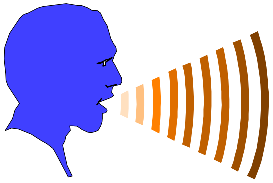
   

EE3414 Multimedia Communication Systems - I

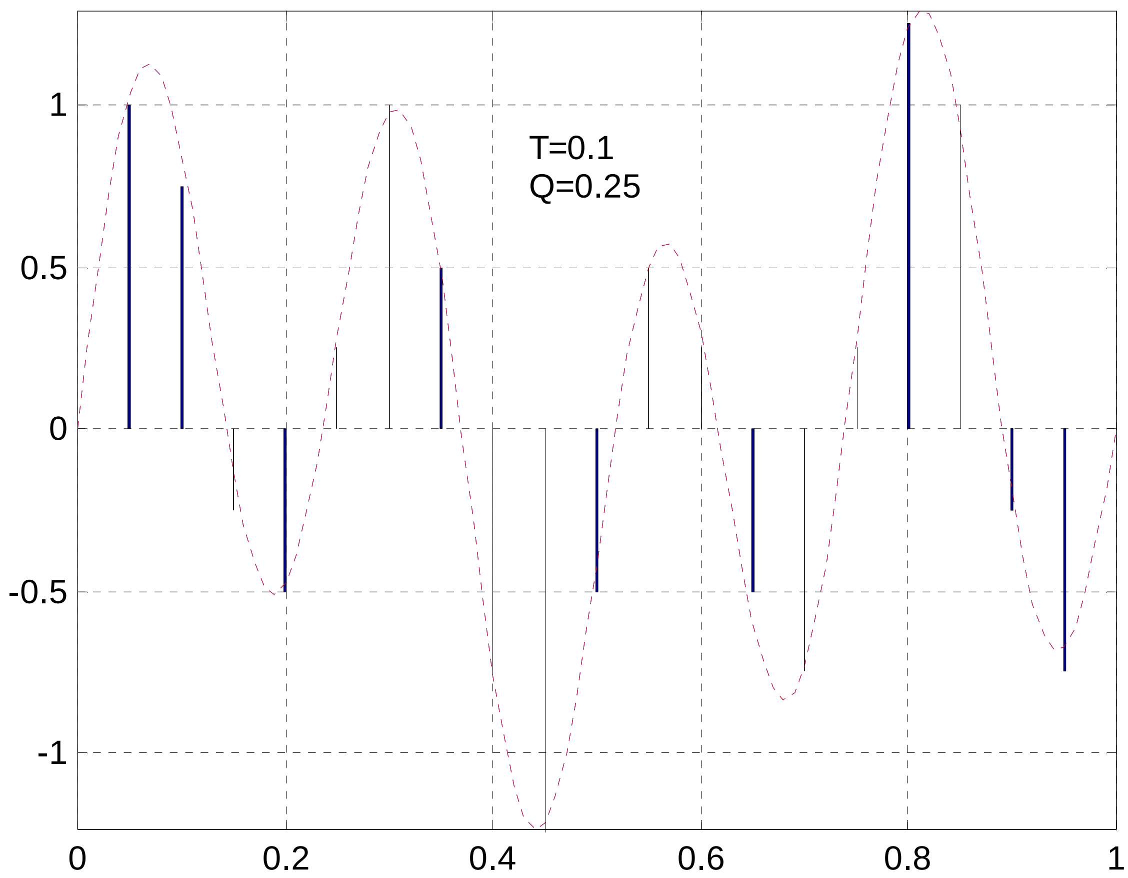
Sampling and Interpolation

Yao Wang
Polytechnic University, Brooklyn, NY11201
http://eeweb.poly.edu/~yao

Outline

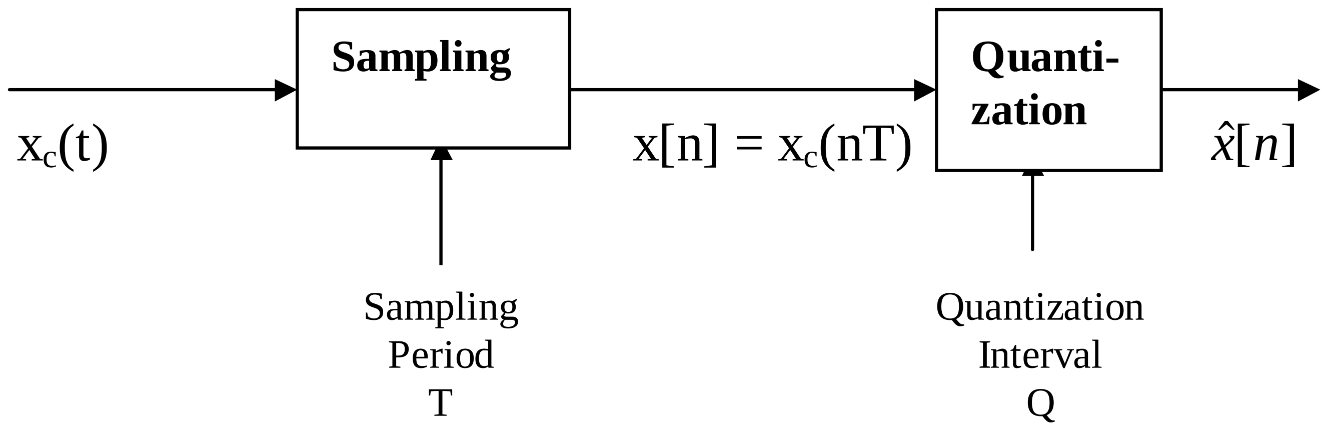
- Basics of sampling and quantization
  - A/D and D/A converters
- Sampling
  - Nyquist sampling theorem
  - Aliasing due to undersampling:
    - temporal and frequency domain interpretation
    - Sampling sinusoid signals
- Reconstruction from samples
  - Reconstruction using sample-and-hold and linear interpolation
  - Frequency domain interpretation (sinc pulse as interpolation kernel)
- Sampling rate conversion
  - Down sampling
  - Up sampling
  - Demonstration

Analog to Digital Conversion



A2D_plot.m

Two Processes in A/D Conversion



- Sampling: take samples at time *nT*
  - T: sampling period;
  - f_s = 1/T: sampling frequency

$$x[n] = x(nT), -\infty < n < \infty$$

- Quantization: map amplitude values into a set of discrete values $\pm pQ$
  - Q: quantization interval or stepsize

$$\hat{x}[n] = Q[x(nT)]$$

How to determine T and Q?

- T (or f_s) depends on the signal frequency range
  - A fast varying signal should be sampled more frequently!
  - Theoretically governed by the Nyquist sampling theorem
    - $f_s > 2 f_m$ (f_m is the maximum signal frequency)
    - For speech: $f_s >= 8$ *KHz;* For music: $f_s >= 44$ *KHz;*
- Q depends on the dynamic range of the signal amplitude and perceptual sensitivity
  - Q and the signal range D determine bits/sample R
    - $2^R=D/Q$
    - For speech: $R = 8$ *bits;* For music: $R =16$ *bits;*
- One can trade off T (or f_s) and Q (or R)
  - lower R -> higher f_s; higher R -> lower f_s
- We consider sampling in this lecture, quantization in the next lecture

Nyquist Sampling Theorem

- Theorem:
  - If $x_c(t)$ is bandlimited, with maximum frequency f_m (or $\omega_m = 2\pi f_m$)
  - and if $f_s = 1/T > 2 f_m$ or $\omega_s = 2\pi/T > 2\omega_m$
  - Then $x_c(t)$ can be reconstructed perfectly from $x[n]= x_c(nT)$ by using an ideal low-pass filter, with cut-off frequency at $f_s/2$
  - $f_{s0} = 2 f_m$ is called the *Nyquist Sampling Rate*

- Physical interpretation:
  - Must have at least two samples within each cycle!

