MULTIRATE SIGNAL PROCESSING

Multirate Signal Processing

- Definition: Signal processing which uses more than one sampling rate to perform operations
- *Upsampling* increases the sampling rate
- *Downsampling* reduces the sampling rate
- Reference: *Digital Signal Processing,* DeFatta, Lucas, and Hodgkiss

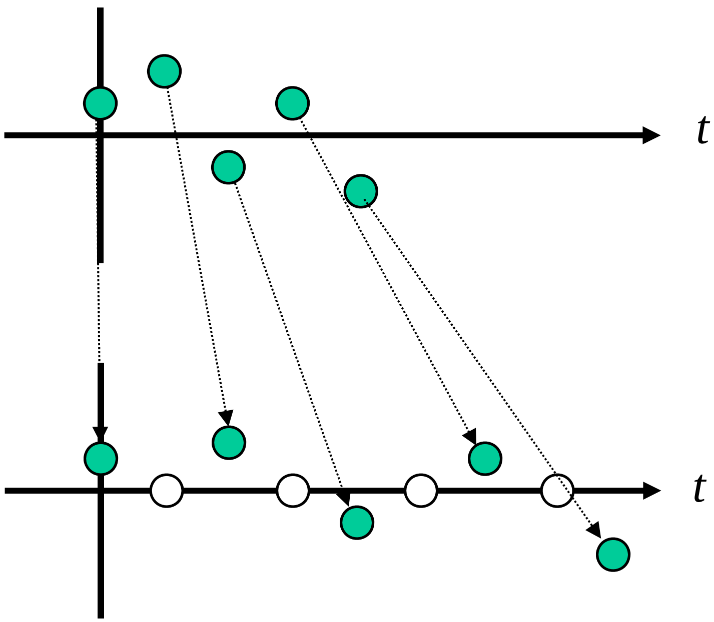
Multirate Signal Processing

- Advantages of lower sample rates
  - May require less processing
  - Likely to reduce power dissipation, $P = C\,V^2 f$, where f is frequently directly proportional to the sample rate
  - Likely to require less storage
- Advantages of higher sample rates
  - May simplify computation
  - May simplify surrounding analog and RF circuitry
- Remember that information up to a frequency f requires a sampling rate of at least $2f$. This is the *Nyquist sampling rate.*
  - Or we can equivalently say the Nyquist sampling rate is ½ the sampling frequency, f_s

