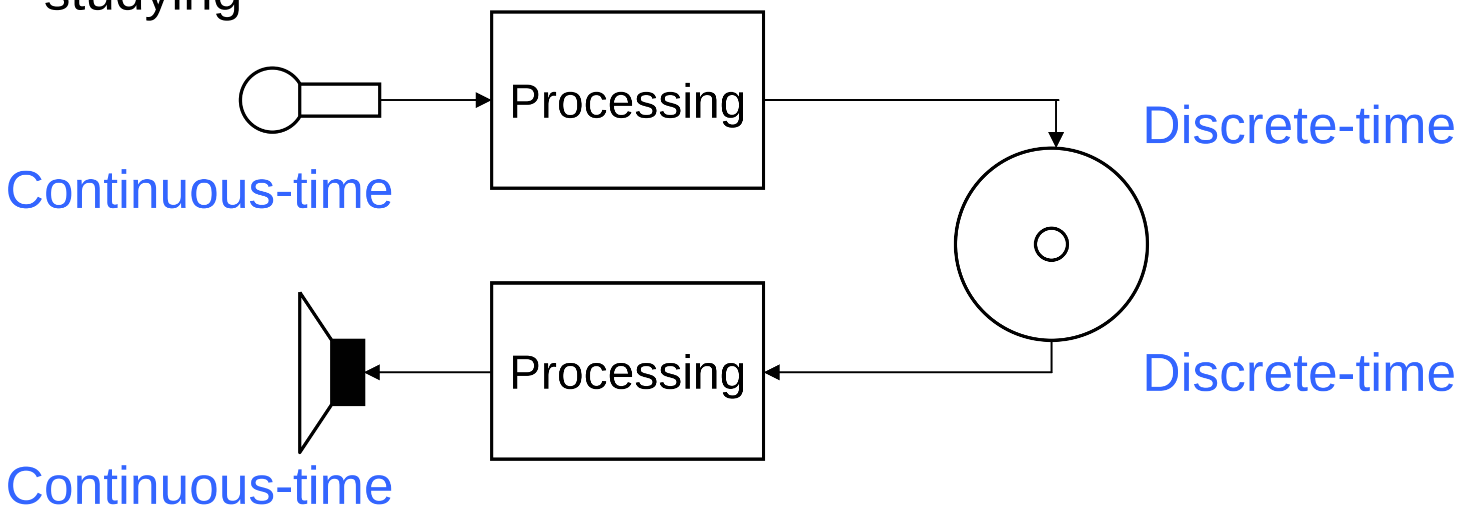
Information Sciences: Signal Processing

Lecture 3: Sampling and Interpolation

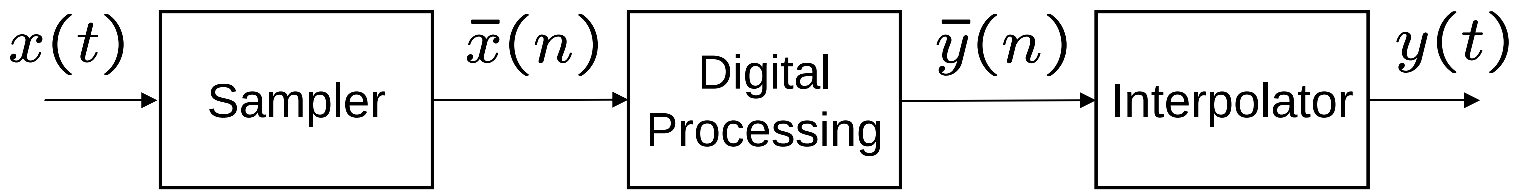
- In this lecture we will talk about the possibility of using discrete-time signals to process (and record) continuous-time signals
- The intuitive idea is that a discrete-time signal should be a good approximation of a continuous-time signal, if the sampling frequency is large enough.
- However, how large should the sampling frequency be? What happens if it is not the case?

- Let's consider the block diagram of the system that we are studying



- Input and output are continuous-time signal. However, we need to use discrete-time signals in order to work with optical disks
- We need an element to transform a signal from continuous-time to discrete-time. This is the sampler
- Conversely, we need to reconstruct a continuous-time signal from a discrete-time signal. We use an interpolator to do that

- We can redraw the block diagram in the following way

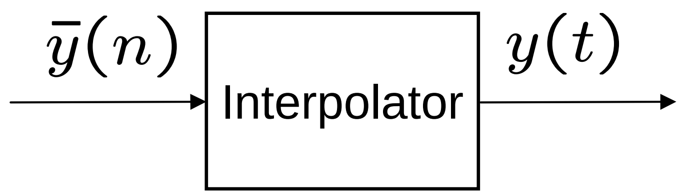


- x(t) and y(t) are the input and output signal, i.e. the acoustic pressures measured by the microphone and produced by the loudspeaker
- We neglect any additional processing done by microphone, loudspeaker, amplifier, etc. and we concentrate on the conversions introduced by the sampler and the interpolator
- The block “Digital Processing” represents, in our case, the disk. So, we should have $\bar{y}(n) = \bar{x}(n)$ In general, it can be something ,more complicate like a moving average (we neglect quantization)

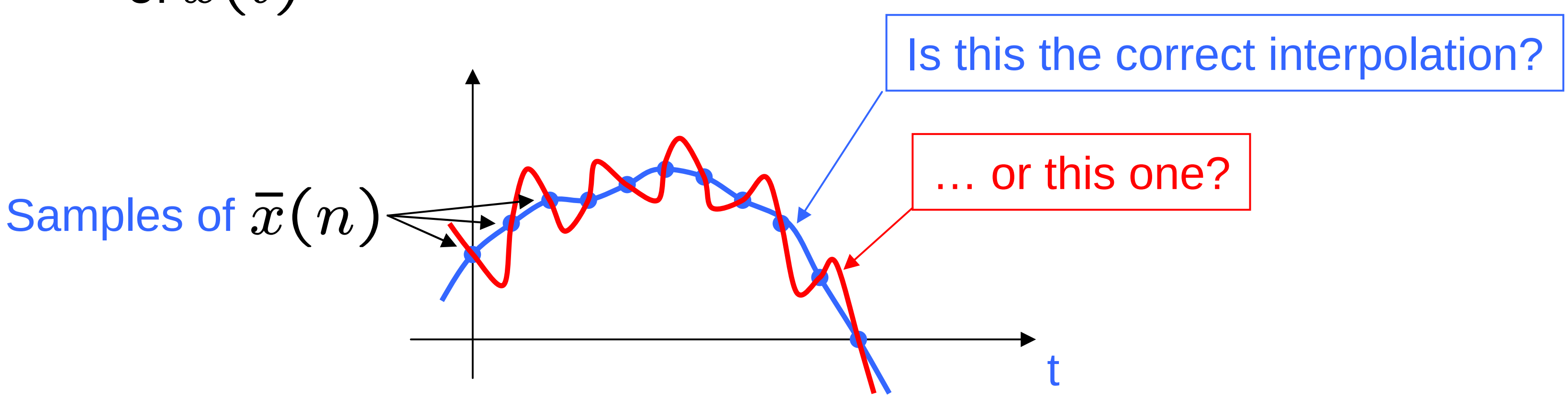
- The sampler measure the input signal x(t) only at certain instants multiple of the sampling period T_S i.e. we map the temporal index n to the continuous time t according to

$$n \to t = nT_S$$

- Notation: in this lecture, we use the overline to denote the sampled version of a certain signal, e.g. $\bar{x}(n)$ is the sampled version of $x(t)$
- Remember that $f_S = 1/T_S$ is called the sampling frequency