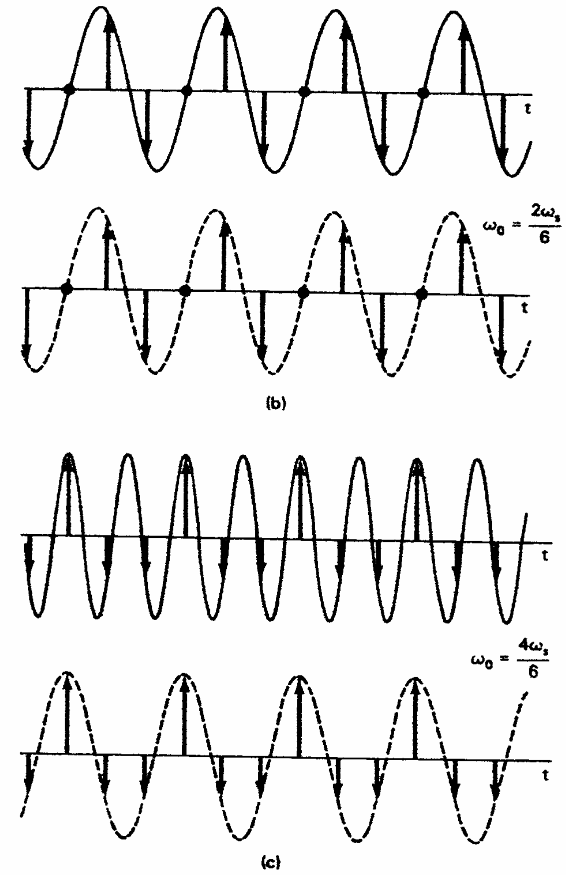
Temporal Domain Interpretation: Sampling of Sinusoid Signals

Sampling above Nyquist rate
$\omega_s=3\omega_m>\omega_{s0}$

Reconstructed =original

Sampling under Nyquist rate
$\omega_s=1.5\omega_m<\omega_{s0}$

Reconstructed \= original



Aliasing: The reconstructed sinusoid has a lower frequency than the original!

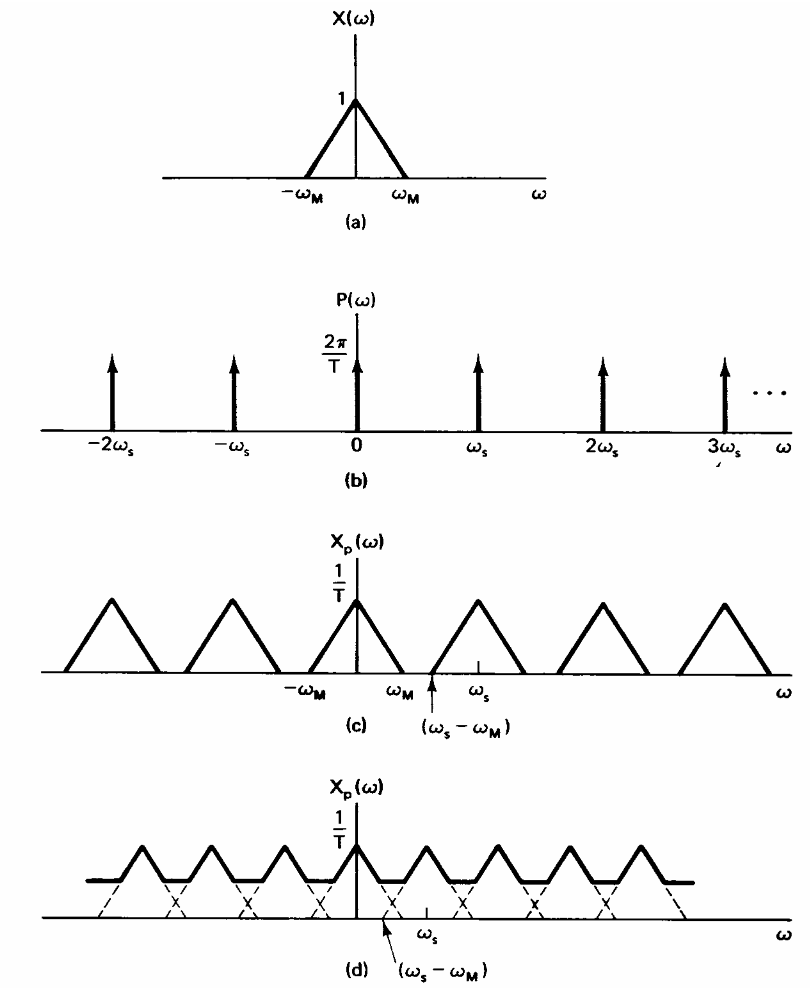
Frequency Domain Interpretation of Sampling

Original signal

Sampling impulse train

Sampled signal $\omega_s > 2\ \omega_m$

Sampled signal $\omega_s < 2\ \omega_m$ (Aliasing effect)

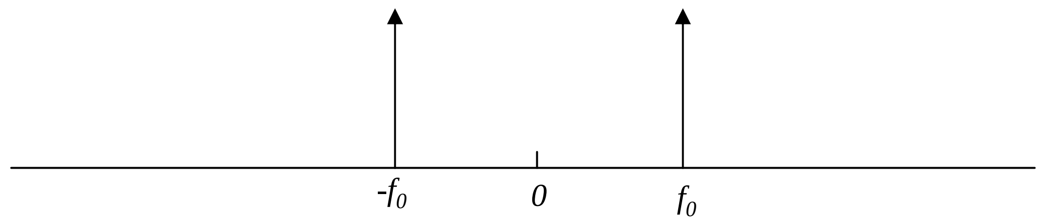


The spectrum of the sampled signal includes the original spectrum and its aliases (copies) shifted to *k* f_s *, k=+/- 1,2,3,…* The reconstructed signal from samples has the frequency components upto $f_s/2$.