Upsampling

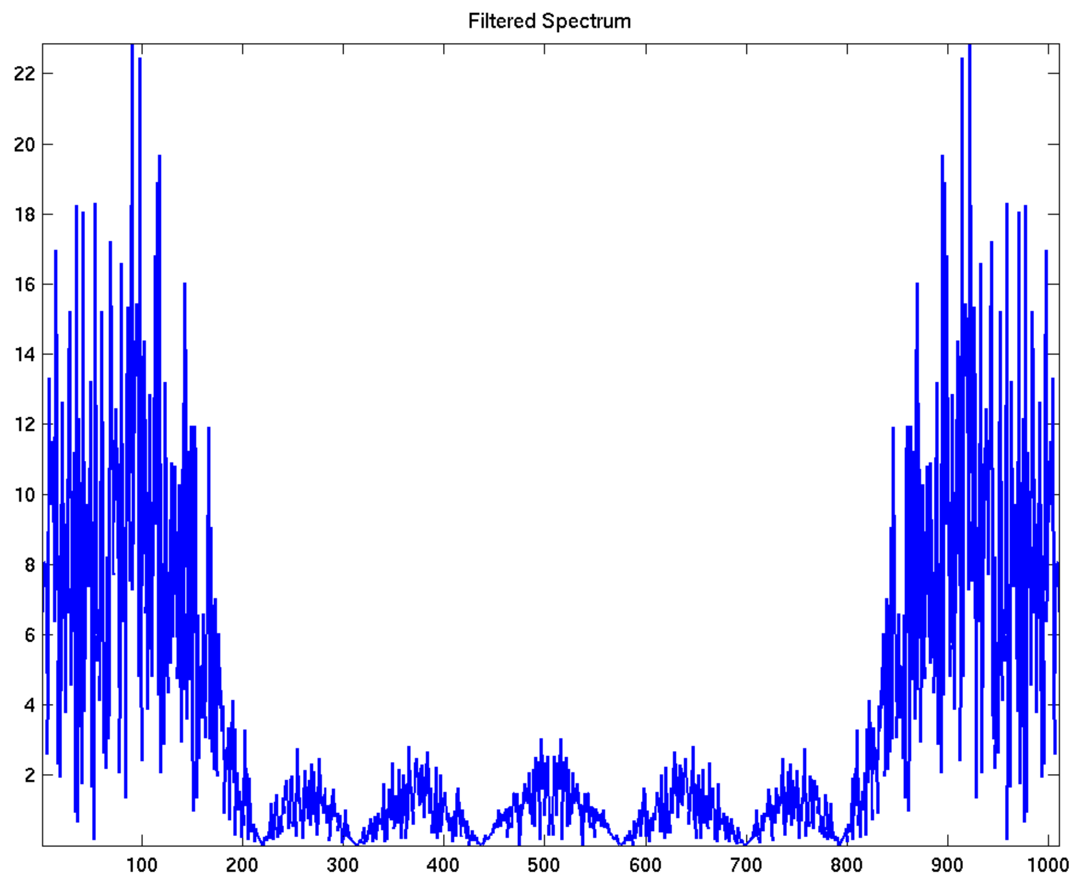
Upsampling or Interpolation

- For an upsampling by a factor of *I*, add *I*-1 zeros between samples in the original sequence
- An upsampling by a factor *I* is commonly written $\uparrow I$
  For example, upsampling by two: $\uparrow 2$
- Obviously the number of samples will approximately double after $\uparrow 2$
- Note that if the sampling frequency doubles after an upsampling by two, that the original sample sequence will occur at the same points in time



Original Signal Spectrum

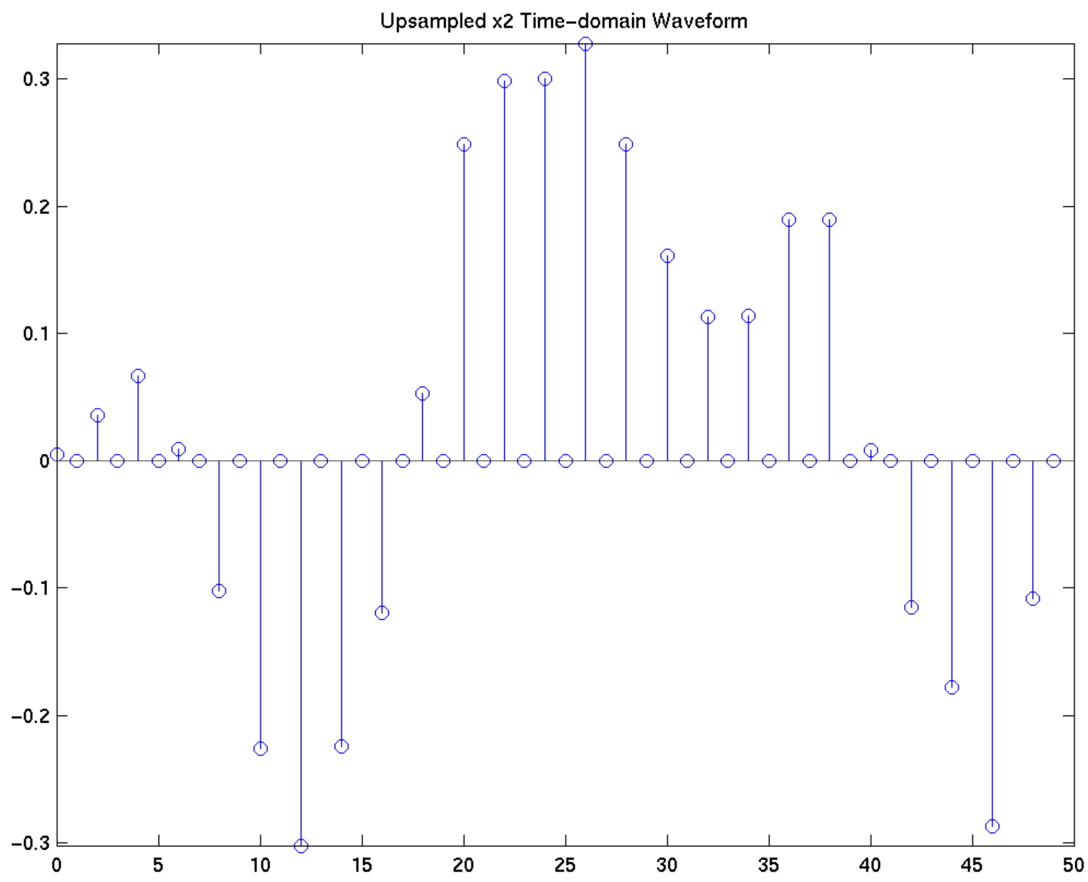
- Example signal with most energy near DC
- Notice 5 spectral “bumps” between large signal “bumps”



