EE3S1 Signal Processing – DSP

Lecture 3: Downsampling and upsampling (Ch. 6.5 and 13.5)

Alle-Jan van der Veen

–

22 Sept 2025

Sampling, revisited

- Suppose we have sampled a signal $x(t)$ at a certain rate F_s, but later would have liked another rate. How can we "resample" $x[n]$?
- In other cases, we may have deliberately oversampled a signal, to simplify analog hardware implementations. How does that work?

Outline

- Sample rate conversion
  - Downsampling by factor D
  - Upsampling by a factor U
  - Sample rate conversion by a rational factor U/D
- Implementation
  - Multistage sample rate conversion
  - Polyphase filters (future lecture)

Prior knowledge

- Sampling
- DTFT

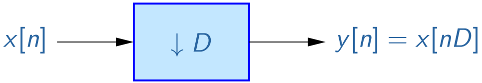
The slides cover Ch. 6.5 and part of 13.5 of the Holton book.

Downsampling by a factor D

Suppose we sampled an analog signal $x(t)$ at a rate $F_s = 1/T_s$, but want to reduce the rate by a factor of D. The new rate is $F_s' = 1/T_s'$.

$$T_s' = DT_s \quad \Rightarrow \quad y[n] = x(nT_s') = x(nDT_s) = x[nD]$$

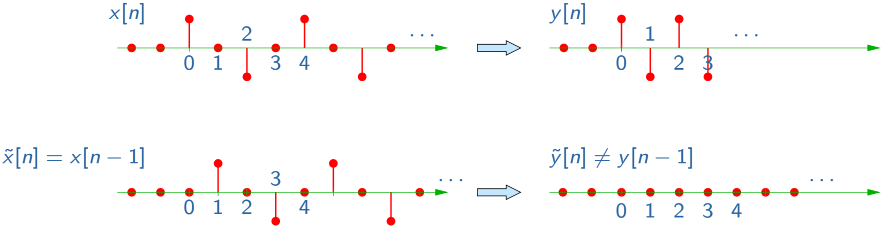
Thus, we simply drop samples of $x[n]$ and keep only every D-th sample.



Note: be aware that the notation $x[n]$ and $y[n]$ doesn't show that the two signals have a different rate!

Decimation by a factor D

Downsampling (decimation) is a linear time-varying operation:



- A delay on the input doesn't lead to a delay on the output.
  Therefore, the analysis is a bit more involved (convolution property of LTI systems doesn't hold; the order of blocks can't be reversed)

Downsampling by a factor D

How does the spectrum $Y(\omega)$ of the downsampled signal relate to the original spectrum $X(\omega)$?

- Define a sampling function $s[n]$ as a delta train:

$$s[n] = \sum_{k=-\infty}^{\infty} \delta[n - kD] = \begin{cases} 1, & n = 0, \pm D, \pm 2D, \cdots \\ 0, & \text{otherwise} \end{cases}$$

- We can write $s[n]$ as a sum of D exponentials (as before on sampling):

$$s[n] = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D}$$

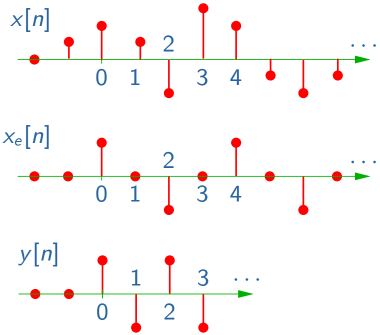
- The spectrum (DTFT) of $s[n]$ is another delta train:

$$S(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} \mathcal{F}\left\{e^{j2\pi kn/D}\right\} = \frac{2\pi}{D} \sum_{k=0}^{D-1} \delta(\omega - 2\pi k/D) \qquad -\pi \leq \omega \leq \pi$$

Downsampling by a factor D

To derive the spectrum $Y(\omega)$, we split the downsampling into 2 steps:

- Set samples to zero using the sample function: $x_e[n] = x[n]s[n]$; this leads to loss of information
- Drop the zero samples to obtain $y[n]$ (now without loss of information)



Downsampling by a factor D

We first look at $x_e[n] = x[n]s[n]$, which replaces samples of $x[n]$ by zero.

