Hz. There should be an antialiasing filter:

- that rejects frequencies above 64Hz,
- with a stopband ripple,  $\delta_s \approx 0.001$ ,
- and a passband ripple of  $\delta_p \approx 0.001$ .
- The transition width should be  $f_{tw} = 4\text{Hz}$ ,
- so that frequencies below 60Hz are kept.

**A.** A Blackman window can achieve a stop band ripple 75dB and passband ripple of 0.0014dB.

This can be compared with the requirements of this antialiasing filter of  $\delta_s \approx 0.001$ , which is  $-20 \log(0.001) = 60\text{dB}$  and a passband ripple  $\delta_p \approx 0.001$ , or  $20 \log(1 + 0.001) = 0.0087\text{dB}$ .

However, according to the low pass FIR filter design guidelines in [Van de Vegte, 2002]<sup>1</sup>, the number of filter coefficients for a Blackman window will then be:

$$N = 5.98 \times \frac{f_s}{f_{tw}} = 5.98 \times 4096/4 = 6123.5$$

So the number of filter coefficients is very high.

<sup>1</sup>Van de Vegte, "Fundamentals of Digital Signal Processing" Prentice Hall, 2002.

Notes

Multistage Sample Rate Conversion

**Multiple stages for decimation** (or interpolation) can reduce the number of filter coefficients in the filter specifications.  
The signal can be decimated more than once, using

a gradual change in sampling frequency.

Conventional decimation:



Decimation in mutiple stages (multistage):



Notes

Multistage Sample Rate Conversion

The multi-stage sample rate conversion decimation values,  $D_i$ :

f\_s / f\_s^new = D = D\_1 x D\_2 x ... x D\_k = product from i=1 to k of D\_i

where all  $D_i$  are integers. So for three stages,  $k = 3$  and

D = D\_1 x D\_2 x D\_3.



Notes

Multistage Sample Rate Conversion Problem

2

Q. The earlier problem can now be implemented using 2 decimation stages. Find out how many filter coefficients are necessary for a 2 stage decimation process.

A. The original sampling frequency  $f_s = 4.096\text{kHz}$  and the new (decimated signal) should have a sampling frequency of  $f_s^{\text{new}} = 128\text{Hz}$ . Multistage decimation with 2 stages requires that:

f\_s / f\_s^new = D = 4096 / 128 = 32 = D\_1 x D\_2.

The multistage decimation values can therefore be  $D_1 = 8$  and  $D_2 = 4$ , creating an intermediate signal with sampling frequency:  $f_s^{(1)} = f_s / 8 = 512\text{Hz}$ . The transition width can be longer with this higher sampling rate. We can keep the same passband frequency (60Hz). The transition width can go up to half the sampling rate:

f\_tw^(1) <= 512Hz / 2 - 60Hz = 196Hz

The number of Blackman filter coefficients for this stage is:

N\_1 = 5.98 x f\_s^(1) / f\_tw^(1) = 5.98 x 512 / 196 = 16, (rounded up to integer value).

Notes

Multistage Sample Rate Conversion Problem

2, cont'd

So  $N_1 = 8$  filter coefficients are required for the first decimation stage. The intermediate signal sampled at 512Hz is to be decimated by a factor of 4 to 128Hz for the second stage:

f\_s^new = f\_s^(2) = 512Hz / 4 = 128Hz.

The transition width for this (final) stage can then be:

f\_tw^(2) = 128Hz / 2 - 60Hz = 4Hz

So that the number of Blackman filter coefficients for this stage is:

N\_2 = 5.98 x f\_s^(2) / f\_tw^(2) = 5.98 x 128 / 4 = 192.

192 filter coefficients are required for this final stage. The combined filter coefficients for the two stages is:

N\_1 + N\_2 = 16 + 192 = 208,

which is considerably less than the original non-multistage decimation antialiasing filter requiring  $N = 6124$  coefficients.

Notes



# Lecture Summary

This lecture has covered

- Decimation,
- Interpolation,
- Non-integer sample rate conversion,
- Multistage sample rate conversion.

There are *many more* to topics and techniques in *multirate digital signal processing* including:

- Implementation techniques, e.g. polyphase filters
- and Applications.

Notes

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---