π 2π

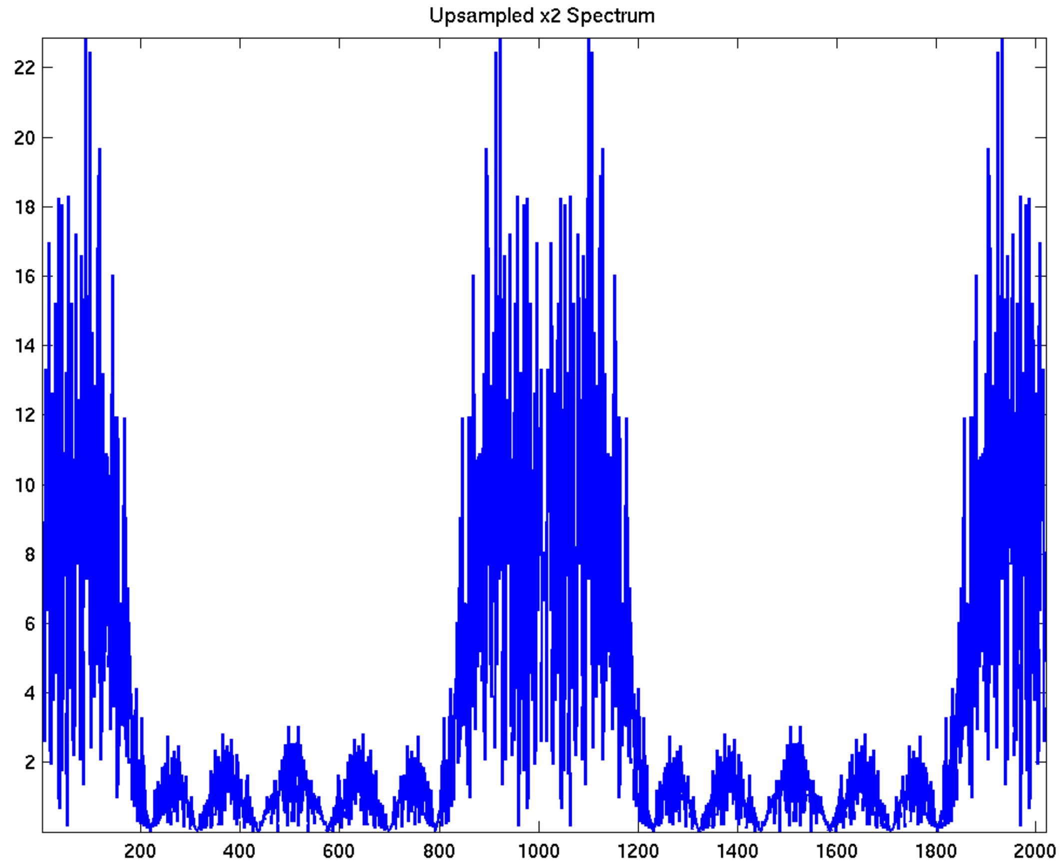
Upsampled Signal (Time)

- One zero is inserted between the original samples for 2x upsampling



Upsampled Signal Spectrum (Frequency)

- Spectrum of 2x upsampled signal
- Notice the location of the (now somewhat compressed) five "bumps" on each side of π