- We use the fact that a product in time domain relates to a convolution in frequency domain:

$$
\begin{aligned}
X_e(\omega) &= \frac{1}{2\pi} X(\omega) * S(\omega) \\
&= \frac{1}{2\pi} \frac{2\pi}{D} \sum_{k=0}^{D-1} X(\omega) * \delta(\omega - 2\pi k/D) \\
&= \frac{1}{D} \sum_{k=0}^{D-1} X(\omega - 2\pi k/D)
\end{aligned}
$$

- This is a sum of shifted copies of $X(\omega)$. The shifts are multiples of $2\pi/D$. This can lead to aliasing!

Downsampling by a factor D

Next, to obtain $y[n]$ we drop the zero samples (without further loss of information):

$$y[n] = x_e[nD] \quad \Leftrightarrow \quad Y(\omega) = \sum_{n=-\infty}^{\infty} y[n] e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_e[nD] e^{-j\omega n}$$

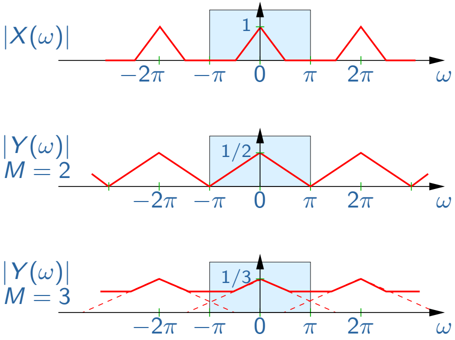
$$= \sum_{n=-\infty}^{\infty} x_e[n] e^{-j\omega n/D} = X_e(\omega/D)$$

- Thus: compressing $x_e[n]$ by dropping the zero samples leads to an expansion of the spectrum.
- Overall,

$$Y(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(\frac{\omega - 2\pi k}{D}\right)$$

Downsampling by a factor D

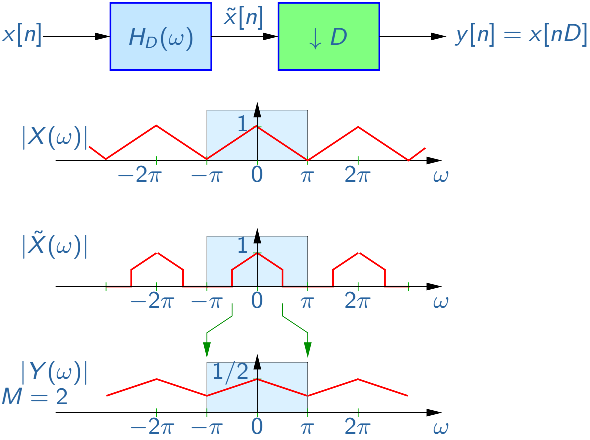
The spectrum of the decimated signal $Y(\omega)$ consists of shifted and stretched copies of the original spectrum $X(\omega)$.



- No aliasing if $X(\omega) = 0$ for $\frac{\pi}{D} \leq \omega \leq \pi \Rightarrow$ the signal should satisfy Nyquist at the new (lower) rate.

Downsampling by a factor D

How to avoid aliasing? Use a lowpass filter! (called *decimation filter*)

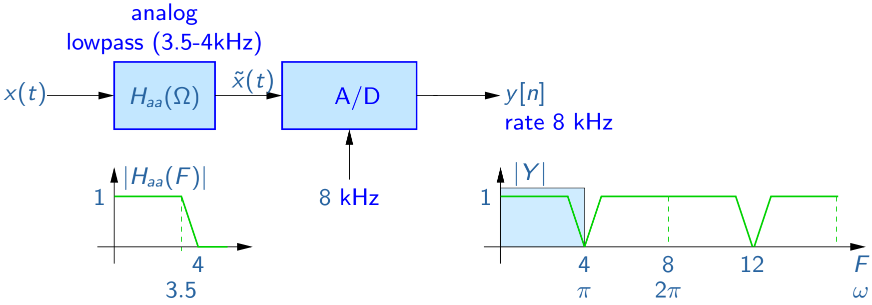


The lowpass filter has a cut-off at $\frac{\pi}{D}$.