- The interpolator maps a discrete-time signal into a continuous-time signal. Ideally, we would like to have y(t)=x(t), i.e. perfect reconstruction
- This is not possible, even if the disk is ideal, i.e. $\bar{y}(n) = \bar{x}(n)$ the discrete-time signal $\bar{x}(n)$ is an ambiguous representation of $x(t)$



- Why do we use digital systems? ("digital" is another word for discrete-time, probably better for marketing…)
- Theoretically, we can obtain the same result using analog systems (i.e. continuous-time) for example:

Edison's phonograph ca. 1899

Philips compact cassette 1963

Philips Laserdisk 1969

- However, it is much easier to work with digital signals on certain media (such as optical disks). Also, digital systems are more reliable and stable over time (the behavior of the system does not change). They are also more flexible (a new software can be installed to implement new features)

- Let's study more in detail the ambiguity of a sampled version of a continuous time signal. We consider initially the case of a single sinusoid, i.e.

$$y(t) = \sin(2\pi f t)$$

- If we sample such a sinusoid we obtain a discrete time signal of the form $$\bar{y}(n) = \sin(2\pi f n T_S)$$

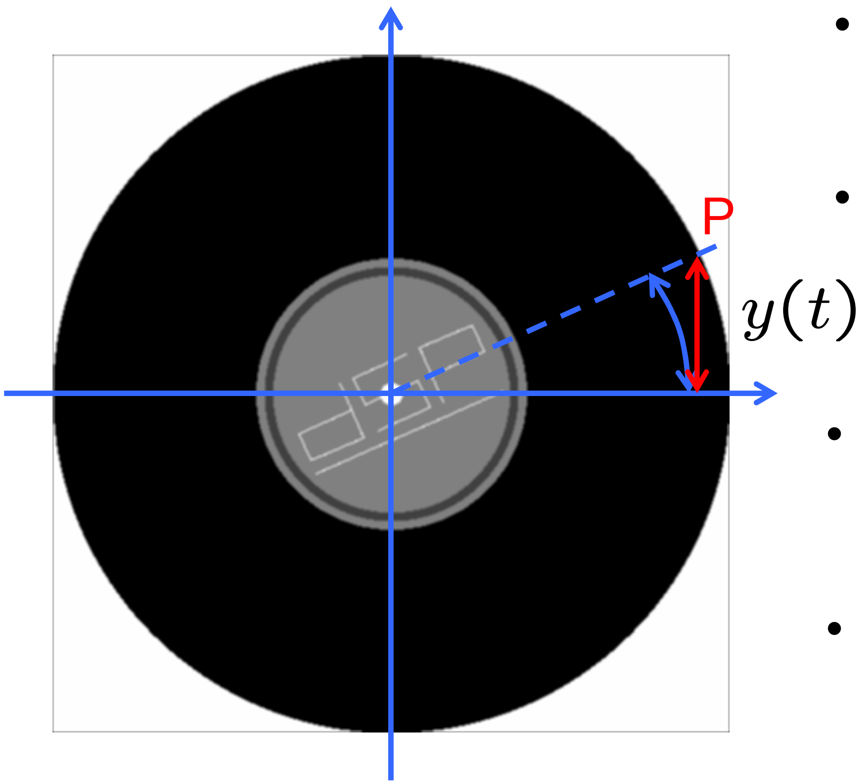
  where T_S is the sampling period

- The problem that we want to investigate is to find if there are other sinusoids of the form

$$y_P(t) = \sin(2\pi f_P t)$$

  such that when sampling $y_P(t)$ we obtain the same discrete time signal $\bar{y}(n)$. Of course, we would like that $f_P = f$ Unfortunately, this is not the case…

- To develop an intuition, we imagine the following experience



- You are in a disco and the DJ is passing the latest hit of the "DSP," your favorite band!
- The disk rotates on the turntable at constant speed corresponding to a frequency of rotation f (rotations per second)
- You observe the rotation of the label. At time t, the angle of the label is $2\pi ft$
- If you follow the movement of the point P, on the border of the disk, you see that the y coordinate of P is exactly the signal y(t) defined previously (we assume the radius of the disk equal to 1)

- All the lights are switched off and a stroboscopic spot is blinking at constant rate f_S
- Assume, for simplicity, f=1 (one rotation of the disk in one second) and f_S =20. What would you observe?

- You observe the disk only 20 times during one rotation.
- The position of the point P is the signal $\bar{y}(n)$ defined earlier
- We have the impression that we can easily follow the movement of the disk, despite sampling

- What happens if the DJ changes the speed of rotation f of the disk? The number of positions that we observe is reduced

- If f=2, 10 positions…
- If f=5, 4 positions…
- If f=20/3, 3 positions…
- If f=10, 2 positions…
  Here we may hesitate.. Is the disk rotating clockwise or counterclockwise?
- If f=15, … the rotation seems to be clockwise now and the speed of rotation 5 (4 position) i.e. same result of f=-5
  What happens?

- Explanation:
  - We don't know the movement of the disk when the light is switched off. Our visual system perceives motion as if it occurred in the direction of the minimum angle, but there is an ambiguity.
  - Between two consecutive positions, the disk may do any number of additional complete rotations, either clockwise or counterclockwise
  - If f is the actual frequency of rotation there are several frequencies f_P that explain the observations, i.e. produce the same sampled positions. The values of f_P are given by

$$f_P = f + N f_S$$

Where N is an arbitrary integer

- Proof:
  - The true frequency f produces sampled positions at angles

$$2\pi f T_S n, \qquad n \in \mathbb{Z}$$

  - The frequency f_P corresponds to the angles

$$2\pi f_P T_S n = (2\pi f T_S + \underbrace{2\pi N f_S T_S}_{\text{Number of (presumed) complete rotations between consecutive blinks}})n = 2\pi f T_S n$$

- This ambiguity on the disk rotation corresponds to an ambiguity on the frequency a sinusoid:

$$\bar{y}(n) = \sin(2\pi f T_S n) = \sin(2\pi f_P T_S n)$$

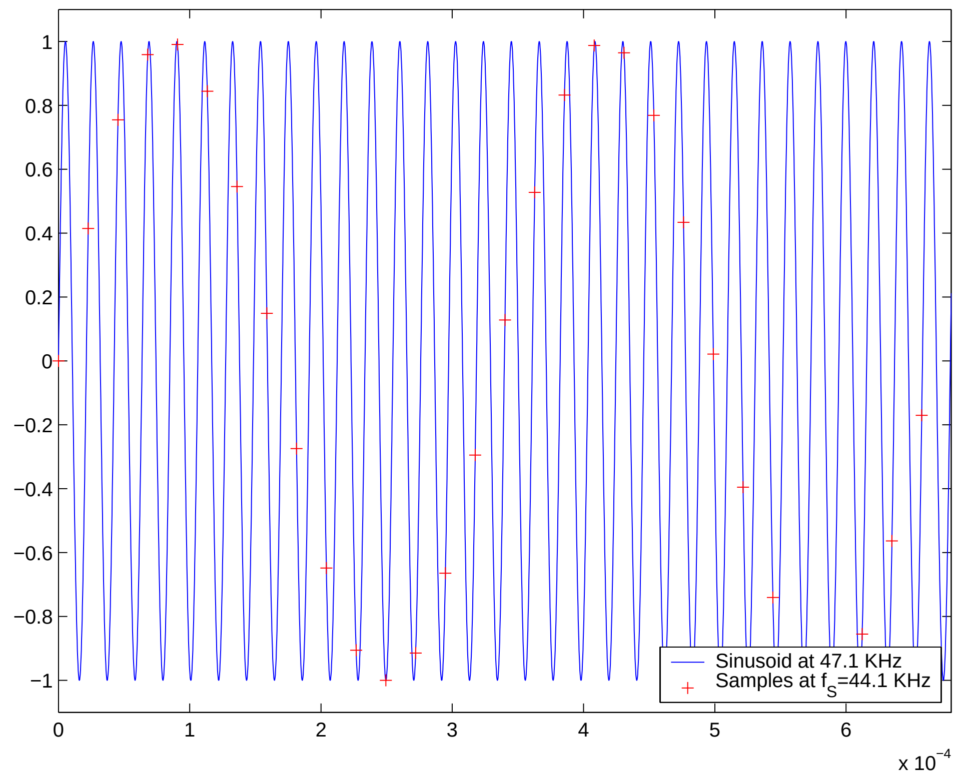
- If we choose one of the valid frequencies f_P, we can define a continuous time sinusoid