π 2π

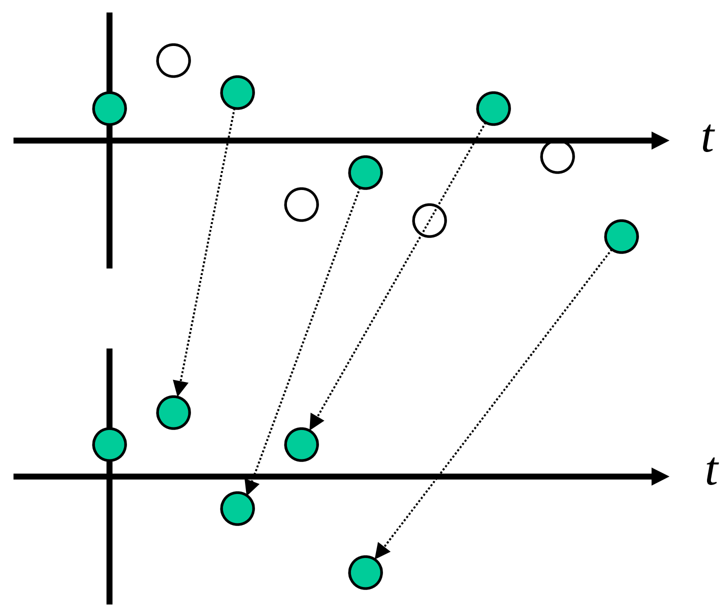
Post Upsampling Processing

- We likely want to attenuate images centered at π from $\pi/2$ to $3\pi/2$ because these are artificial artifacts caused by the upsampling process
- A low-pass filter will keep the original desired signal and remove the artifacts ("image") from $\pi/2$ to $3\pi/2$
- This low-pass filter is so common, that it has its own name: an *anti-image filter*