Example: mobile phone speech signal

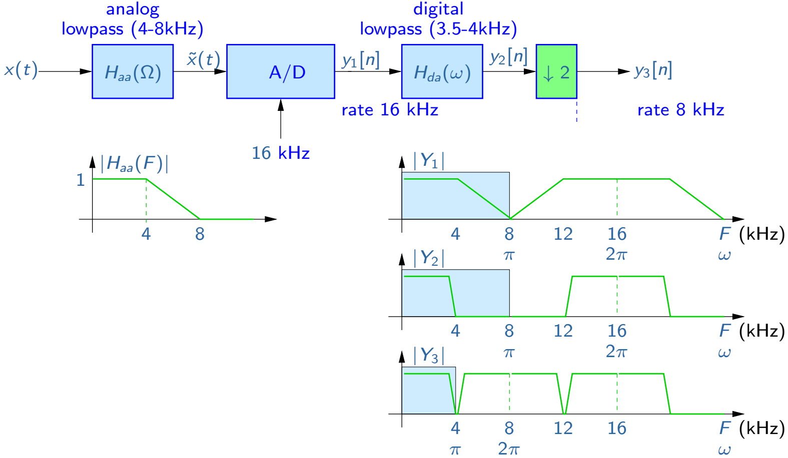
Consider a speech signal $x(t)$. For telephony, only frequencies up to 3.5 kHz are important, so $F_s = 8$ kHz should be sufficient.

Before sampling, we should apply an anti-aliasing filter $H_{aa}(\Omega)$ with a transition band between 3.5 and 4 kHz. This is very sharp, and gives an analog filter of high complexity!



Example: audio signal

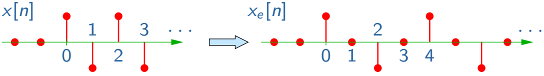
Alternative: sample at 16 kHz, and then downsample by a factor 2. The analog filter now has a reasonable transition band, the complexity has shifted to the digital domain.

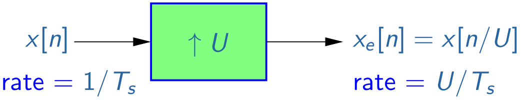


Upsampling with a factor U

Upsampling by a factor U means to insert $U-1$ zeros:

$$x_e[n] = \begin{cases} x[k], & n = kU \\ 0, & \text{otherwise} \end{cases}$$





Then, derive as before that

$$X_e(\omega) = X(\omega U)$$

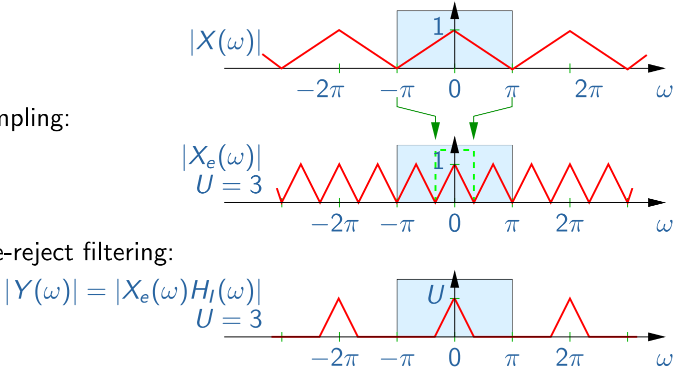
- The sampling rate of the signal increases by a factor U. There is no information loss!

Upsampling by a factor U

$$X_e(\omega) = X(\omega U)$$

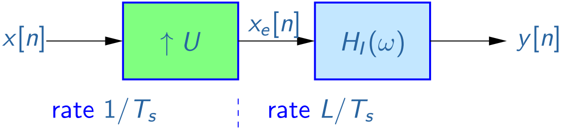
after upsampling:

after image-reject filtering:



(Explanations on the next slide)

