

Analog to Digital Conversion: Sampling from the Perspective of Pattern Recognition



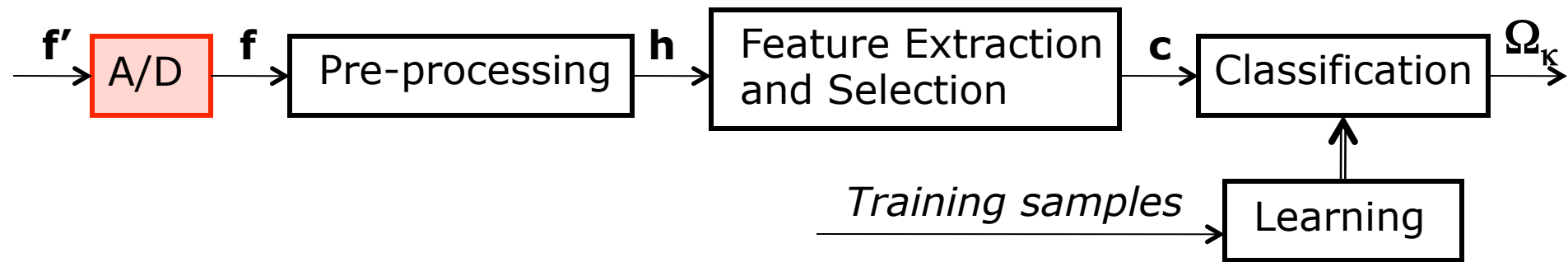
Dr. Elli Angelopoulou

Lehrstuhl für Mustererkennung (Informatik 5)

Friedrich-Alexander-Universität Erlangen-Nürnberg



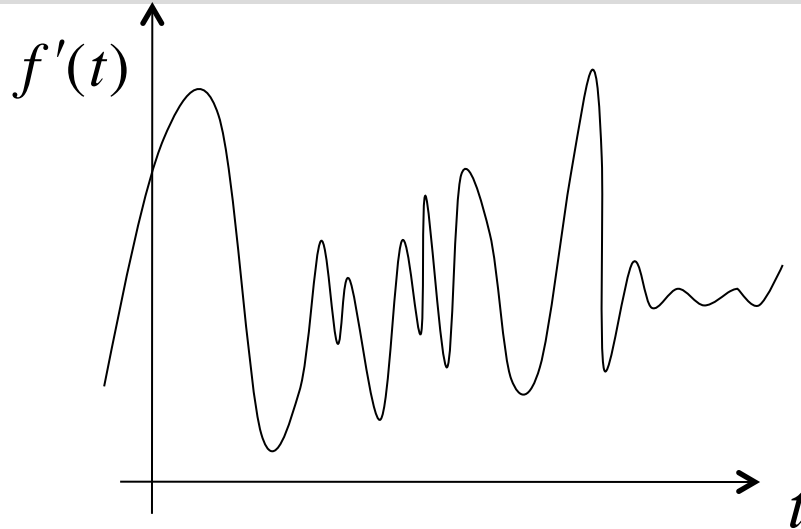
Pattern Recognition Pipeline



- The goal of analog to digital conversion is to gather sensed data f' and change it to a representation that is amenable to further digital processing.



Need for A/D Conversion



- Continuous range of t values
- Continuous range of amplitude $f'(t)$ values.
- We can only store a finite amount of values
- in a finite number of bits (discrete values).
- Goal: Find a discrete representation such that the original analog signal can be accurately reconstructed.

On Accuracy



- We want to have the analog signal **accurately reconstructed**.
- What is accurate?



On Accuracy

- We want to have the analog signal **accurately reconstructed**.
- What is accurate? Ideally no loss of information.
- Sometimes in order to get better speeds we accept some minimal information loss.
- We often have to face trade-offs:
 - voice recording where you skip letters, or pauses
 - digital images with aliasing effects
 - movies with blocky look



The two Aspects of A/D Conversion

- The function $f'(t)$ must be represented by a *vector* \vec{f} or by a *sequence of numbers* using a *finite* number of values.
- For higher dimensional signals, like an image, the input function $f'(x,y)$ must be represented by a *vector* \vec{f} or by a *sequence of numbers* at distinct locations (i,j) . At each such location there is only a *finite* number of values that can be stored.
- In the context of pictures:
 - How many pixels do I need? (How many (i,j) locations?)
 - How many bits per pixel?

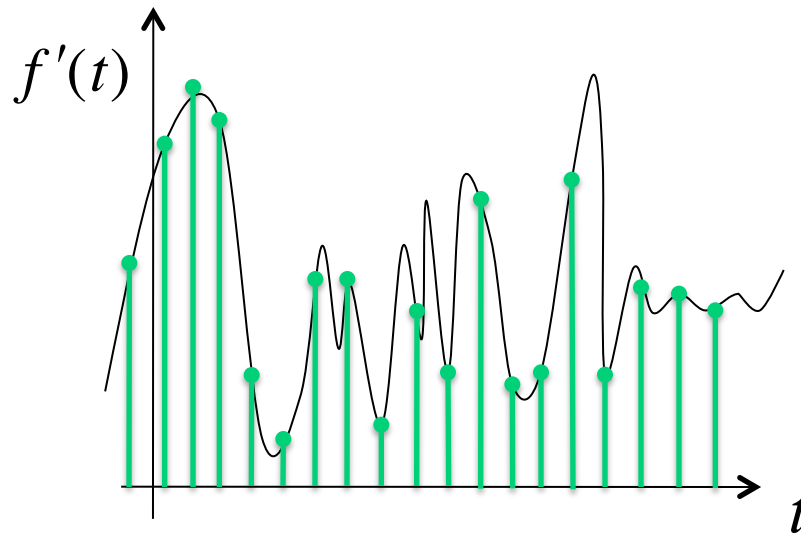


A/D Conversion Steps

■ The A/D conversion (coding) involves:

1. measuring the amplitude values (or function values) at a finite number of positions:

sampling,



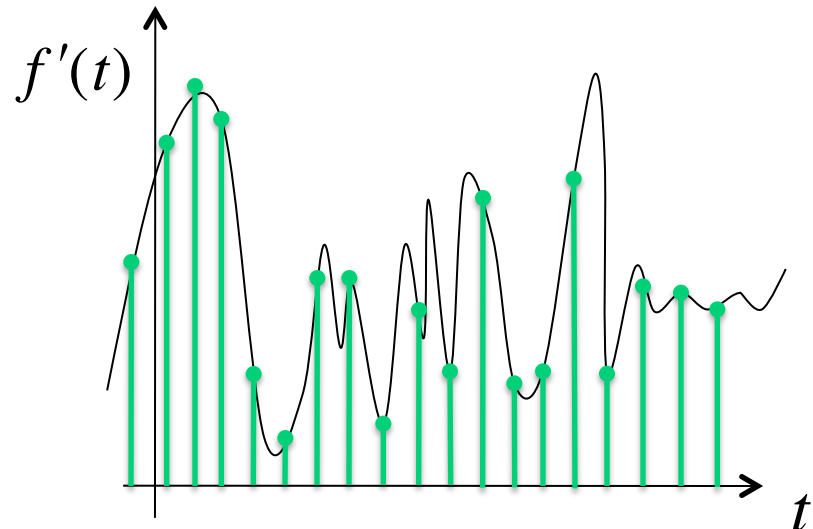
2. representing the amplitude values by a finite number of natural numbers:

quantization



Sampling Issues

- We could have different sampling steps along the sampling axis, but in most cases we assume Regular (equidistant) sampling.
- Even under regular sampling, one must decide:
 - ✧ Where do we take the samples along the analog signal so that we can properly reconstruct the original function.
 - ✧ In other words: What is the sampling interval dt or dx ?





Quantization Issues

- Along the vertical axis we also have continuous values that we can only store using a finite number of natural numbers.
- Typical image options:
 - 8 -16 bits per pixel for grayscale images
 - 8 -16 bits per color channel (R,G,B) per pixel for color images
 - 1 bit per pixel for black/white images
 - special encoding per application
- Unlike sampling, quantization intervals are often not equidistant.
- In the case of non-uniform quantization, the behavior of the quantizer is decided by the *characteristic function*, which relates the input continuous values to their discrete representation.



A/D Analysis Tools

■ Important questions:

1. How do we decide the sampling rate?
2. How do we derive the characteristic function of the quantizer?
3. How can these affect my pattern recognition system?

■ In order to fulfill the necessary performance guarantees (accurate reconstruction) we need to use the appropriate tools.

■ Sampling Tool: Fourier Analysis

Fourier Analysis allows us to study signals as a collection of periodic signals. This periodicity (frequency) then guides the sampling rate.

■ Quantization Tool: Probability Theory

Study previous behavior of the signal. High probability values use dense quantization. Lower probability values use sparse quantization.