Downsampling

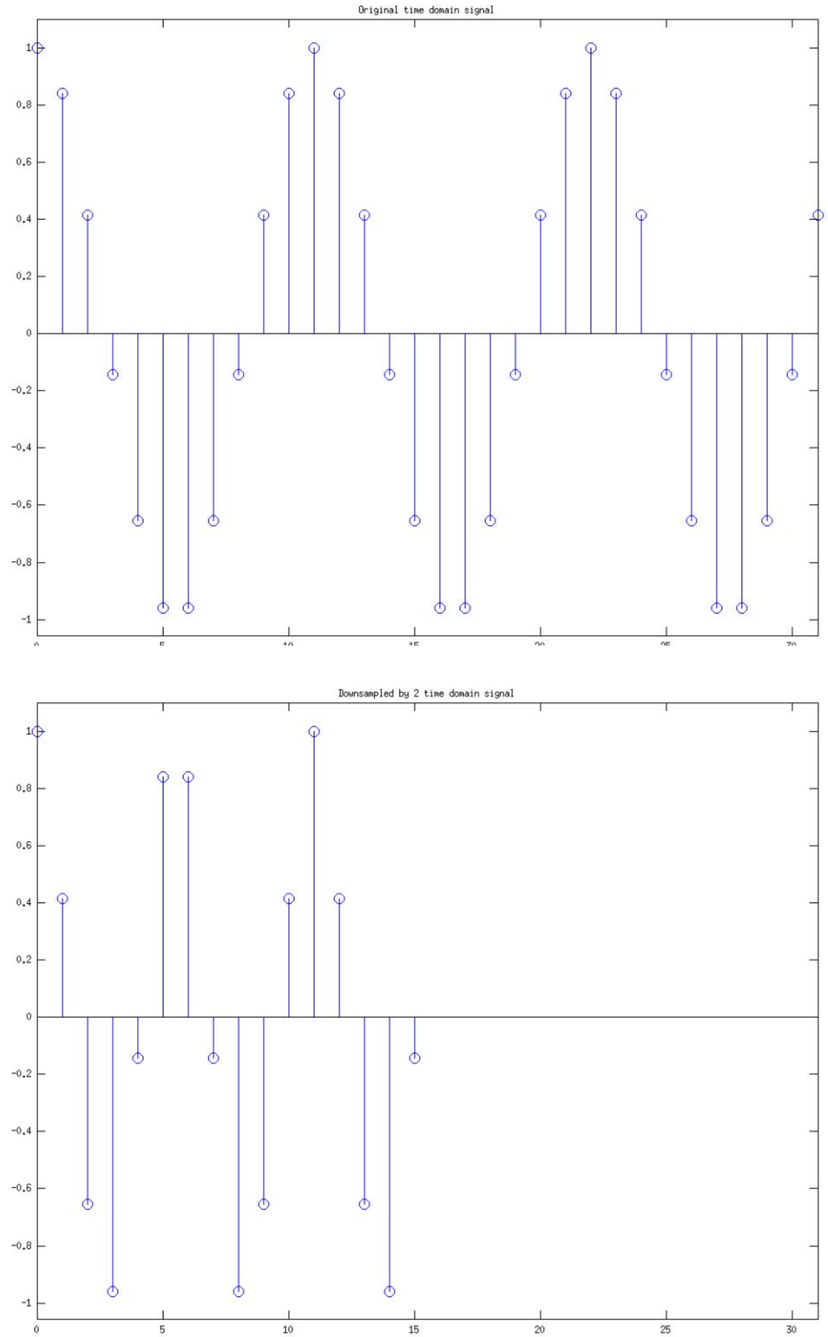
Downsampling or Decimation

- To decimate by a factor *D*, keep one of every *D* samples—on a periodic basis
- Downsampling by a factor *I* is commonly written $\downarrow$I
  For example, downsampling by two: $\downarrow$2
- Obviously the number of samples will be approximately cut in half after $\downarrow$2



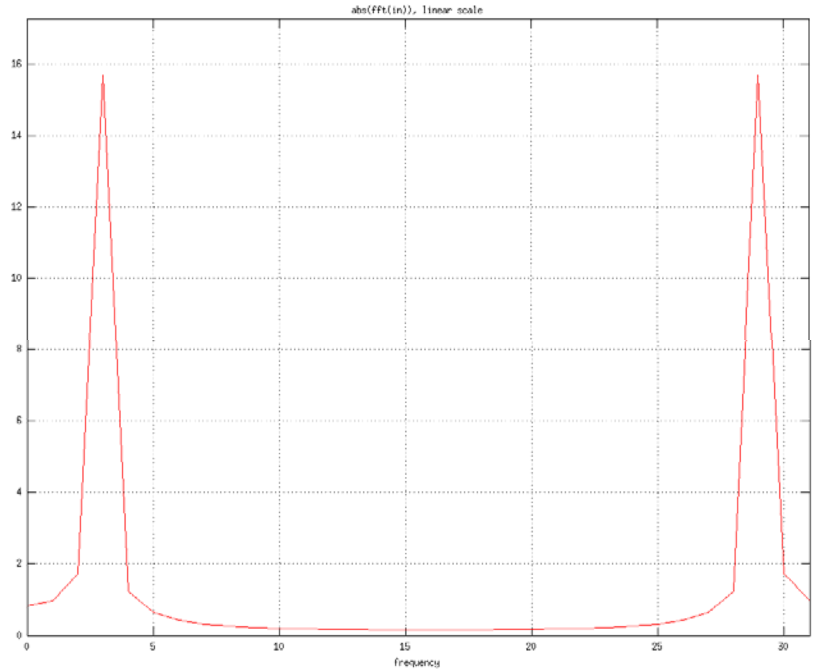
Decimation Example 1—Time Domain

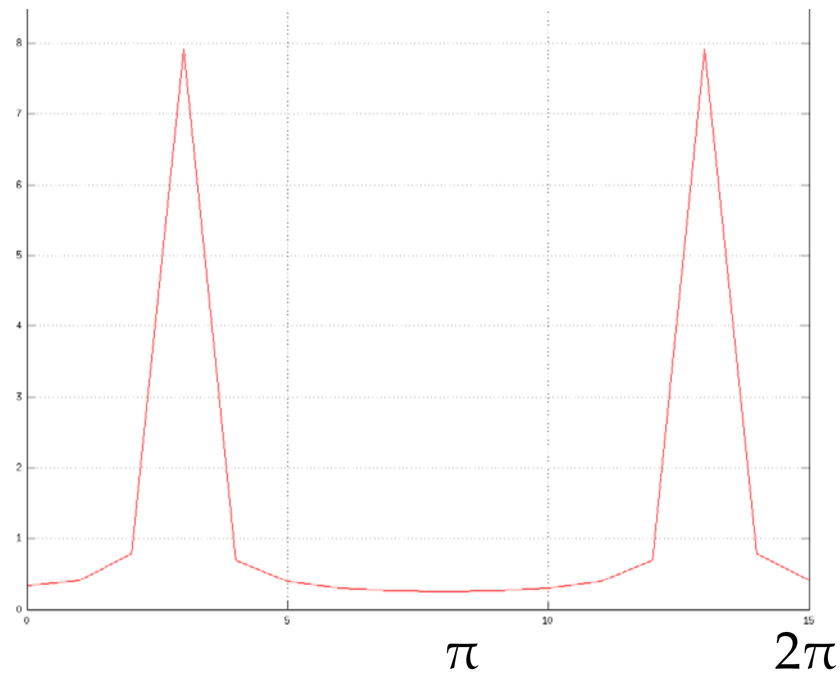
- 32 samples in original waveform
- Downsampled by 2
- 16 samples in downsampled waveform