Upsampling by a factor U



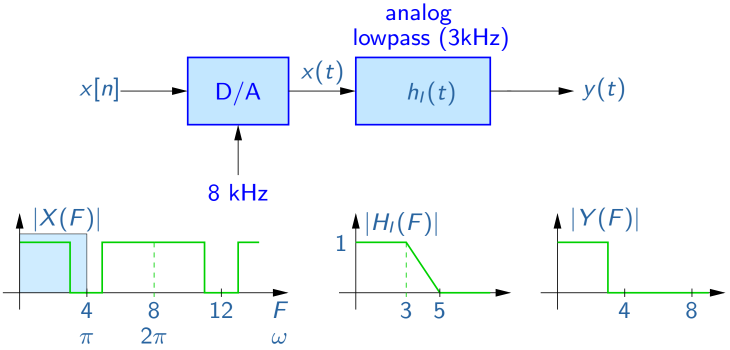
- The spectrum of the signal contracts by a factor U, and $U-1$ extra copies occur in the fundamental interval.
- We can remove the copies using a digital low-pass filter:

$$H_I(\omega) = \begin{cases} U, & 0 \leq |\omega| \leq \pi/U \\ 0, & \text{otherwise} \end{cases}$$

  This is called an *interpolation* (or image-reject) filter. In time domain, it "interpolates" the samples that were zero.
- The overall system is equivalent to a system sampled at a rate of U/T_s *provided there is no aliasing at the initial rate* $1/T_s$

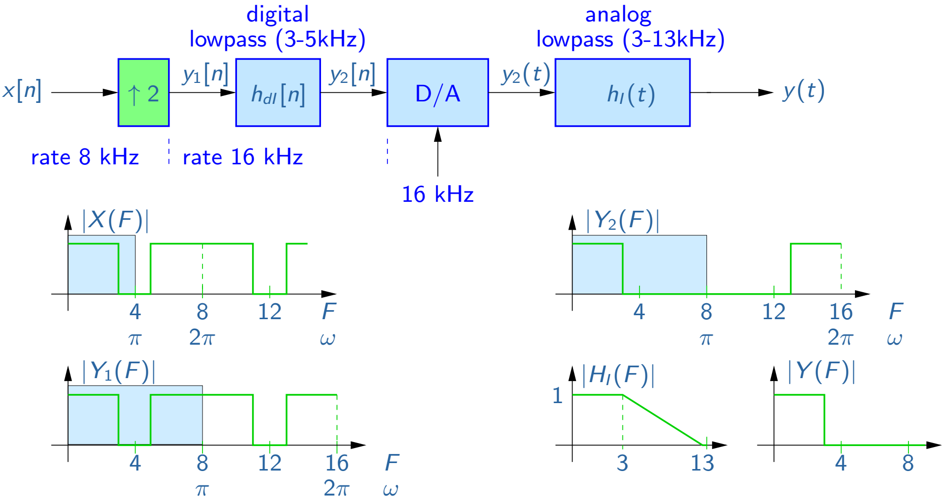
Example: speech signal

Let's assume that our speech signal has to be converted back to analog and the highest frequency is 3 kHz, sample rate 8 kHz. After D/A conversion we need an image rejection (interpolation) filter with transition band between 3–5 kHz.



Example: speech signal

Alternative: first upsample with a factor of 2, and then do the D/A



Now the analog lowpass filter has a more reasonable transition band.

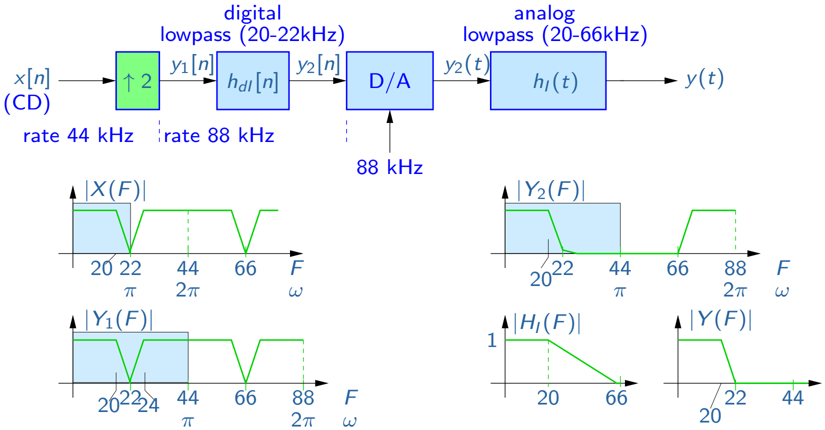
Similar exercise

A CD stores audio samples with a sampling rate of 44.1 kHz. The highest audible frequency by the human ear is around 20 kHz.