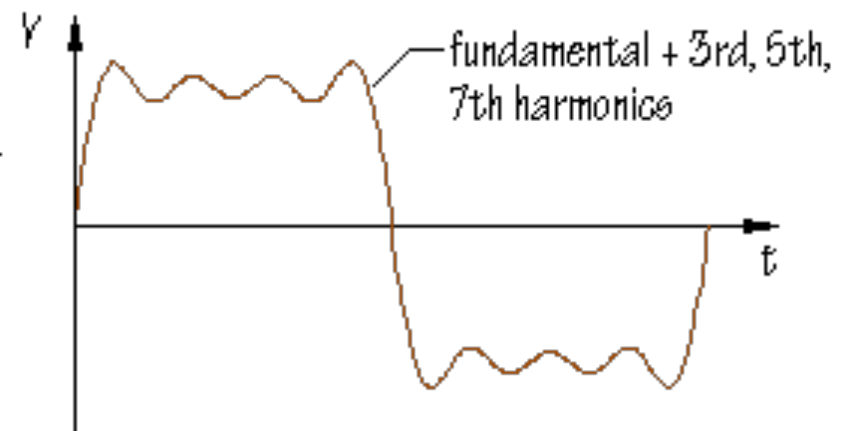
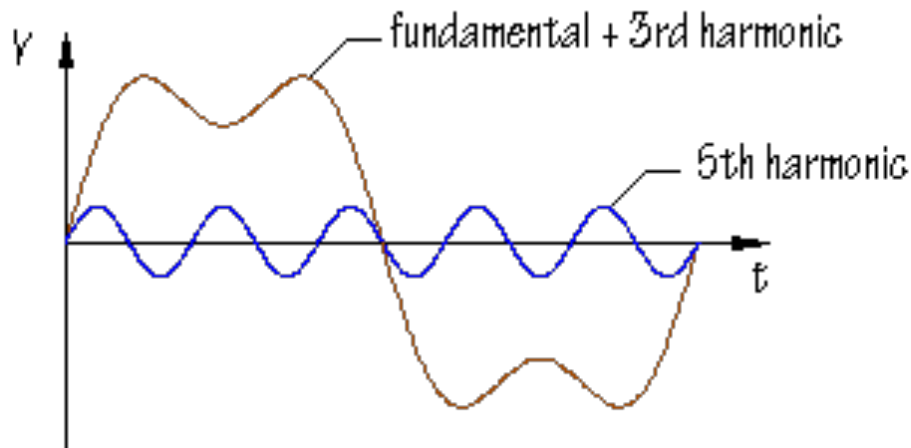
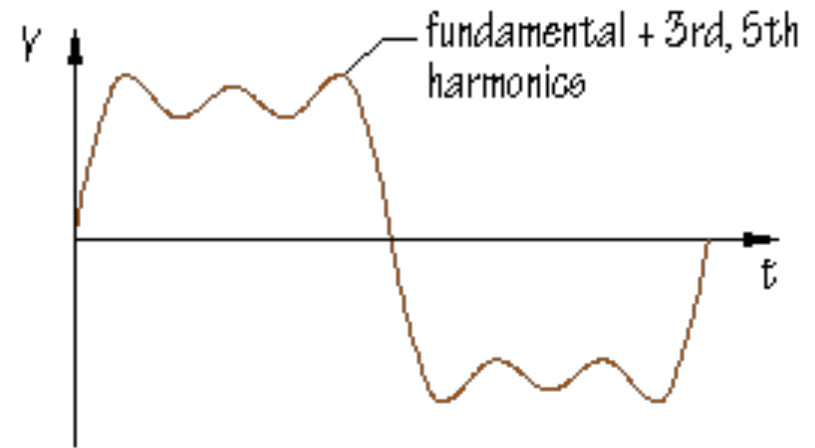
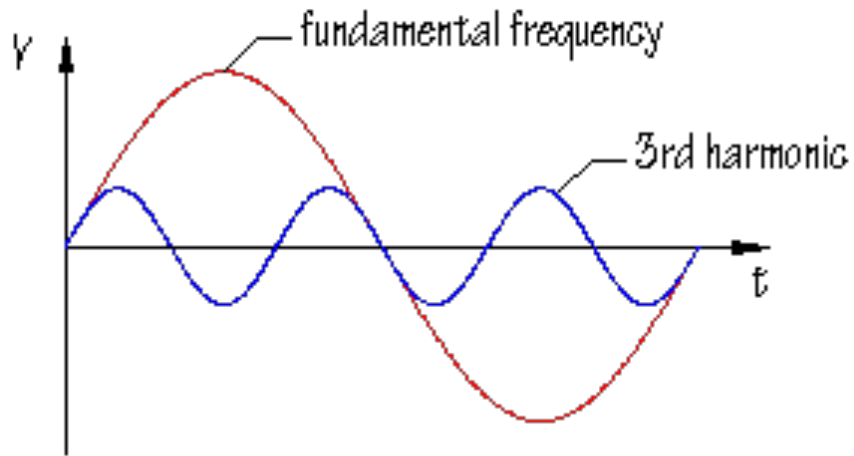
Fourier Analysis



- Based on the Fourier series.
- The original theory showed how any periodic function can be decomposed to a set of sines and cosines.
- The theory was generalized for non-periodic functions.
- Through the Fourier analysis we have a technique of decomposing complex patterns into a collection of simpler patterns.



Signal Decomposition Example





Fourier Transform

- How do we find the underlying sines and cosines of a function $f(x)$?
- In other words how do we get the Fourier series of $f(x)$?
- Using the Fourier Transform:

$$\mathcal{F}(\omega) = \text{FT}\{f(x)\} = \int_{-\infty}^{\infty} f(x)e^{-j\omega x} dx$$

where $\mathcal{F}(\omega)$ is the signal in the frequency domain and ω is the frequency of the sinusoidal wave.

Note: the signal must be absolutely integrable, $\int_{-\infty}^{\infty} |f(x)| dx < \infty$

- Given $\mathcal{F}(\omega)$, how do we get the original signal $f(x)$ back?
- Using the Inverse Fourier Transform:

$$f(x) = \text{FT}^{-1}\{\mathcal{F}(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\omega)e^{jx\omega} d\omega$$



Fourier Series

- A periodic function $f(x)$ has the Fourier series

$$\mathcal{F}(\omega) = \sum_{k=-\infty}^{\infty} a_k e^{2\pi j \left(\frac{k\omega}{2\omega_0} \right)}$$

where ω_0 is the periodicity of the signal and a_k are the Fourier coefficients.

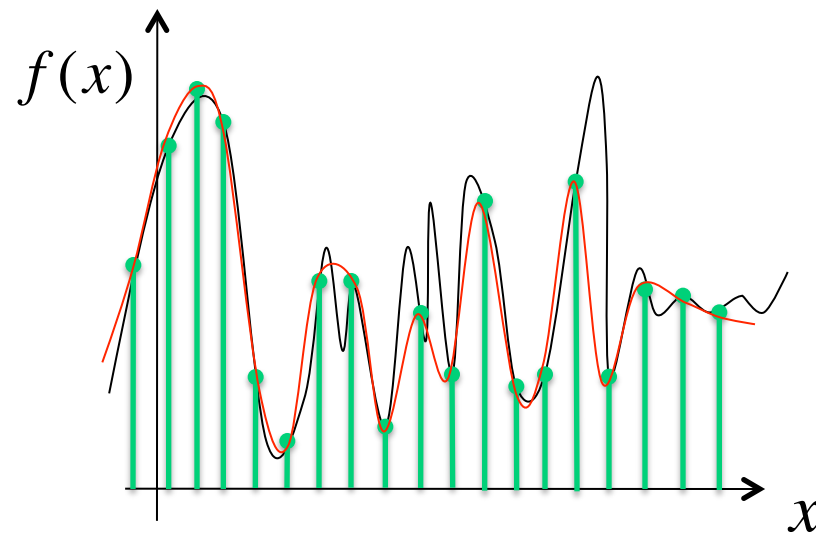
$$a_k = \frac{1}{2\omega_0} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{-2\pi j \left(\frac{k\omega}{2\omega_0} \right)} d\omega$$

Note: Though both equations have $\mathcal{F}(\omega)$, the Fourier coefficients a_k have a band-limited integral, which can be evaluated. Thus a_k becomes a function of ω .