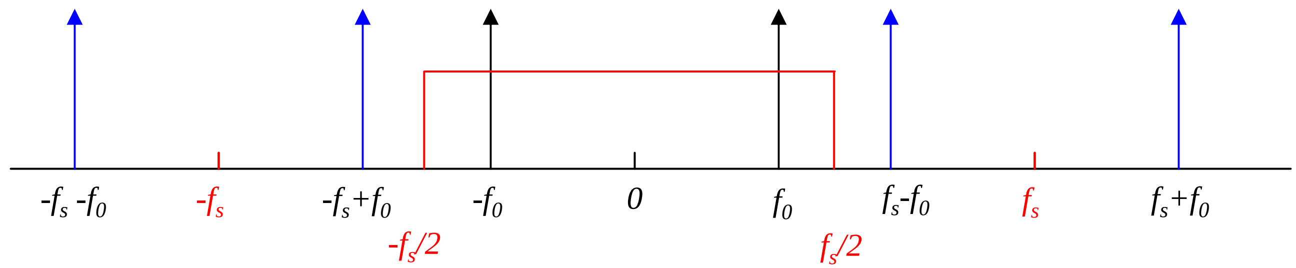
When $f_s < 2f_m$ *, aliasing occur.*

Sampling of Sinusoid in Frequency Domain

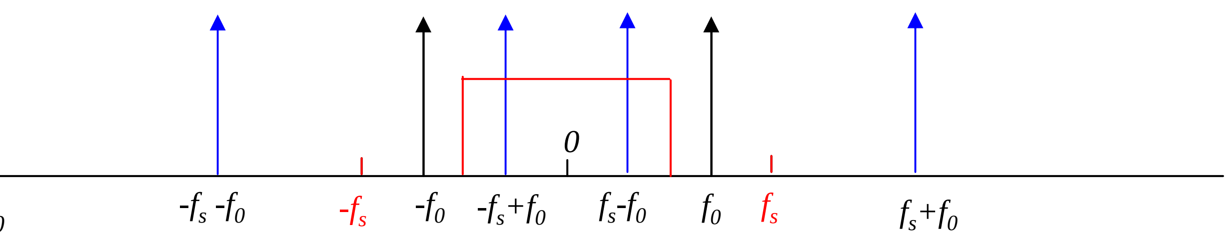
Spectrum of
$cos(2\pi f_0 t)$



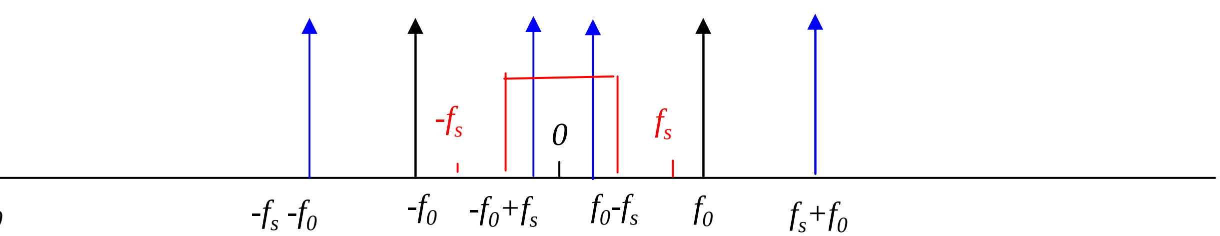
No aliasing
$f_s > 2f_0$
$f_s - f_0 > f_0$
Reconstructed signal: f_0



With aliasing
$f_0 < f_s < 2f_0$ (folding)
$f_s - f_0 < f_0$
Reconstructed signal: $f_s - f_0$



With aliasing
$f_s < f_0$ (aliasing)
$f_0 - f_s < f_0$
Reconstructed signal: $f_s - f_0$



More examples with Sinusoids

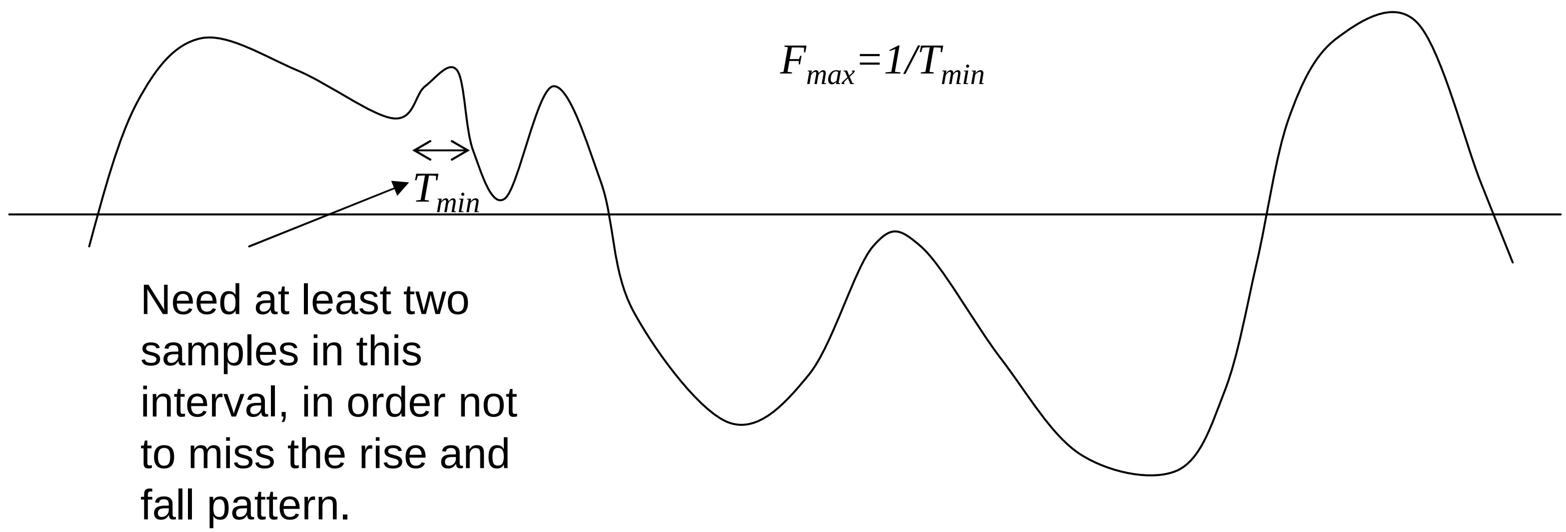
- Demo from DSP First, Chapter 4, aliasing and folding demo
  - Aliasing: $f_s < f_m$ *(perceived frequency:* $f_m - f_s$*)*
  - Folding: $f_m < f_s < 2f_m$ *(perceived frequency:* $f_s - f_m$*)*
  - No need to distinguish these two phenomena. Both lead to a false frequency lower than the original frequency

Strobe Movie

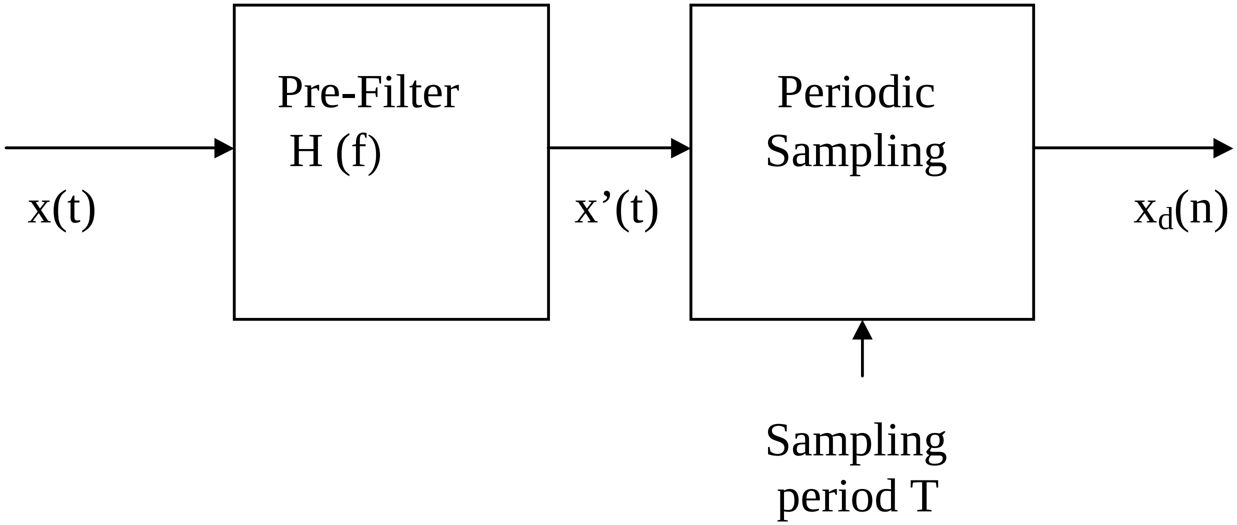
- From DSP First, Chapter 4, Demo on “Strobe Movie”

How to determine the necessary sampling frequency from a signal waveform?