- On the CD player, the text "2 times oversampling" is written. What does it mean?
- Draw the block scheme of the reconstruction, including the frequency diagram of all signals.

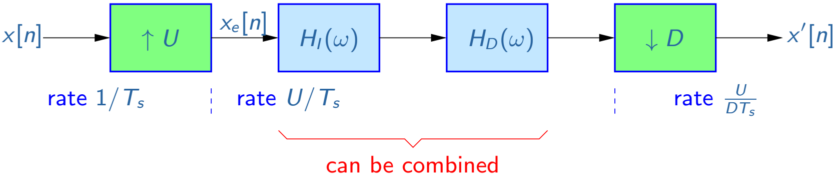
Similar exercise (cont'd)

This is about D/A conversion. The oversampling refers to inserting zeros between the available samples, which will increase its rate. The digital interpolation filter (lowpass filter) will interpolate the added zeros. After D/A conversion, the analog filter will be simpler.



Resampling by a rational factor U/D

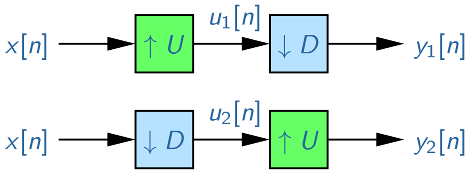
We can combine upsampling with U and downsampling with D to implement a sample rate conversion with any rational factor U/D.



- The two low-pass filters can be combined into a single one with a cut-off of $\omega_c = \min(\frac{\pi}{U}, \frac{\pi}{D})$.

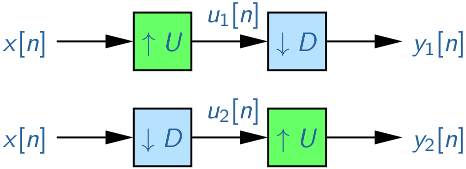
Resampling by a rational factor U/D

Can we swap the ordering of upsampling and downsampling?



Resampling by a rational factor U/D

Can we swap the ordering of upsampling and downsampling?

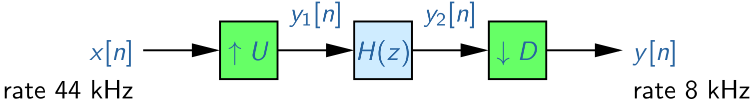


- No! If $U = D$ then $y_0[n] \neq y_1[n]$.
- In fact, $y_0[n] = y_1[n]$ if and only if U and D are relative primes. (Show this by working out an example, e.g. $U = 2$, $D = 3$)

Resampling by a rational factor U/D

Example

Given is a discrete-time audio signal $x[n]$ in CD quality: the sample rate is 44 kHz. To transmit this over a telephone link, we must reduce the sample rate to 8 kHz.

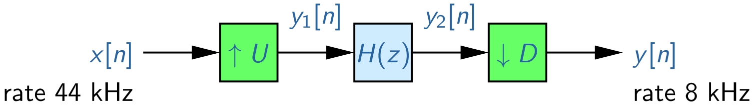


- What are suitable values for the upsampling factor U and downsampling factor D?

Resampling by a rational factor U/D

Example

Given is a discrete-time audio signal $x[n]$ in CD quality: the sample rate is 44 kHz. To transmit this over a telephone link, we must reduce the sample rate to 8 kHz.



- What are suitable values for the upsampling factor U and downsampling factor D?
  $U = 2$, $D = 11$, so that $44\frac{U}{D} = 8$.

Resampling by a rational factor U/D

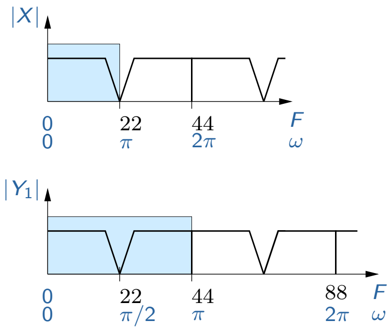
Example (cont'd)

- What is a specification for the filter $H(z)$?