$$y_P(t) = \sin(2\pi f_P t)$$

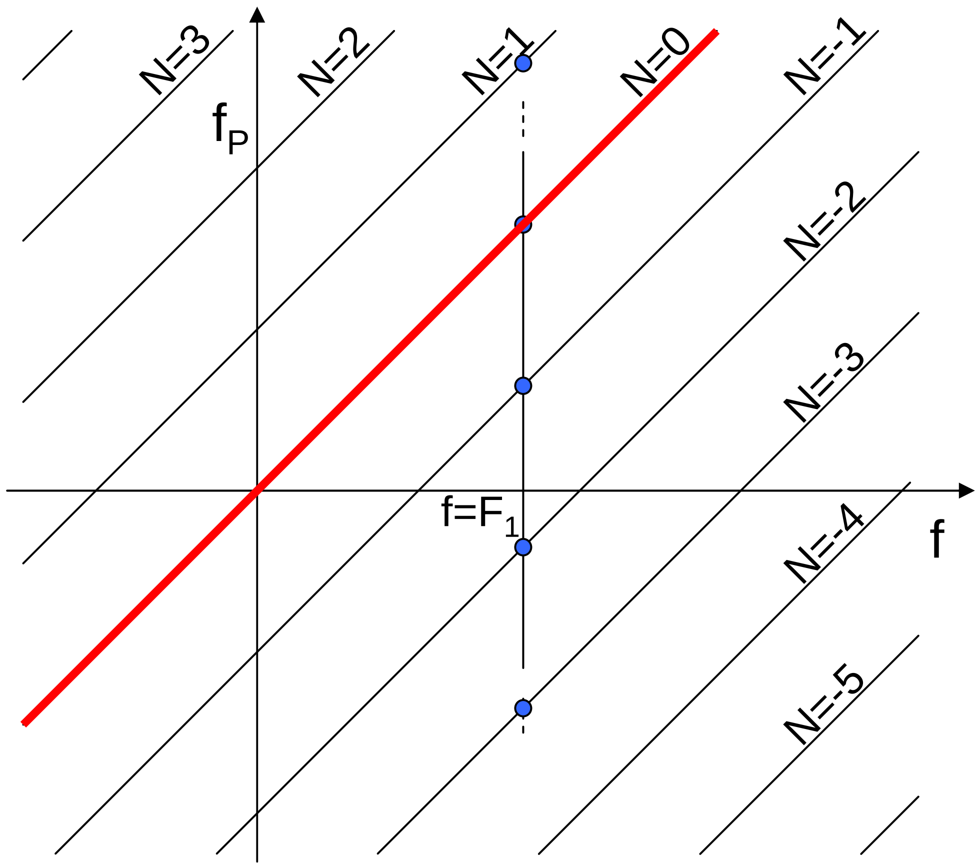
This sinusoid, when sampled, gives the same samples of the sinusoid y(t)

- We call this phenomenon aliasing, since the sinusoid $y_P(t)$ takes the “identity” of y(t) when it is sampled (i.e. it is an alias of y(t))

- Example: audio CD are recorded using a sampling frequency f_S=44.1 KHz, which corresponds to a sampling period T_S=22.7μs/sample. A sinusoid of frequency 47.1 KHz, when sampled, is indistinguishable from a sinusoid of frequency 3KHz, since 47.1 KHz = 3 KHz + f_S



- How can we avoid aliasing? Let's draw a diagram where for each frequency f of the sinusoid, we show the possible frequencies f_P

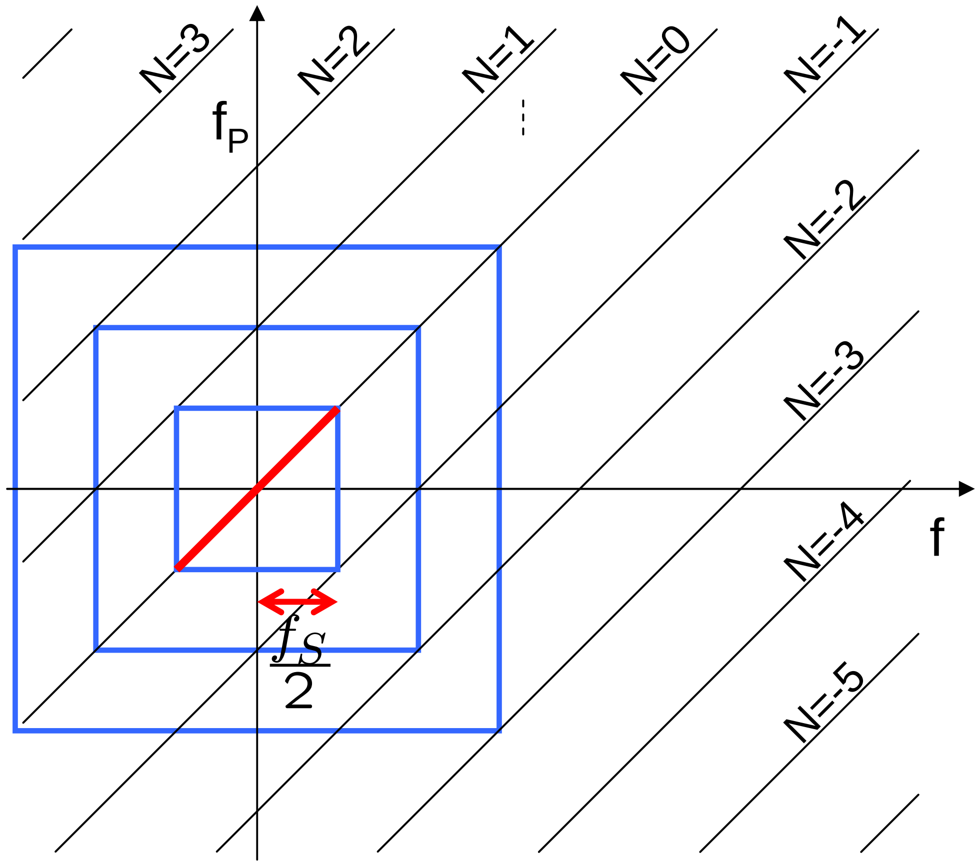


If $f=F_1$, the possible values of f_P are given by the dots

We would like to have only the value corresponding to the line N=0

To obtain that, we restrict the range of the frequency f

- How should we reduce the range of f to avoid ambiguity?