- Given the waveform, find the shortest ripple, there should be at least two samples in the shortest ripple
- The inverse of its length is approximately the highest frequency of the signal



Sampling with Pre-Filtering

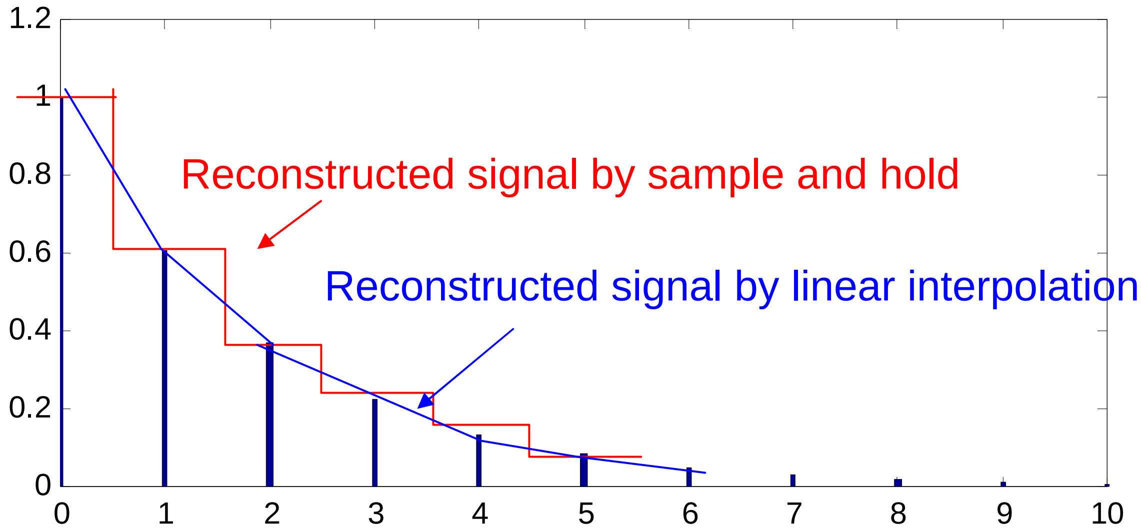


- If $f_s < 2f_m$, aliasing will occur in sampled signal
- To prevent aliasing, pre-filter the continuous signal so that $f_m<f_s/2$
- Ideal filter is a low-pass filter with cutoff frequency at $f_s/2$ (corresponding to sync functions in time)
- Common practical pre-filter: averaging within one sampling interval

How to Recover Continuous Signals from Samples?

- Connecting samples using interpolation kernels
  - Sampling and hold (rectangular kernels)
  - Linear interpolation (triangular kernels)
  - High order kernels
  - Ideal kernel: sinc function

Sample-and-Hold vs. Linear Interpolation

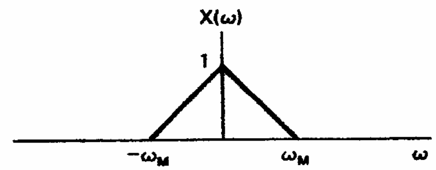


Reconstruction Using Different Kernels

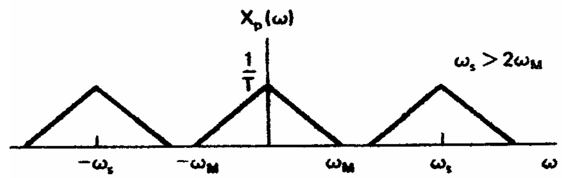
- Demo from DSP First, Chapter 4, demo on "reconstruction"

Frequency domain interpretation

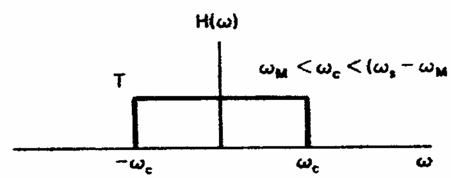
Original signal



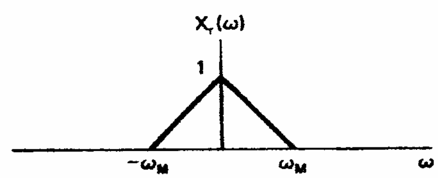
Sampled signal
$\omega_s > 2\ \omega_m$



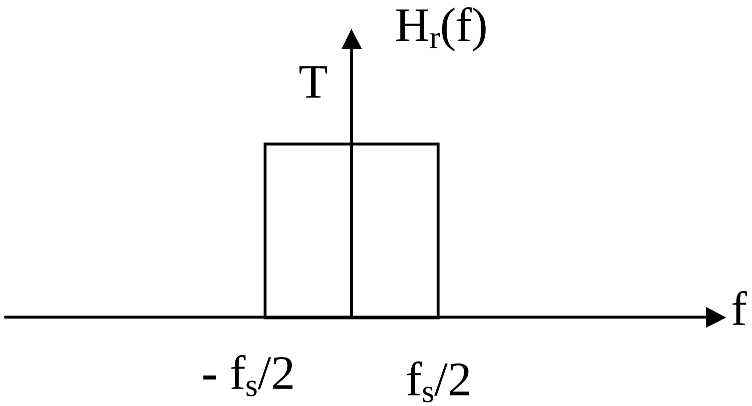
Ideal
reconstruction filter
(low-pass)

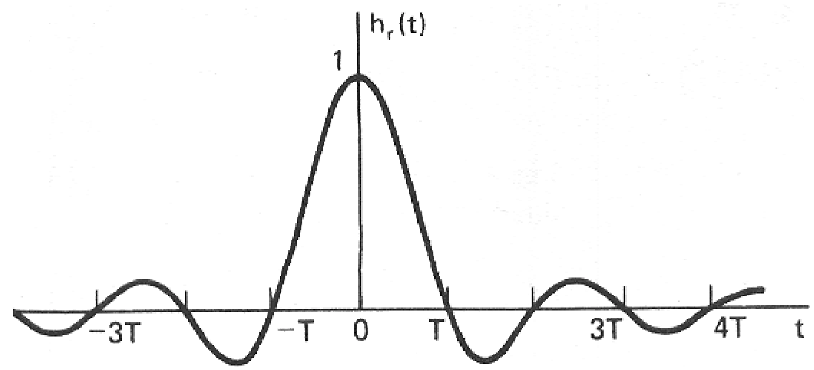


Reconstructed signal
(=Original signal)