The Importance of Nyquist Sampling Thrm



- This theorem provides a theoretical sampling rate at which we will incur (under certain conditions) no loss of information.



- High sampling rate leads to too much data.
- Low sampling rate leads to loss of information.

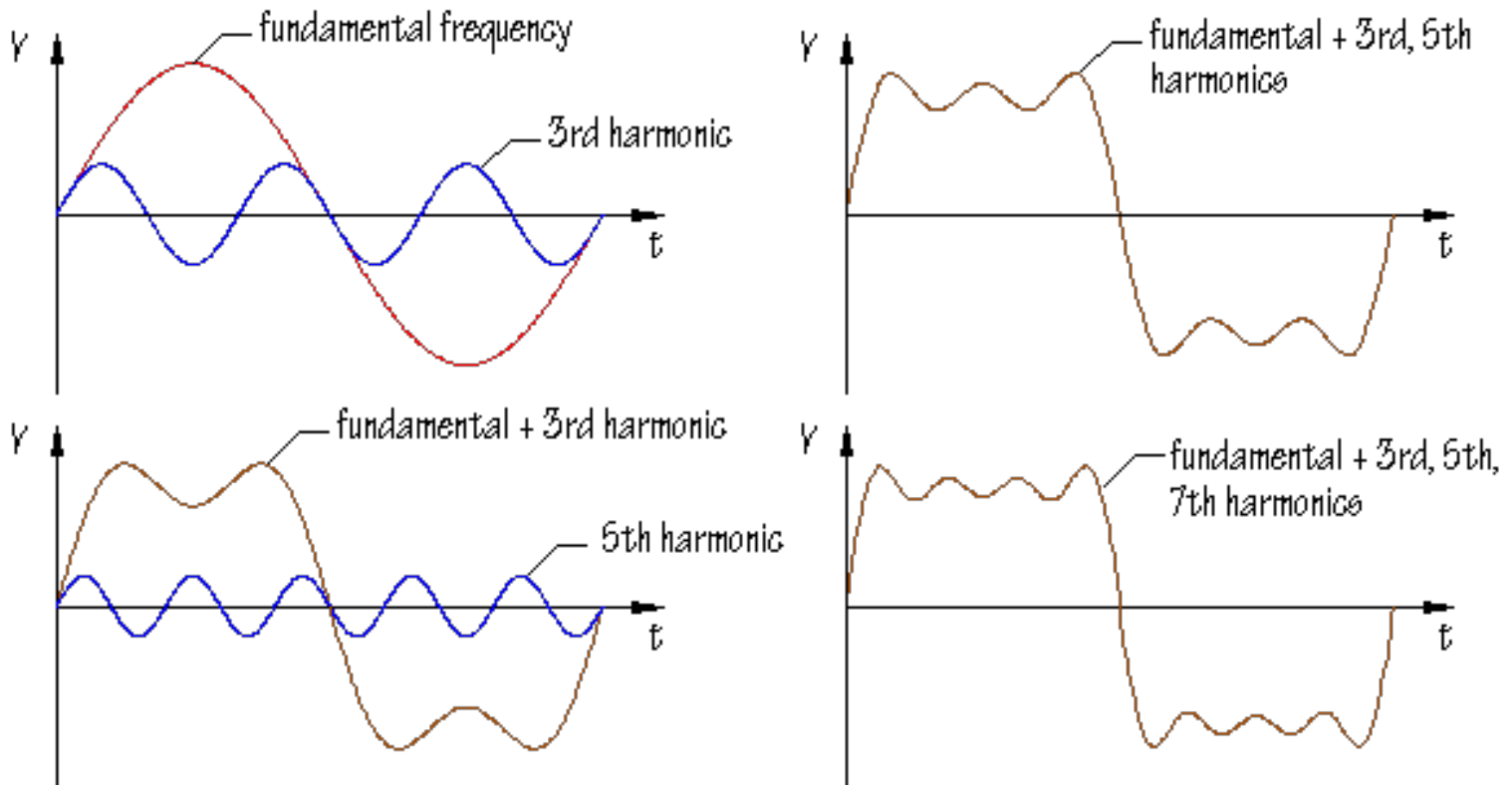


Band-Limited Function

- A function $f(x)$ is band-limited in the frequency range $(-B_x, B_x)$, if $\mathcal{F}(\omega) = 0$ for $|\omega| > \omega_0 = 2\pi B_x$ where ω_0 is the boundary frequency.
- What is so special about frequency band limited functions?
- They are restricted to a finite range of frequencies.
- Band limited \Rightarrow finite number of sin and cos terms
 \Rightarrow finite number of coefficients
 \Rightarrow signal can be reconstructed from a limited number of discrete samples.



Example of a Band-Limited Signal



Plot courtesy of <http://www.doctrronics.co.uk/signals.htm>



Nyquist Sampling Theorem

- Let $f(x)$ be a band-limited function in the frequency range $(-B_x, B_x)$.