We restrict the range of f until there is only one possible frequency f_P

The frequency f has to be in the range

$$f \in (-\frac{f_S}{2}, \frac{f_S}{2})$$

The frequency $f_S/2$ is the limit of the input frequency and it is called Nyquist frequency

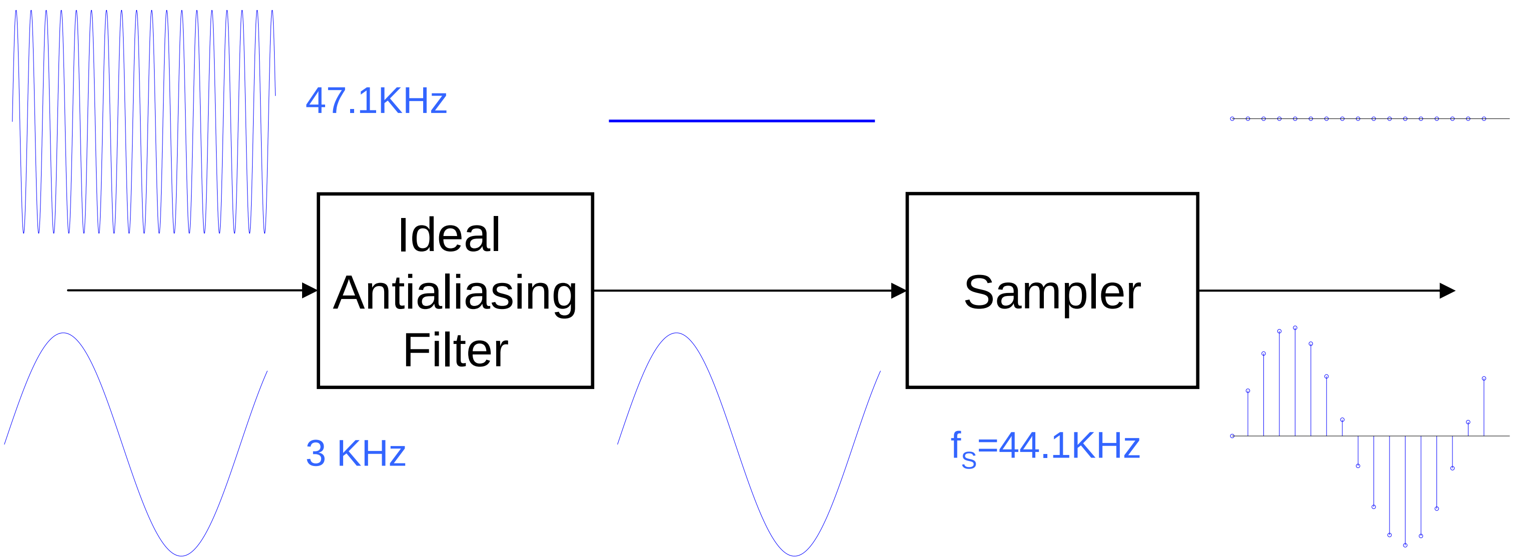
- How can we avoid that the frequency goes out of the range?
- In the previous lecture we saw that filters can transform signals For example, a tone control can attenuate some frequencies, leaving unchanged some other ones
- We use a filter that suppress completely frequencies larger than the Nyquist frequency. Such a filter is called an antialiasing filter
- It can be shown that the impulse response of the ideal antialiasing filter is given by

$$h(t) = \frac{\sin(\pi f_S t)}{\pi t} = f_S \mathrm{sinc}(f_S t)$$

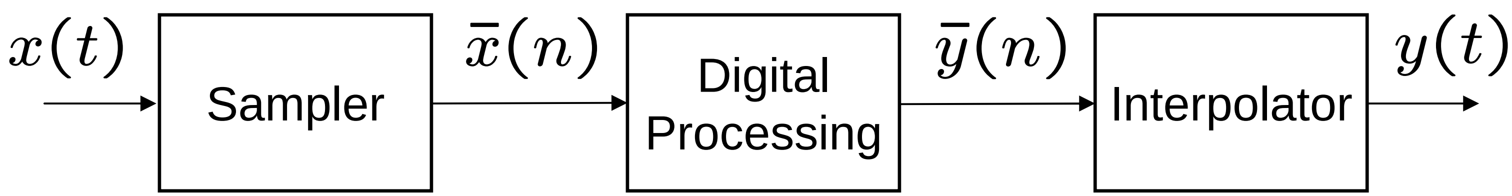
where $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ is called the sinc function

- The ideal antialiasing filter is non causal (i.e. it can “see” the input in the future) and cannot be realized. The solution is to approximate the ideal filter and tolerate some aliasing

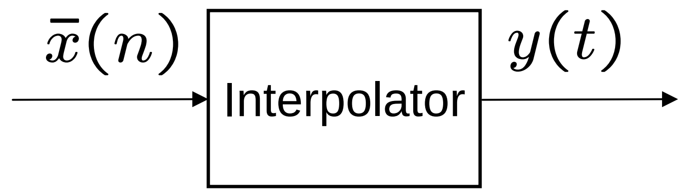
- Example: audio CD are recorded using a sampling frequency f_S=44.1 KHz. The Nyquist frequency is f_N=22.05 KHz. To avoid aliasing, we need an ideal filter that suppress sinusoids of frequency f outside of the range (-22.05 KHz, 22.05 KHz).
- A sinusoid of frequency f=3KHz is left unchanged by the filter and can be reproduced by the CD player. A sinusoid of frequency f=47.1 KHz is suppressed by the antialiasing filter and cannot be reproduced.



- Now that we solved the ambiguity on the sampled sinusoid, we can consider again the original block diagram and address the problem of interpolation



- Assume that the processing (i.e. the recording of the disk) is ideal and $\bar{y}(n) = \bar{x}(n)$. We would like to interpolate the samples $\bar{x}(n)$ such that the resulting signal y(t) is equal to the input signal x(t) (or at least it should be a good approximation)

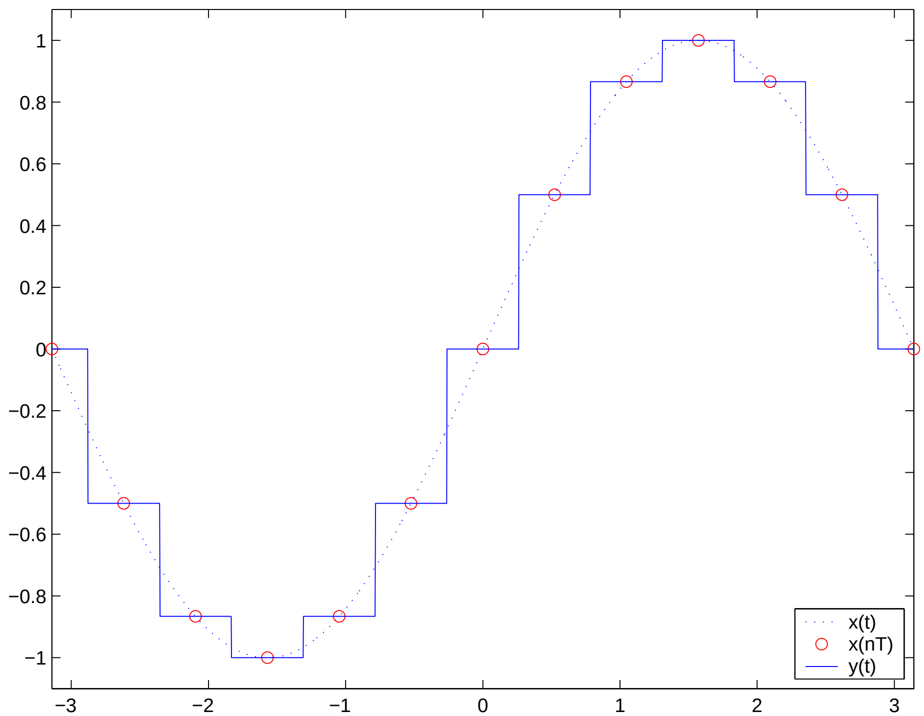


- The interpolator should be able to generate a continuous time signal y(t) such that:
  - I. The signal passes through the interpolated samples, i.e

$$y(nT_S) = \bar{x}(n)$$

- Again, if we consider the case of a single sinusoid, we would like that
  - II. If the samples are taken from a sinusoid of frequency f in the range ($-f_S/2$, $f_S/2$), then the interpolator reconstructs a sinusoid of frequency f