Ideal Interpolation Filter





$$H_r(f) = \begin{cases} T & |f| < f_s/2 \\ 0 & \text{otherwise} \end{cases}$$

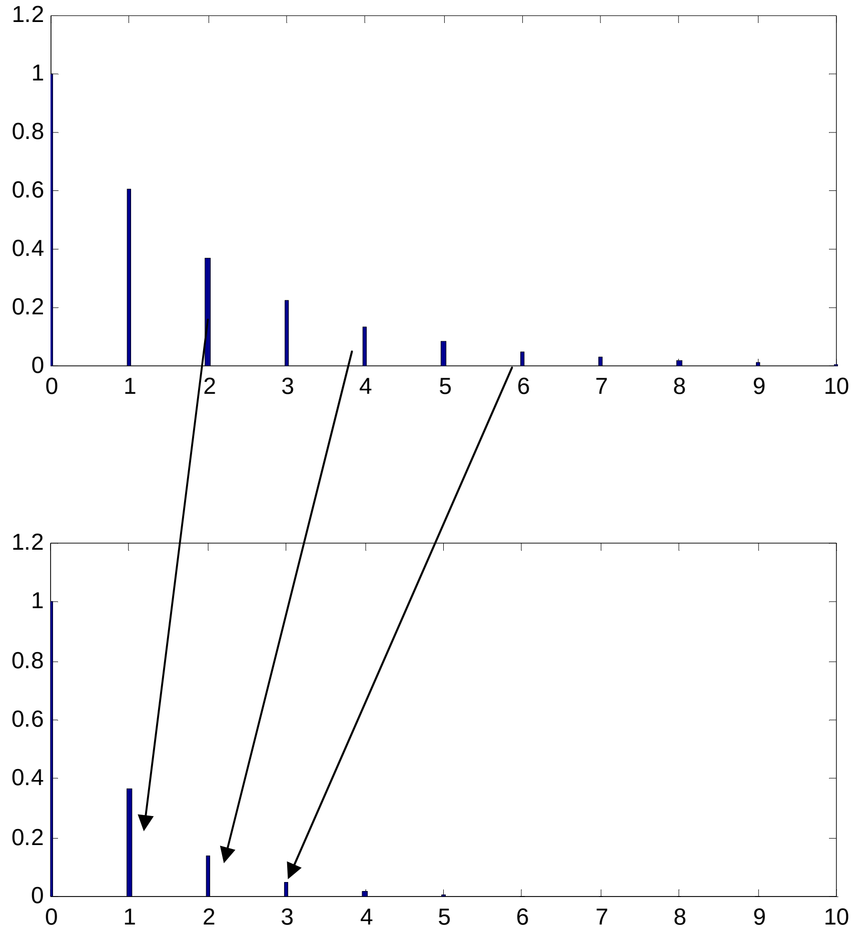
$$h_r(t) = \frac{\sin \pi t/T}{\pi t/T}$$

$$x_r(t) = x_s(t) * h_r(t) = \sum_{n=-\infty}^{\infty} x[n] h_r(t - nT) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}$$

Sampling Rate Conversion

- Given a digital signal, we may want to change its sampling rate
  - Necessary for image display when original image size differs from the display size
  - Necessary for converting speech/audio/image/video from one format to another
  - Sometimes we reduce sample rate to reduce the data rate
- Down-sampling: reduce the sampling rate
- Up-Sampling: increase the sampling rate

Down-Sampling Illustration

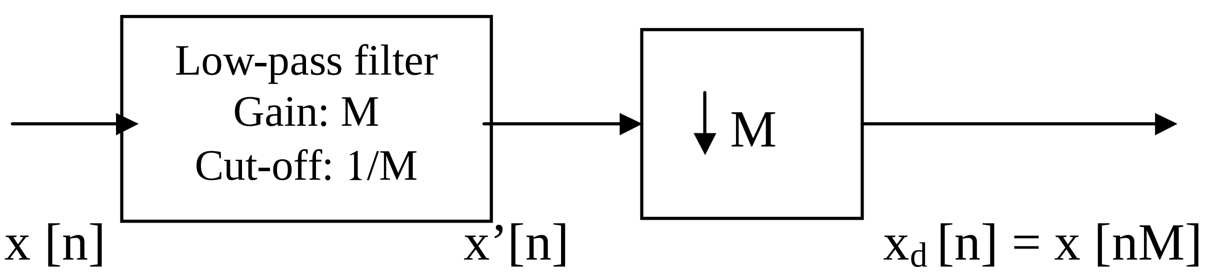


Down-sampling by a factor of 2 = take every other sample

To avoid aliasing of any high frequency content in the original signal, should smooth the original signal before down-sampling -- Prefiltering

Down Sampling by a Factor of M

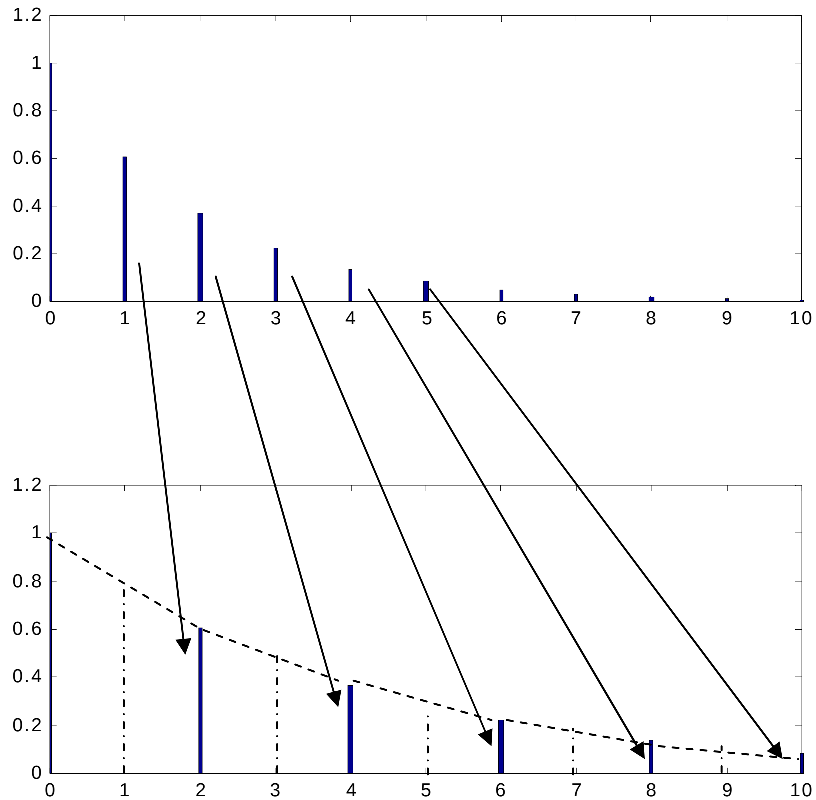
- Take every *M-th* sample from existing samples
  - $T' = MT$, $f_s' = f_s/M$
- Should apply a prefilter to limit the bandwidth of the original signal to 1/M-th of the original
- Without prefiltering, aliasing occur in the down-sampled signal.
- Ideal prefilter: low pass filter with cut-off frequency at 1/M (maximum digital frequency=1, corresponding to fs/2)
- Practical filter: averaging or weighted average over a neighborhood