- Then $f(x)$ is completely determined by the samples

$$f_k = f(k \Delta x) \quad \text{where } k = 0, \pm 1, \pm 2, \dots$$

if the sampling interval is chosen as

$$\Delta x \leq \frac{1}{2B_x} = \frac{\pi}{\omega_0}$$

- The original signal $f(x)$ can be reconstructed without any error using the following interpolation

$$f(x) = \sum_{k=-\infty}^{\infty} f_k \frac{\sin(2\pi B_x (x - k\Delta x))}{2\pi B_x (x - k\Delta x)} = \sum_{k=-\infty}^{\infty} f_k \operatorname{sinc}(2\pi B_x (x - k\Delta x))$$



Main Idea of Proof

- Goal: To show that by using the sampling rate recommended by Nyquist's sampling theorem, we incur no information loss.
- We want to show that the $f(x)$ we reconstruct from the samples f_k is identical to the original *band-limited* signal.
- We will use the Fourier Transform, the Inverse Fourier Transform and the Fourier Series to prove the theorem.
- Recall that for a band-limited signal $\mathcal{F}(\omega) = 0$ for $|\omega| > \omega_0 = 2\pi B_x$



Step 1

- If we had the Fourier Transform of the reconstructed signal, $\mathcal{F}(\omega)$, we could compute $f(x)$ via the Inverse Fourier Transform, as follows:

$$f(x) = \text{FT}^{-1}\{\mathcal{F}(\omega)\} = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{jx\omega} d\omega$$

- Problem: We don't have $\mathcal{F}(\omega)$.
- If we treat $f(x)$ as a periodic signal with period ω_0 , we can get $\mathcal{F}(\omega)$ using the Fourier Series representation.



Step 2

- In order to use the Fourier Series representation we need the Fourier coefficients.

$$a_k = \frac{1}{2\omega_0} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{-2\pi j \left(\frac{k\omega}{2\omega_0} \right)} d\omega$$

- Rewrite this equation so that it looks like an Inverse Fourier Transform ($f(x) = \text{FT}^{-1}\{\mathcal{F}(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\omega) e^{jx\omega} d\omega$).

$$a_k = \frac{\pi}{2\pi\omega_0} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{j\omega \left(\frac{-2\pi k}{2\omega_0} \right)} d\omega$$



Step 2 - continued

- The Fourier series coefficients are now:

$$a_k = \frac{1}{2\pi} \frac{\pi}{\omega_0} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{j\omega \left(\frac{-k\pi}{\omega_0} \right)} d\omega$$

- But according to the Inv. FT, $f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\omega) e^{j\omega x} d\omega$. Thus:

$$a_k = \frac{\pi}{\omega_0} f\left(-\frac{k\pi}{\omega_0}\right)$$

- The Nyquist Sampling theorem recommends a sampling rate of $\Delta x \leq \frac{1}{(2B_x)} = \frac{\pi}{\omega_0}$. If we use such a sampling rate:

$$a_k = \Delta x f(-k \Delta x)$$



Step 3

- Take the a_k and put them back in the Fourier series and hopefully we get the Fourier Series to look like an interpolation formula.

$$\mathcal{F}(\omega) = \sum_{k=-\infty}^{\infty} a_k e^{2\pi j \left(\frac{k\omega}{2\omega_0} \right)}$$

$$\mathcal{F}(\omega) = \sum_{k=-\infty}^{\infty} \Delta x f(-k \Delta x) e^{jk\omega\Delta x}$$

- Via a variable substitution we get:

$$\mathcal{F}(\omega) = \sum_{k=-\infty}^{\infty} \Delta x f(k \Delta x) e^{-jk\omega\Delta x}$$



Step 4

- Now we have an estimate of the Fourier Transform that we obtained directly from our discrete samples.
- We are ready to use the Inverse Fourier Transform to see which signal we reconstruct from these samples.

$$f(x) = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} \mathcal{F}(\omega) e^{jx\omega} d\omega = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} \left(\sum_{k=-\infty}^{\infty} \Delta x f(k \Delta x) e^{-jk\omega\Delta x} \right) e^{jx\omega} d\omega$$

$$f(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \Delta x f(k \Delta x) \int_{-\omega_0}^{\omega_0} e^{-jk\omega\Delta x} e^{jx\omega} d\omega$$

$$f(x) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \Delta x f(k \Delta x) \int_{-\omega_0}^{\omega_0} e^{j\omega(x-k\Delta x)} d\omega$$



Step 4 - continued

- We can then evaluate the integral

$$f(x) = \sum_{k=-\infty}^{\infty} \frac{\Delta x}{2\pi} f(k \Delta x) \left[\frac{e^{j\omega(x-k\Delta x)}}{j(x-k\Delta x)} \right]_{-\omega_0}^{\omega_0}$$

$$f(x) = \sum_{k=-\infty}^{\infty} \frac{\Delta x}{2\pi} f(k \Delta x) \left(\frac{1}{j} \right) \left(\frac{e^{j\omega_0(x-k\Delta x)} - e^{-j\omega_0(x-k\Delta x)}}{(x-k\Delta x)} \right)$$

- Recall that $\sin \theta = \frac{1}{2j} (e^{j\theta} - e^{-j\theta})$. Hence

$$f(x) = \sum_{k=-\infty}^{\infty} \frac{\Delta x}{2\pi} f(k \Delta x) \frac{2 \sin(\omega_0(x-k\Delta x))}{(x-k\Delta x)}$$



Step 4 – last part

- If we have $f_k = f(k \Delta x)$ and $\Delta x = \frac{1}{2B_x}$ and $\omega_0 = 2\pi B_x$

$$f(x) = \sum_{k=-\infty}^{\infty} f(k \Delta x) \frac{\Delta x \sin(\omega_0(x - k\Delta x))}{\pi (x - k\Delta x)}$$

$$f(x) = \sum_{k=-\infty}^{\infty} f_k \frac{\sin(2\pi B_x(x - k\Delta x))}{2\pi B_x(x - k\Delta x)}$$

- Thus, if we use the Nyquist sampling rate, we can reconstruct the original signal by interpolating the discrete samples.