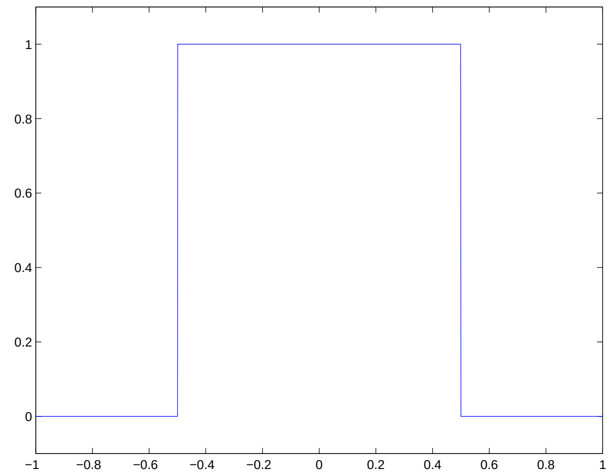
- Let’s consider a very simple interpolation method, called the zero-order hold
- Very rough assumption: we interpolate with a piecewise constant function passing through the samples, e.g.:



- Let's define the function rect(t) as

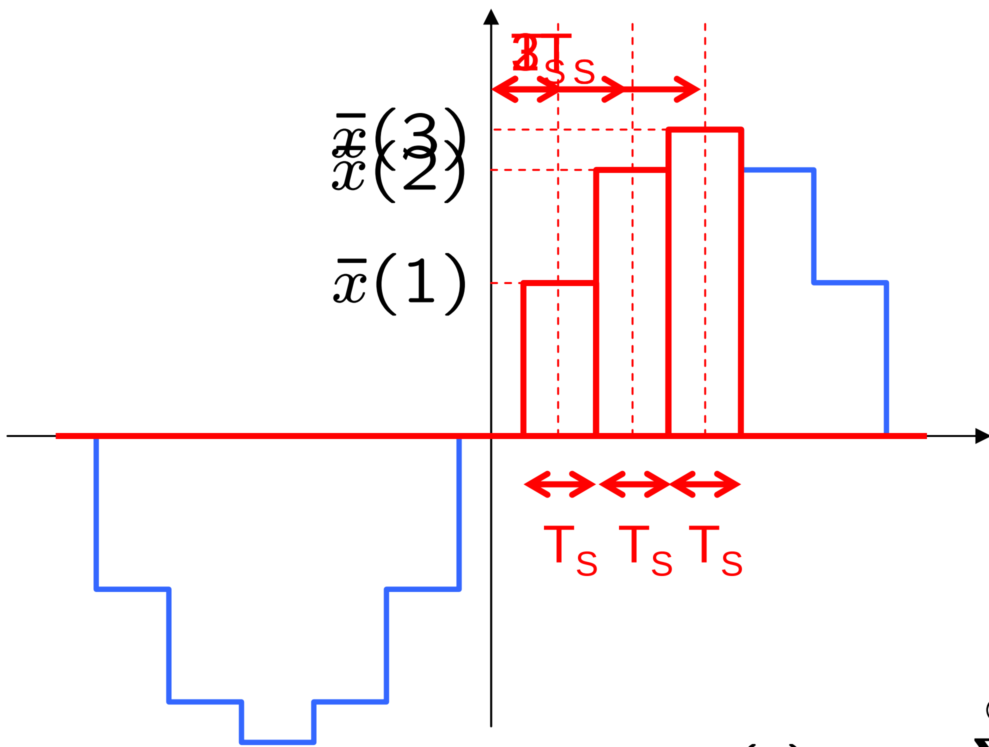
$$\text{rect}(t) = \begin{cases} 1 & \text{if } -1/2 < t < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

i.e. the graph is



- We want to express the interpolation by using a sum of translated rect functions

- Each sample is associated to a rect function:



$$\bar{x}(1)\mathrm{rect}\left(\frac{t-T_S}{T_S}\right) + \ldots$$

$$\bar{x}(2)\mathrm{rect}\left(\frac{t-2T_S}{T_S}\right) + \ldots$$

$$\bar{x}(3)\mathrm{rect}\left(\frac{t-3T_S}{T_S}\right) + \ldots$$

$$\vdots$$

$$y(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n)\mathrm{rect}\left(\frac{t-nT_S}{T_S}\right)$$

- The zero-order hold produces a signal that passes through the samples; however, the results is very far from a sinusoid. How can we do better?
- Remark that the interpolation has the structure:

$$y(t) = \sum_{n=-\infty}^{\infty} \bar{x}(n)h(\frac{t - nT_S}{T_S})$$

where h is the rect function. However, we can use any function h we want, provided that h(n)=δ(n) (to satisfy condition I)

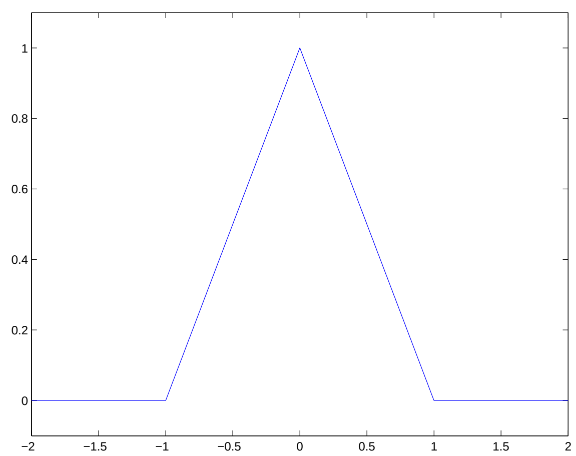
- You can verify that this type of interpolator is a linear system (for every function h). This will be useful later.