Decimation Example 1—Frequency Domain

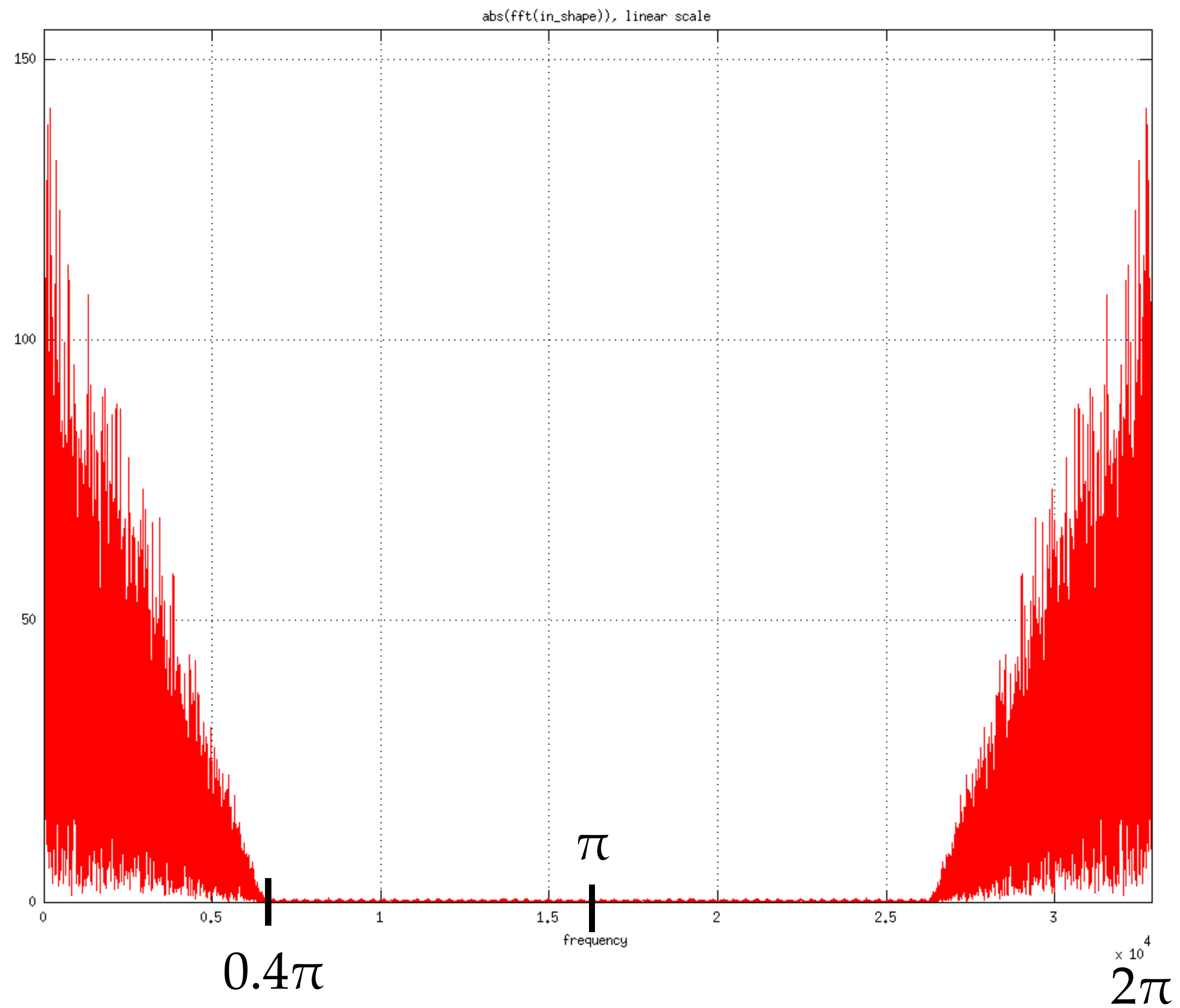
- Downsampled by 2
- Frequencies in downsampled waveform are 2x higher in the "digital frequency" domain
- They interfere when input frequencies reach $\pi/2$ (downsampled frequencies reach π)