Down-Sampling Example

- Given a sequence of numbers, down-sample by a factor of 2,
  - Original sequence: 1,3,4,7,8,9,13,15…
  - Without prefiltering, take every other sample:
    - 1,4,8,13,…
  - With 2-sample averaging filter
    - Filtered value=0.5*self+0. 5*right, filter h[n]=[0.5,0.5]
    - Resulting sequence:
      - 2, 5.5,8.5,14,…
  - With 3-sample weighted averaging filter
    - Filtered value=0.5*self+0.25*left+0.25*right, filter h[n]=[0.25,0.5,0.25]
    - Resulting sequence (assuming zeros for samples left of first):
      - 1.25, 4.5,8,12.5,…

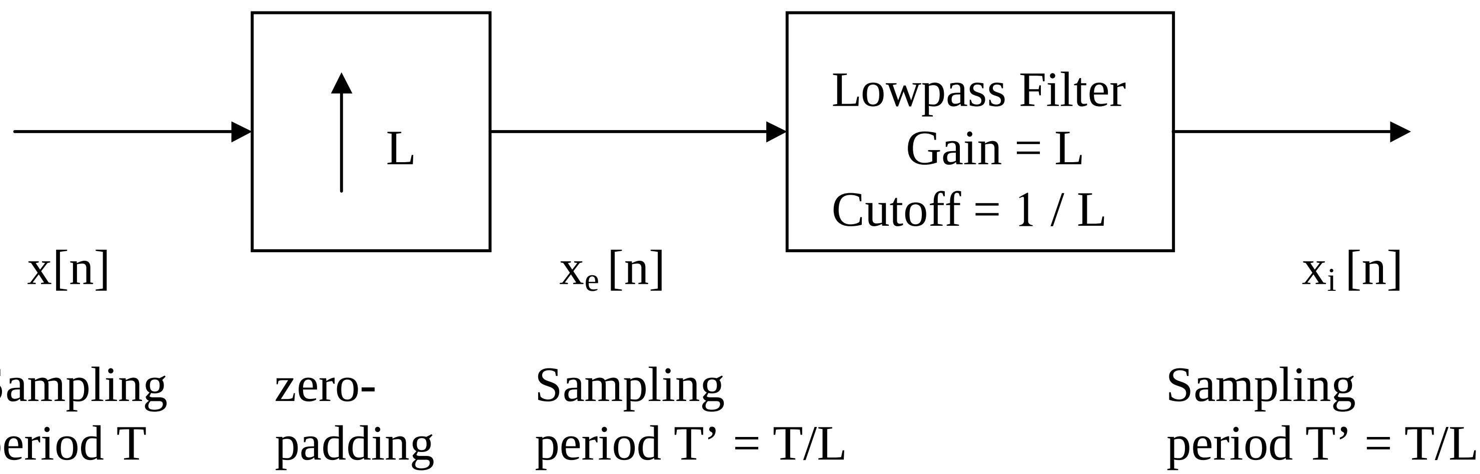
Upsampling by linear interpolation



Missing samples need to be filled from neighboring available samples using interpolation filter

Up Sampling by a Factor of L

- Insert *L-1* samples between every two existing samples
    - *T'=T/L, fs'=fs*L*
  - The estimation of the missing samples from original samples is known as interpolation
- Interpolation can be decomposed into two steps
  - Zero-padding: insert L-1 zeros in between every two samples
  - Low-pass filtering: to estimate missing samples from neighbors
  - Simplest interpolation filter: linear interpolation



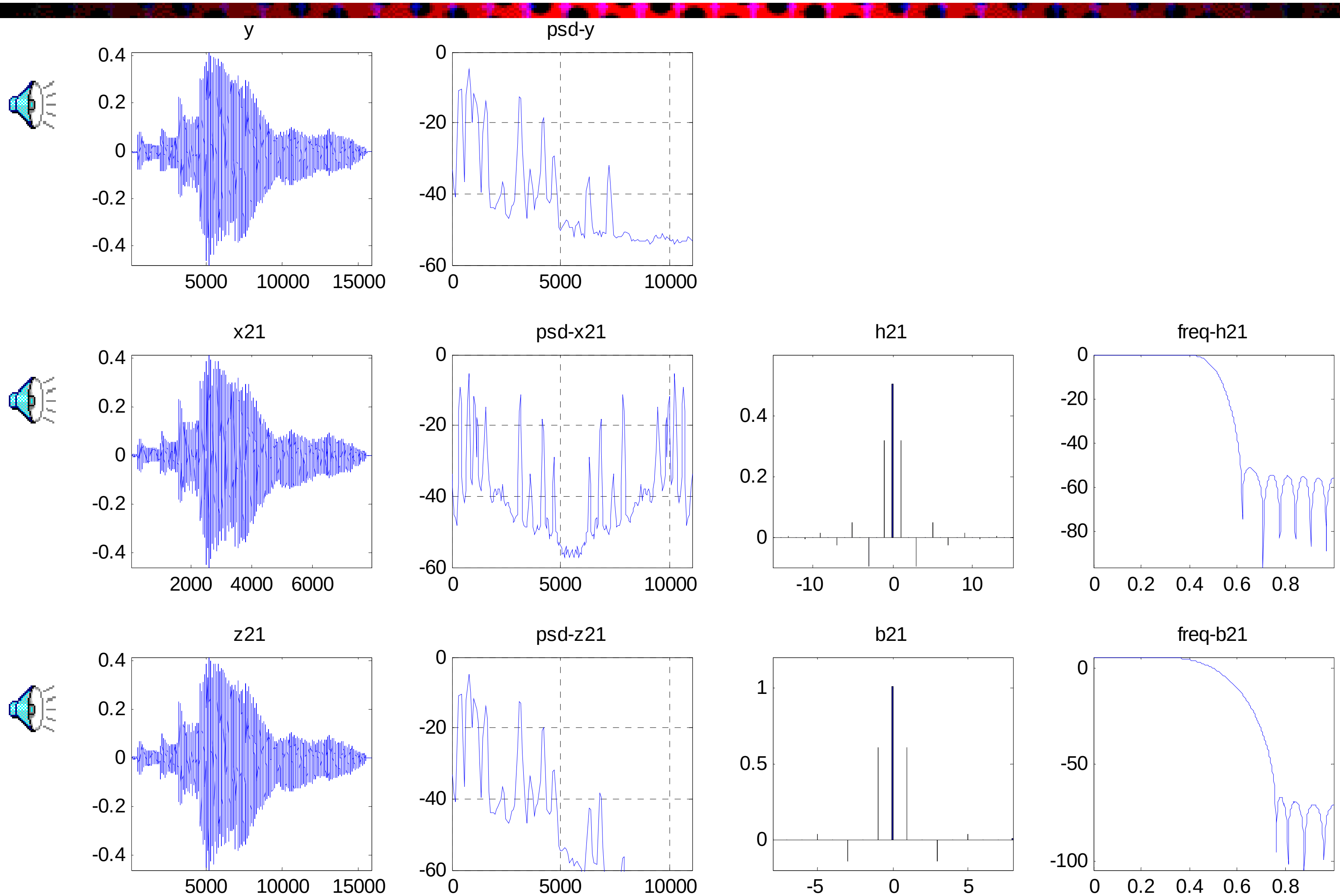
Up-Sample Example

- Given a sequence of numbers, up-sample by a factor of 2
  - Original sequence: 1,3,4,7,8,9,13,15…
  - Zero-padding:
    - 1,0,3,0,4,0,7,0,…
  - Sample and hold
    - Repeat the left neighbor, filter h[n]=[1,1]
    - 1,1,3,3,4,4,7,7,…
  - With linear interpolation
    - New sample=0.5*left+0. 5*right, filter h[n]=[0.5,1,0.5]
    - Resulting sequence:
      - 1,2,3,3.5,4,5.5,7,8,…,…