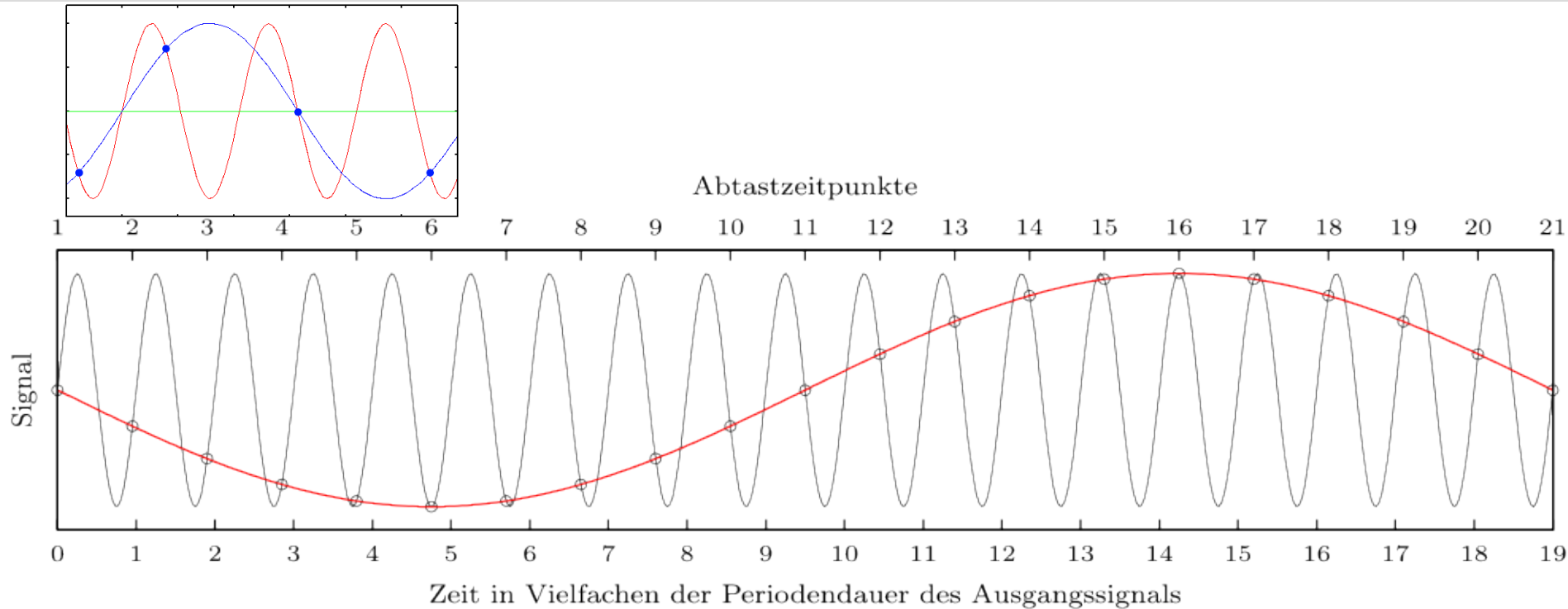


On the Nyquist Sampling Theorem

- So, the **precise** reconstruction of $f(x)$ requires:
 - an sampling interval $\Delta x = \frac{1}{2B_x}$
 - an infinite number of samples.
- In practice we are usually dealing with limited time, so we typically prefilter the signal and choose $\Delta x < \frac{1}{2B_x}$.
- Theorem: There is no function (in L_2), which is both band-limited and time-limited (except for the identity function).
- The smaller the function in the spatio-temporal domain, the larger it is in the frequency domain and vice versa.



Impact of Nyquist Sampling Theorem



Undersampling: Sampling with a sampling frequency which is lower than twice the highest frequency of the sampled signal.

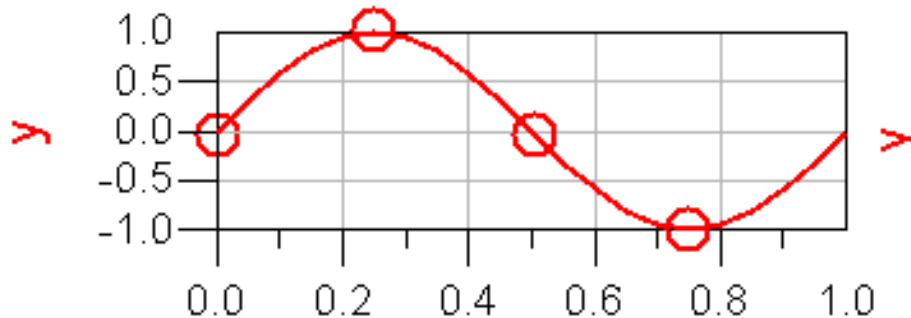
The high frequency waveform is perceived (is aliased) as a lower-frequency waveform.