- The idea is to use a function h smoother than the rect

- Let's take for example

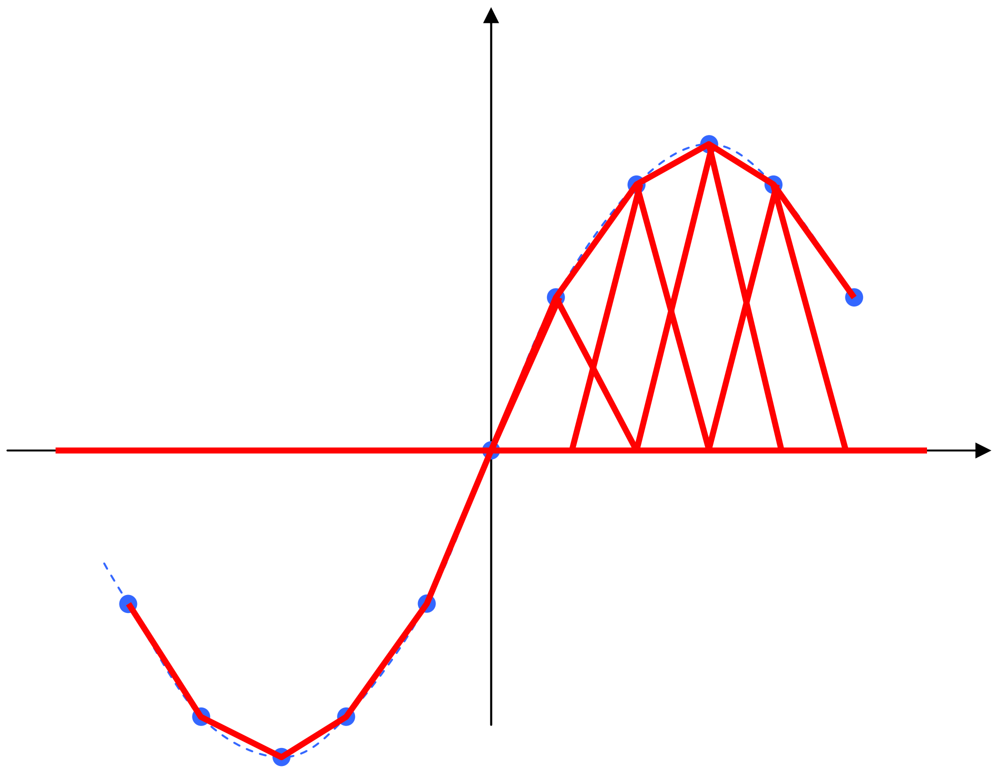
$$h(t) = \mathrm{tri}(t) = \begin{cases} 1 - |t| & \text{if } -1 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

now the graph of h(t) is



- This is a valid function h, since h(n)=δ(n) and it is smoother than the rect
- What is the interpolated signal?

- By summing all the contributions…



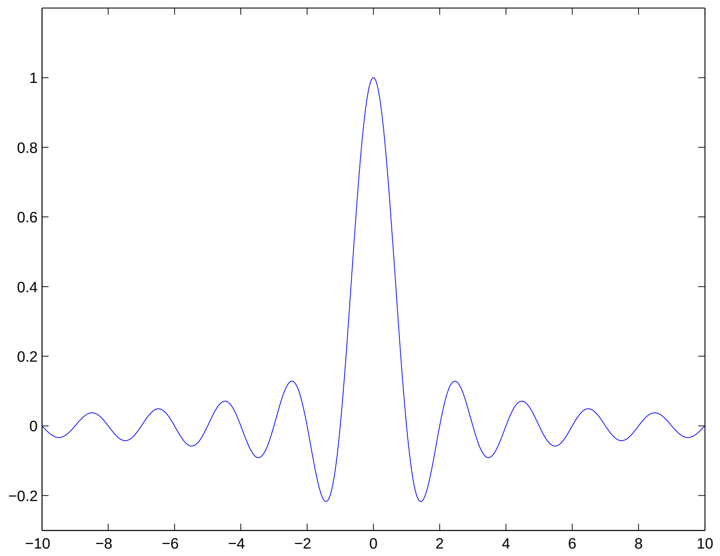
- The interpolating function is piecewise linear and passes through the sampling points. This is called a linear interpolation. Don't confuse this with linear systems!

- Linear interpolation takes into account pairs of consecutive samples to produce the interpolating function, while the zero-order hold considered only one point (the nearest). Can we do better by considering more and more consecutive points for the interpolation?
- The answer is yes. We can consider the interpolator that passes through N points and then take the limit of the interpolating function h, when N goes to infinity
- The result is the ideal interpolator. It can be shown that the function h for the ideal interpolator is given by

$$h(t) = \mathrm{sinc}(t)$$

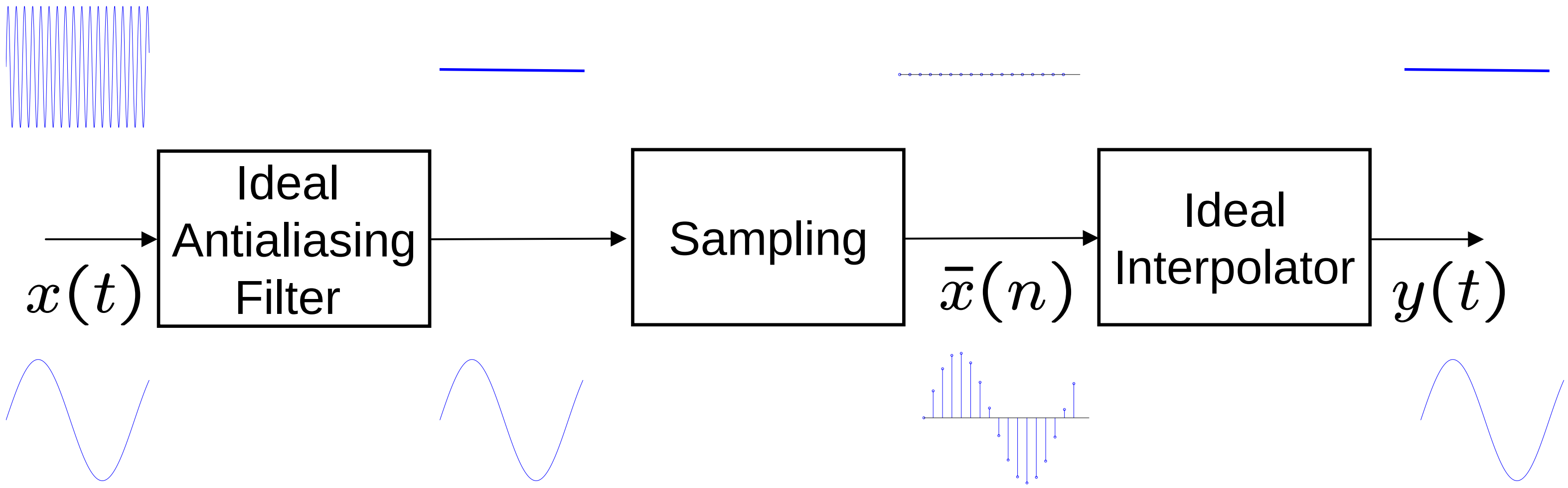
Surprisingly it is the same function that is used to build the ideal antialiasing filter! This is not a coincidence, but you will need some more math to see that

- The graph of the sinc function is given by



- The sinc function is a good interpolator since it passes through the all samples (h(n)=δ(n) and it is smooth (actually it can be considered the “smoothest” interpolator)

- If we reconsider the complete chain…



- We are able to sample, record, and reproduce perfectly any sinusoid of frequency f such that

$$f \in (-\frac{f_S}{2}, \frac{f_S}{2})$$

- Too good to be (completely) true. The ideal interpolator, as the ideal filter, cannot be constructed but only approximated

- What happens if we have something else than a sinusoid?
- Suppose that the input signal is the sum of 2 sinusoids, i.e.

$$y(t) = P_1 \sin(2\pi f_1 t + \phi_1) + P_2 \sin(2\pi f_2 t + \phi_2)$$

  - The antialiasing filter, the sampler, and the interpolator are linear systems; hence, we can study the response for each sinusoid and sum the result!
  - If both f_1 and f_2 are in the range $(-f_S/2, f_S/2)$, then the two sinusoids are represented with no ambiguity by the sampled signal and the ideal interpolator is able to reconstruct them
  - Conclusion: we can sample and interpolate the sum of two sinusoids if their frequencies are such that

$$B = \max(f_1, f_2) < \frac{f_S}{2}$$

  - B is called the bandwidth of the input signal

- We can do the same for a sum of several sinusoids:
  - Suppose that y(t) is the sum of N sinusoids of frequency f_1, f_2,…, f_N, we define the bandwidth B as

$$B = \max(f_1, f_2, \ldots, f_N)$$

  - Then, for the linearity of the system, if $B<f_S/2$, then the sampled signal represents the input signal without ambiguity and the ideal interpolator reconstruct it perfectly
  - If $B \geq f_S/2$then the sinusoids with frequency larger than $f_S/2$ are suppressed by the system, while the others are reconstructed perfectly