Decimation Example 2
Spectrum, Linear Scale

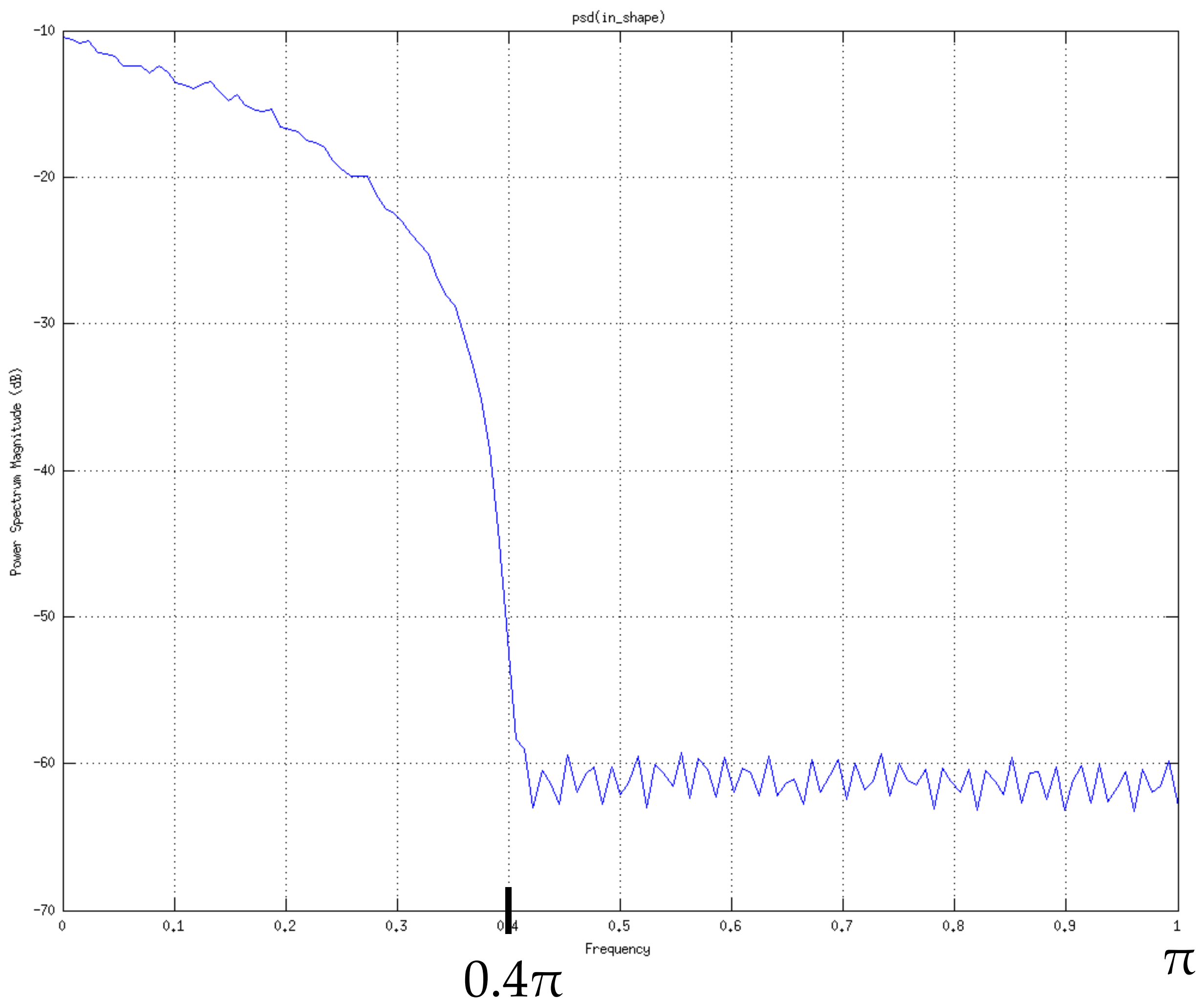
- Notice signal frequency slope from 0–0.4π
- Plotted with abs(fft(•))