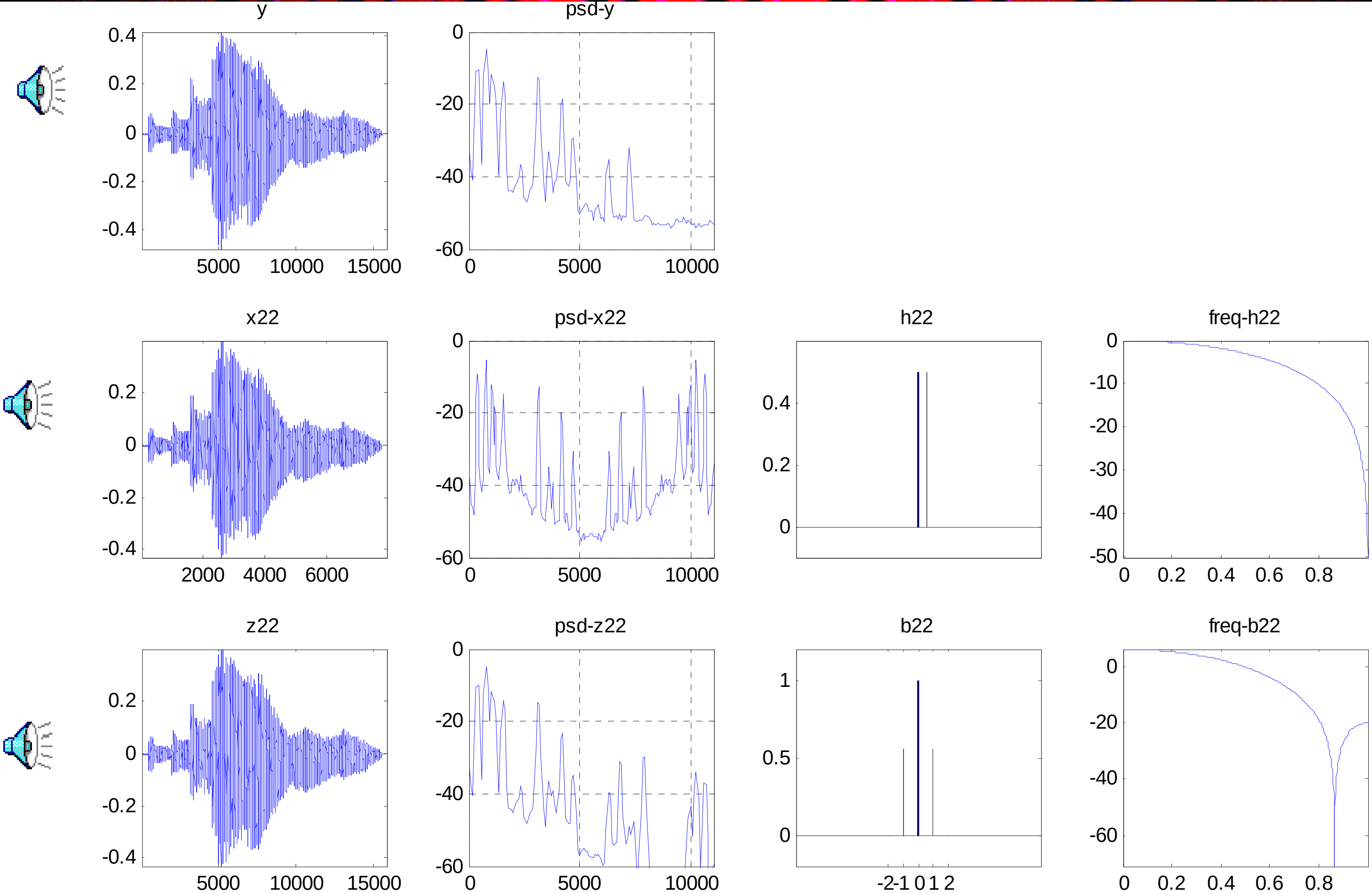
Demonstration

- Demonstrate the effect of down-sampling with different pre-filters, and up-sampling with different interpolation filters
- Compare both sound quality and frequency spectrum
- Matlab code (sampling_demo.m)

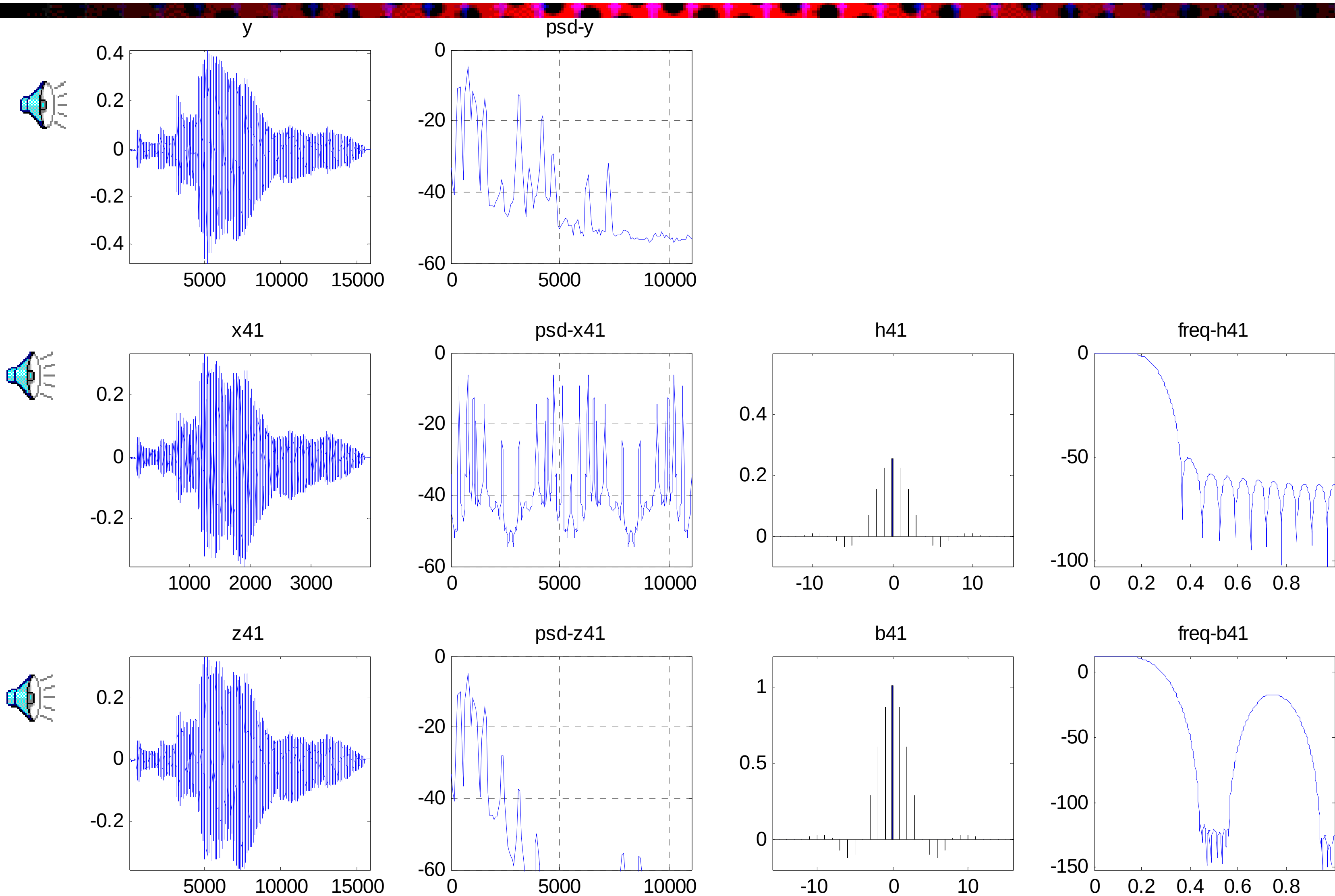
Down-2 followed by up-2, both using good filters