Resampling by a rational factor U/D

Example (cont'd)

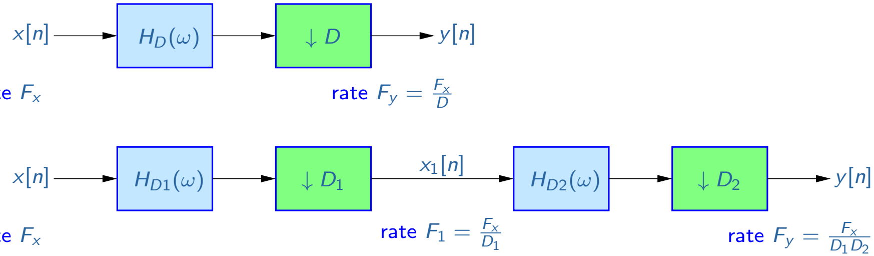
- What is a specification for the filter $H(z)$?



- $H(z)$ is a lowpass filter that should cut off above 4 kHz (corresponding to $\omega_c = \frac{\pi}{11}$).

Multistage resampling

If U or D are too large, it is more efficient to implement conversion in multiple stages. This leads to lower order filters.



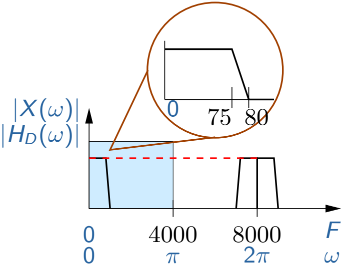
For example: factorize $D = D_1 \cdot D_2$

Multistage resampling
Example

Given an audio signal sampled at $F_x = 8$ kHz. We want to keep the frequencies $0 - 80$ Hz and resample to $F_y = 160$ Hz. Hence, the decimation factor is $D = 50$.

Let us assume the following filter specifications:

- passband: 0–75 Hz, ripple $\delta_1 = 10^{-2}$
- transition band: 75–80 Hz
- stopband: 80–4000 Hz, ripple $\delta_2 = 10^{-4}$



Example (continued)

Heuristic formula to estimate filter order (Kaiser):

$$\hat{N} = \frac{-10\log(\delta_1\delta_2) - 13}{14.6\,\delta f}, \quad \text{where} \quad \delta f = \frac{F_{stop} - F_{pass}}{F_x}$$

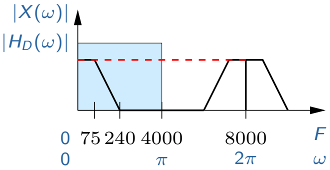
- Here: $\delta f = \frac{5}{8000} = \frac{1}{1600}$ so that $\hat{N} = 5151$: very high

Example (continued)

Instead, we now implement the filtering and downsampling in two stages, $D_1 = 25$ and $D_2 = 2$.

Stage 1: The filter specs of the first decimation filter $H_{D1}(z)$ are:

- passband: 0–75 Hz (unchanged)
- transition band: 75–240 Hz (because $320 - 80 = 240$, see next slide)
- ripples $\delta_1 = 0.5 \cdot 10^{-2}$ and $\delta_2 = 10^{-4}$ (half for δ_1 because we'll have 2 stages)
- new sample rate: $F_1 = F_x / D_1 = 320$ Hz

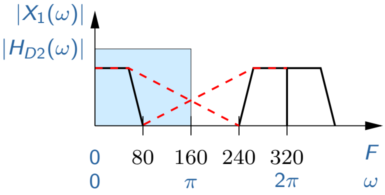


Here, $\delta f = \frac{165}{8000} = \frac{1}{48}$, therefore $\hat{N}_1 = 167$, which is much smaller than before.

Example (continued)

Stage 2:

- passband: 0–75 Hz
- transition band: 75–80 Hz
- ripples $\delta_1 = 0.5 \cdot 10^{-2}$ and $\delta_2 = 10^{-4}$



Here, $\hat{N}_2 = 220$, therefore, the total number of filter coefficients $167 + 220 = 387$ is much smaller than the 5151 using the single stage implementation.

Another advantage is a reduction in data rates to run these filters, which will be discussed later in Ch. 13.

To do:

- Study Chapter 6.5 and 13.5
- Try to make exercise ...
- Check old exams for related exercises