- Example
  - The C major chord (Do+) is composed by the note C, E, and G (Do, Mi, Sol) which can be played by summing 3 sinusoids at frequencies

    f_C=523.2511 Hz, f_E=659.2551 Hz, f_G=783.9909 Hz
  - If f_S=1600Hz, the chord is reproduced perfectly by interpolating the sampled signal. In fact, the bandwidth is B=783.9909 and $B<f_S/2$
  - If f_S=1400Hz, the C and the E are reproduced, but not the G, which is suppressed by the antialiasing filter
  - If we keep f_S=1400Hz and we suppress the antialiasing filter, then aliasing appears for the component at frequency f_G. The ideal interpolator reconstructs it with a sinusoid at frequency f_G-f_S,=-616.0091 Hz. In fact,

$$\sin(2\pi f_G T_S n) = \sin(2\pi (f_G - f_S) T_S n)$$

- What happens for the signals that are not sinusoids?
  - It can be shown that every signal can be expressed by a “sum” (actually an integral) of sinusoids of frequencies varying from zero to infinity. This is called the Fourier transform of the signal
  - Again, we can define the bandwidth B as the maximum frequency of the components of the signal
  - If B is not infinite, than we say that the signal is bandlimited
  - Due to the linearity of the whole system, we can study independently each component of the input signal

This is summarized in the following theorem…

Theorem (Whittaker-Nyquist-Kotelnikov-Shannon)

Let x(t) be a continuous time signal of bandwidth B and $\bar{x}(n) = x(nT_S)$ a sampled version of x(t). If the sampling frequency $f_S = 1/T_S$ is such that

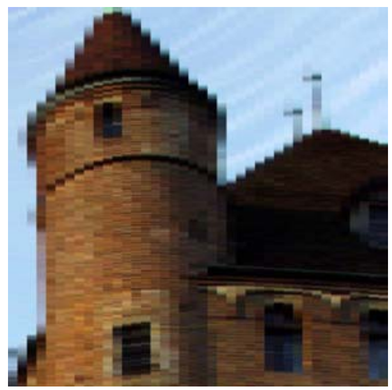
$$B < \frac{f_S}{2}$$

then x(t) can be reconstructed perfectly from the samples $\bar{x}(n)$ by using an ideal interpolator

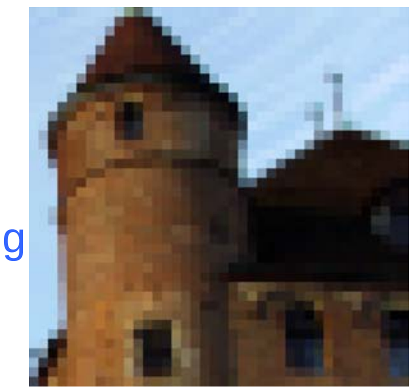
- A digital system process sampled versions of continuous time signals
- The sampler and interpolator allow to convert continuous time signal into discrete time signals and reconstruct them
- Sampled signals cannot represented unambiguously continuous time signals, since many (infinite) continuous time signals are mapped onto the same discrete time signal
- To avoid the ambiguity, the range of frequencies processed by the system is limited by using an antialiasing filter. The range of frequencies is proportional to the sampling frequency
- The ideal interpolator is able to reconstruct perfectly sinusoids in the admitted range of frequencies
- Signals other then sinusoids can always be represented as an infinite sum of sinusoids. If all the components are in the valid range of frequencies, then the signal can be reconstructed perfectly

- What we have for a sinusoid is valid also for images. In addition, we sample along 2 dimensions
- Remember that an image represents light intensity as a function of position

Y sampling



X sampling

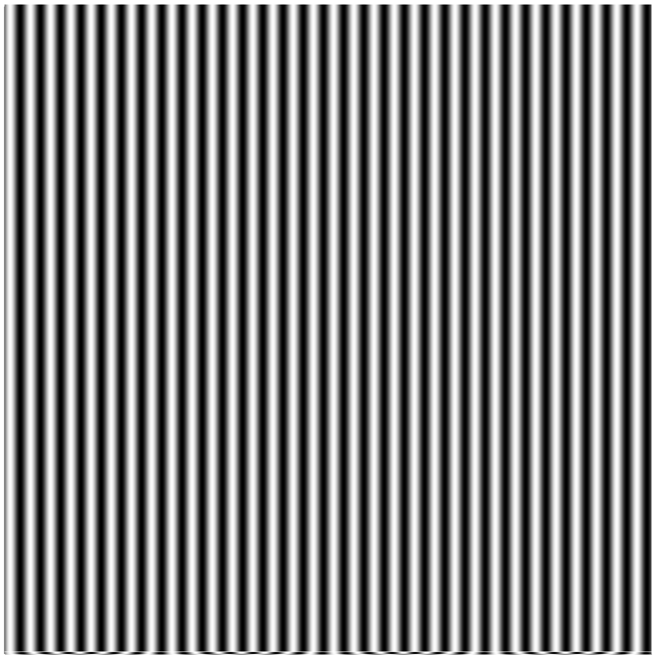


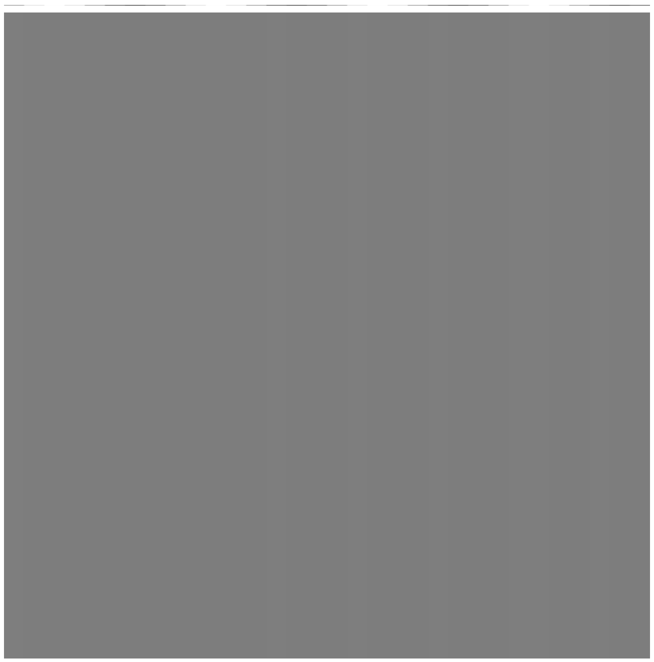
Continuous image

Discrete image

- Each sampling step may potentially give aliasing

- What you have seen for a sinusoid remains valid for an image line, e.g.