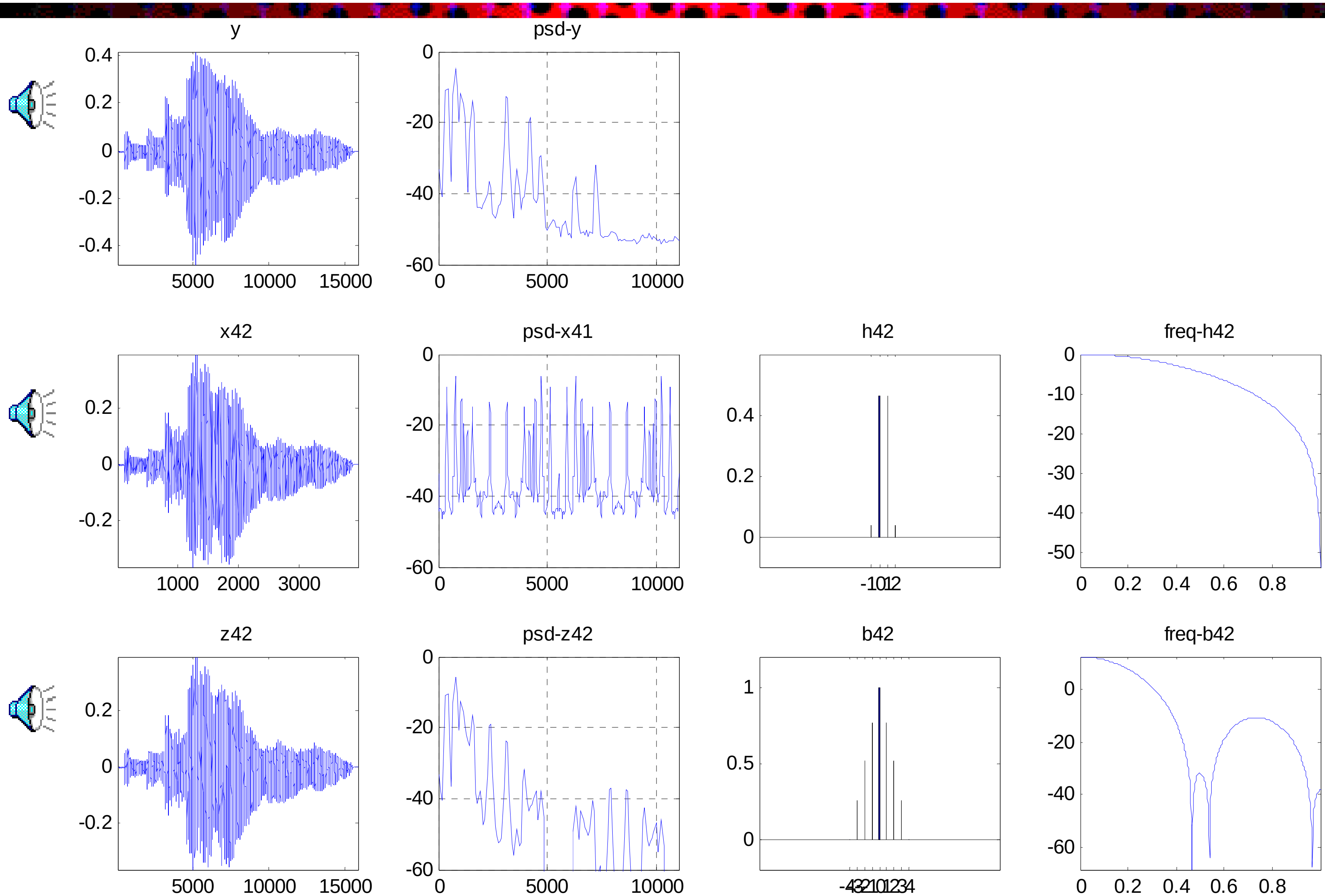
Down-2 followed by up-2, both using bad filters



Down-4 followed by up-4, both using good filters



Down-4 followed by up-4, both using bad filters



MATLAB Code

- Go through the code

What Should You Know (I)

- Sampling:
  - Know the minimally required sampling rate:
    - $f_s > 2\, f_{max}$, $T_s < T_{0,min}/2$
    - Can estimate $T_{0,min}$ from signal waveform
  - Can illustrate samples on a waveform and observe whether the signal is under-sampled.
  - Can plot the spectrum of a sampled signal
    - The sampled signal spectrum contains the original spectrum and its replicas (aliases) at kf_s, $k=+/- 1,2,\ldots.$
    - Can determine whether the sampled signal suffers from aliasing
  - Understand why do we need a prefilter when sampling a signal
    - To avoid alising
    - Ideally, the filter should be a lowpass filter with cutoff frequency at $f_s/2$.
  - Can show the aliasing phenomenon when sampling a sinusoid signal using both temporal and frequency domain interpretation

What Should You Know (II)

- Interpolation:
  - Can illustrate sample-and-hold and linear interpolation from samples.
  - Understand why the ideal interpolation filter is a lowpass filter with cutoff frequency at $f_s/2$.
  - Know the ideal interpolation kernel is the sinc function.
  - Interpolation using the sinc kernel is NOT required

- Sampling Rate Conversion:
  - Know the meaning of down-sampling and upsampling
  - Understand the need for prefiltering before down-sampling
    - To avoid aliasing
    - Know how to apply simple averaging filter for downsampling
  - Can illustrate up-sampling by sample-and-hold and linear interpolation

References

- McClellan, Schafer and Yoder, *DSP First*, Chap. 4
  - Has good conceptual / graphical interpretation

  (copies provided, Sec. 4.3,4.5 not required)
- Y. Wang, *EL514 Lab Manual*, Exp2: Voice and audio digitization and sampling rate conversion. Sec. 1,2. (copy provided)
- Oppenheim and Willsky, *Signals and Systems*, Chap. 7.
  - Optional reading (More depth in frequency domain interpretation)