Plots courtesy of http://music.columbia.edu/cmcc/musicandcomputers/chapter2/02_03.php and <http://de.wikipedia.org/wiki/Alias-Effekt>



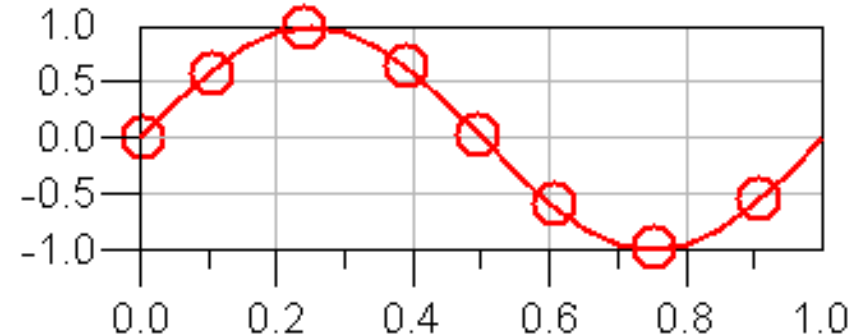
Impact of Nyquist Sampling Theorem (2)

Oversampling



X

Order=3, Oversample=1
Number of samples= $(3+1)*1=4$



X

Order=3, Oversample=2
Number of samples= $(3+1)*2=8$

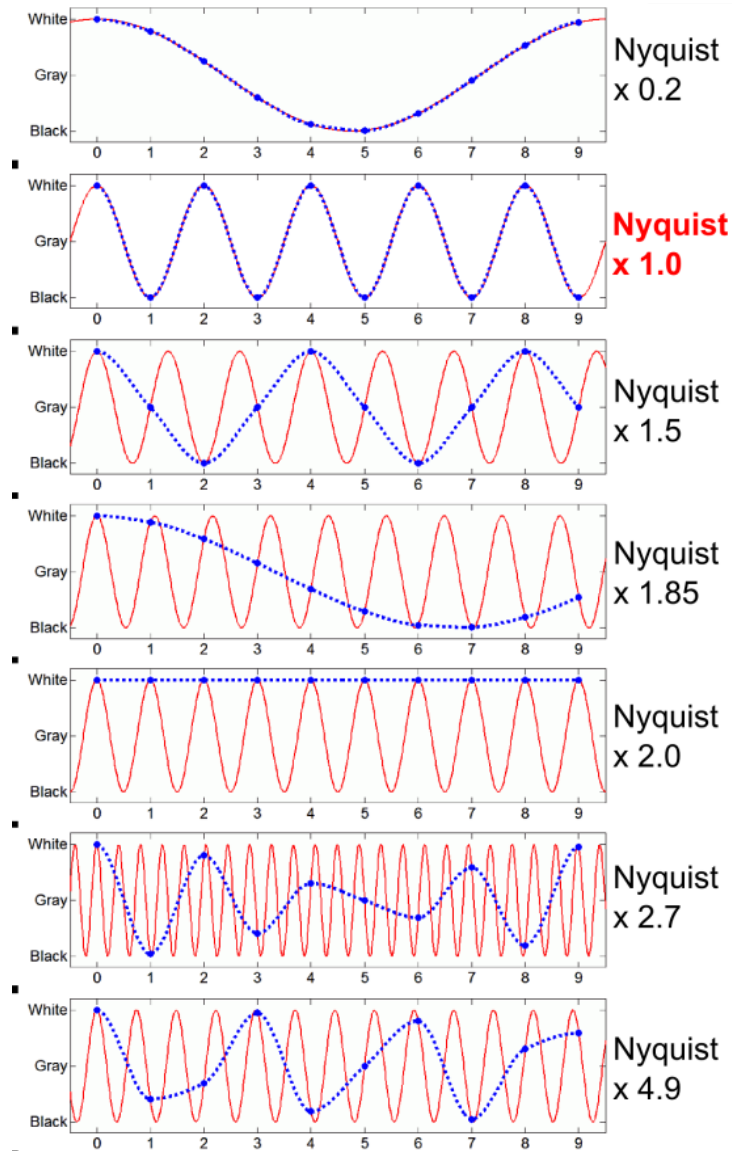
For each period of the input tone, the simulator will take x number of samples, determined by the values for Order and Oversample.

Oversampling: Sampling with a sampling frequency which is much higher than twice the highest frequency of the sampled signal.

It avoids aliasing, improves resolution and reduces noise.

Plot courtesy of <http://cp.literature.agilent.com/litweb/pdf/ads2003a/adshbapp/adshbapp.html>

Impact of Nyquist Sampling Theorem (3)