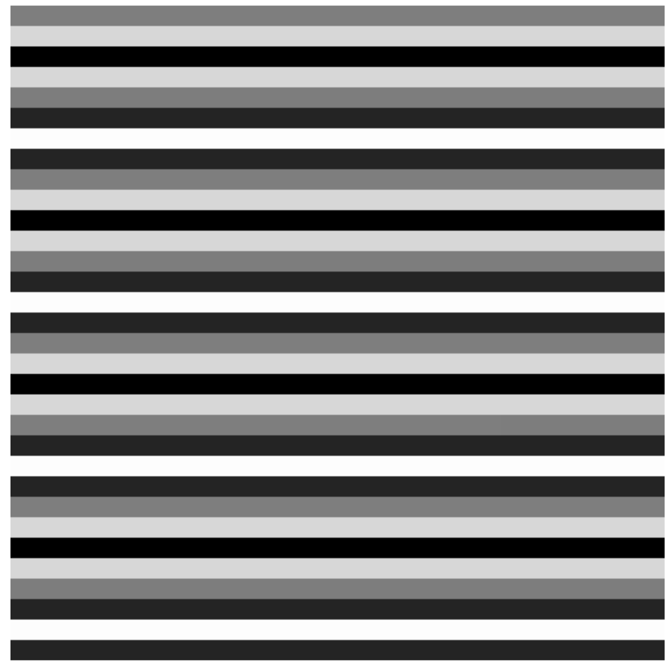


f=[illegible] CPW
f_s=32 CPW

The intepolator is often a zero-order hold (very poor results)

- We obtain the same effect if the sinusoid is along the Y axis
- If the sinusoid is tilted, we may have aliasing on X, Y or both. This gives some patterns called Moire patterns

Angle = [illegible]
f=20 f_X=[illegible] f_Y=[illegible]



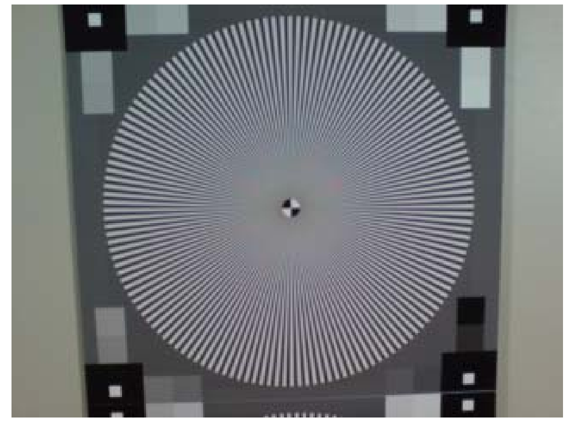
f_{SX} = 32 f_{SY}=32
Aliasing along X

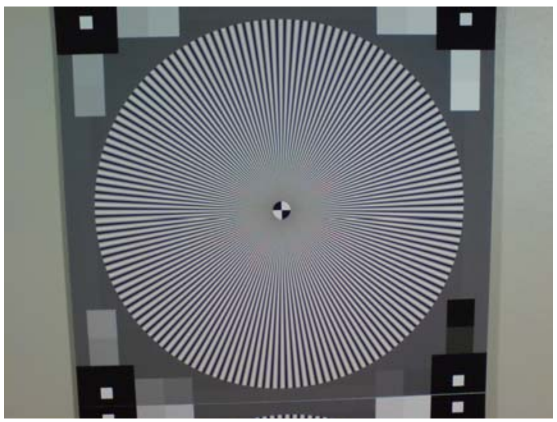
- Do you remember this picture from the 1st lecture?

These artifacts are due to aliasing. The regular structure of the bricks gives a Moire pattern after sampling

- To avoid Moire patterns, camera optics include a low pass filter. However, this is far from being ideal and some amount of aliasing is always present

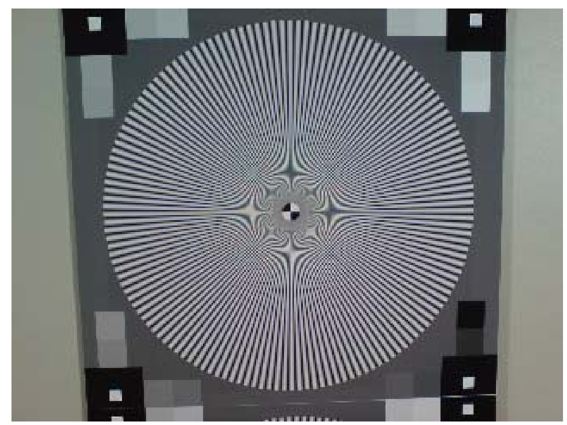
With low pass



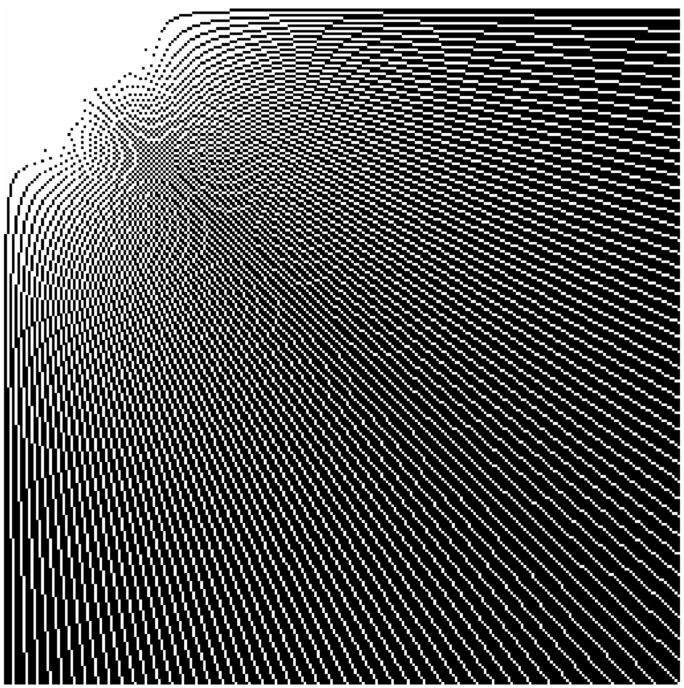


Original Image

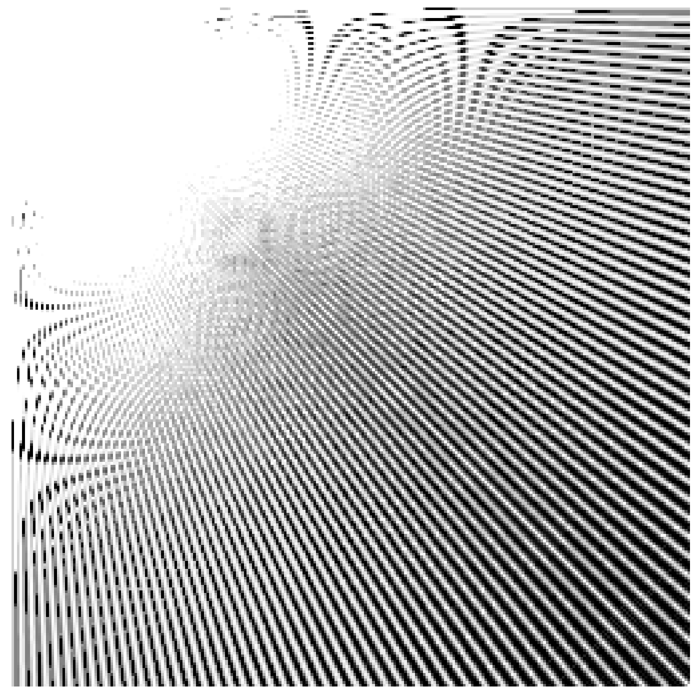
Without low pass



- In computer graphics, scenes are described with mathematical models representing planes, lines, etc. The rendered images are obtained by sampling the continuous space description at the pixel positions. Aliasing artifacts may appear if antialiasing filters are not used



Without antialiasing



With antialiasing