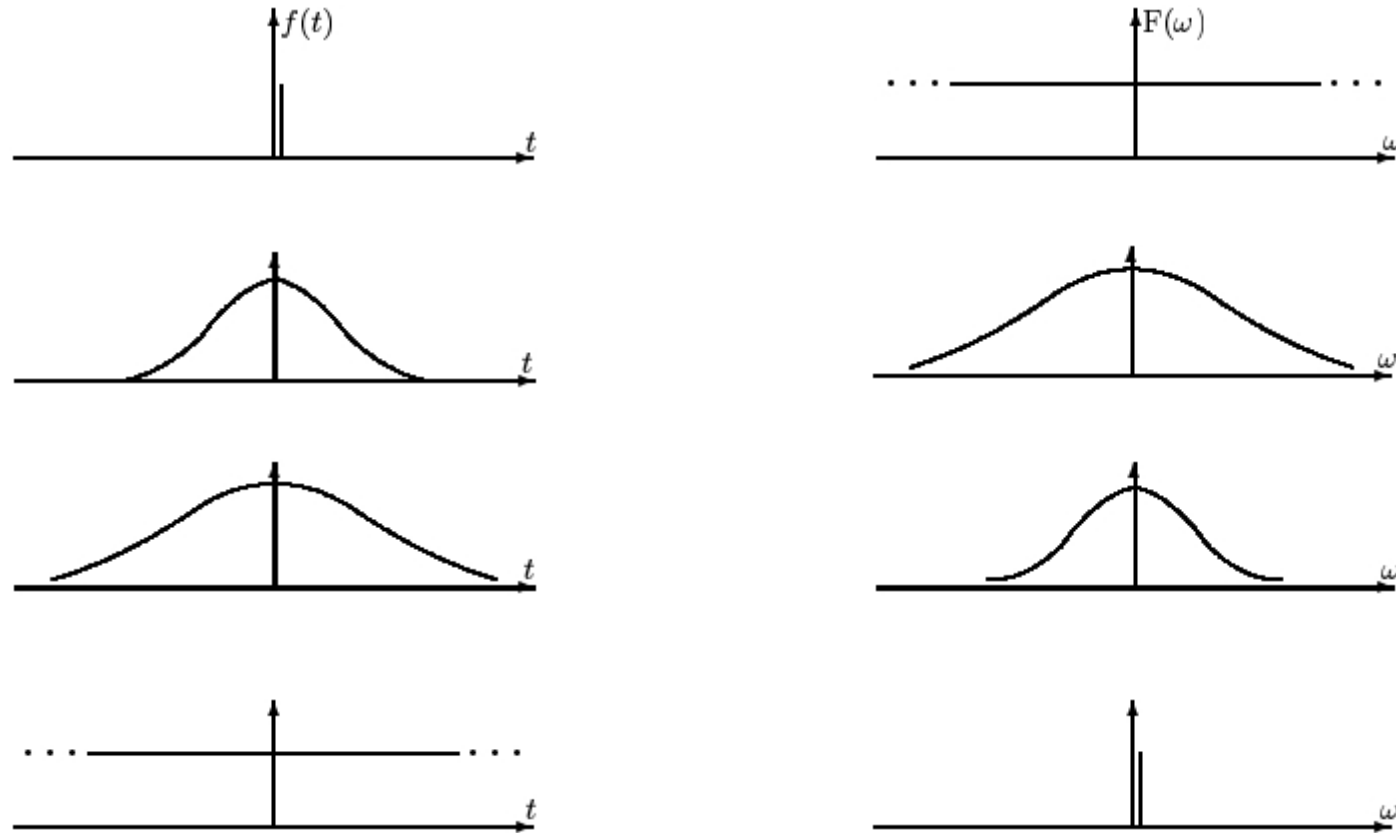


An example of how the captured signal (in blue) will look like compared to the original one (in red) as we increase the Nyquist sampling interval.

Plots courtesy of Francois Malan
<http://francoismalan.com/2012/07/anti-alias-moire-nikon-d800e/>



Temporal vs. Frequency Domain



- Compromise between accuracy of representation (high prec., many samples, small intervals) storage requirements (little storage, few samples, large intervals).



Sampling in 2D

- We need to sample in each direction.

- $f(x, y)$ is coded as $f_{j,k}$ where

$$f_{j,k} = f(x_0 + \Delta x, y_0 + \Delta y)$$

$$\text{where } j = 0, 1, \dots, M_x - 1 \quad k = 0, 1, \dots, M_y - 1$$

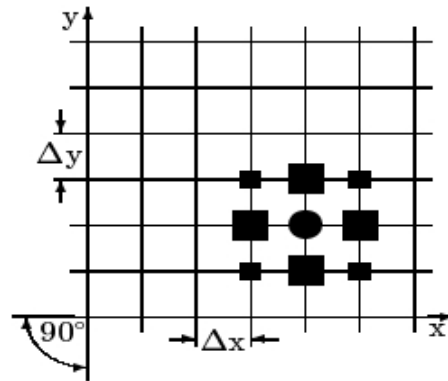
- We typically set $x_0 = y_0 = 0$ and $\Delta x = \Delta y = 1$ resulting in $f_{j,k} = f(j, k)$.
- Such a sampling setup results in a uniform sampling grid.



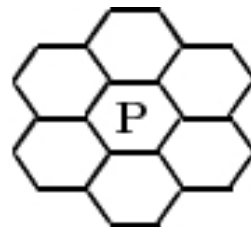
Uniform Sampling Grids

- There are 3 uniform sampling grids on a plane:

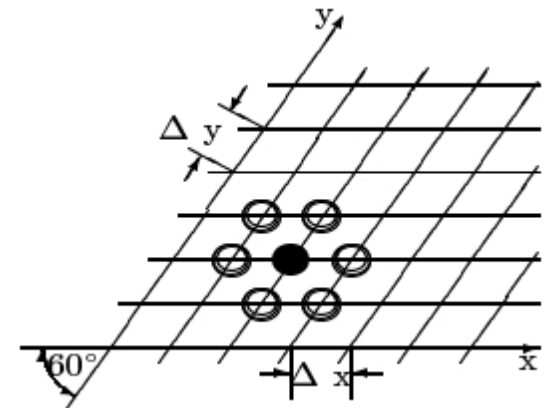
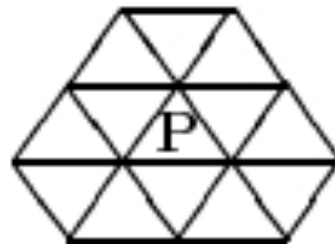
1. Square grid



2. Hexagon grid



3. Triangle-based grid



Sound Frequency Fun



- If you want to get a better feeling of the frequency components of sound signals, you may want to download the following Windows software:

http://www.zeitnitz.de/Christian/scope_de

- It uses the sound captured from your PC's microphone as input to a sound oscilloscope, which let's you visualize the sound-wave captured by your microphone.

Thanks to Felix Furtmayr for suggesting this tool.