downsamp2.m

Decimation Example 2
Spectrum, dB Scale

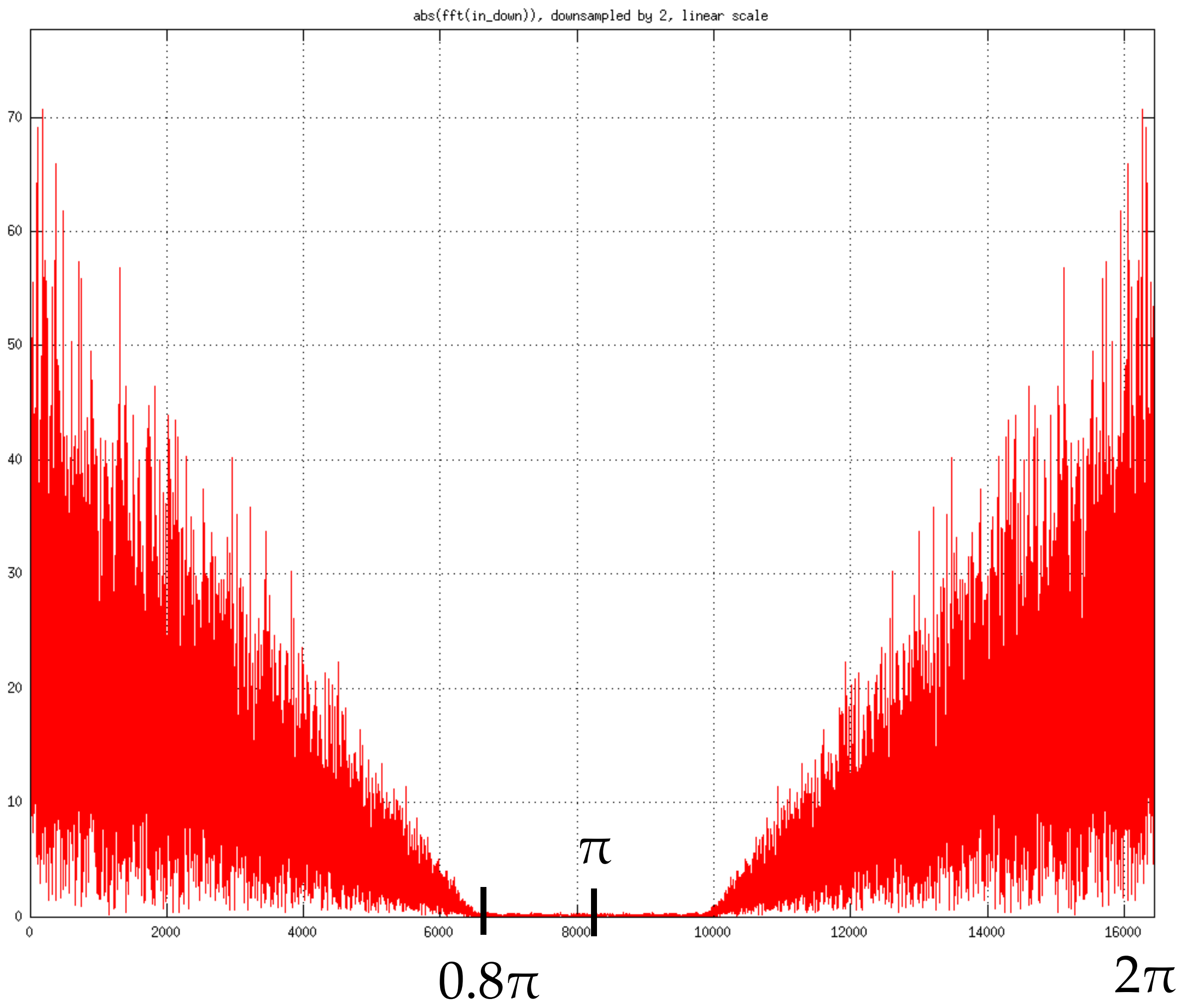
- Spectral slope from 0–0.4π not visible with dB vertical scale
- Notice signal goes to "zero" at 0.4π
- Plotted with psd(•)



downsamp2.m

Decimation Example 2

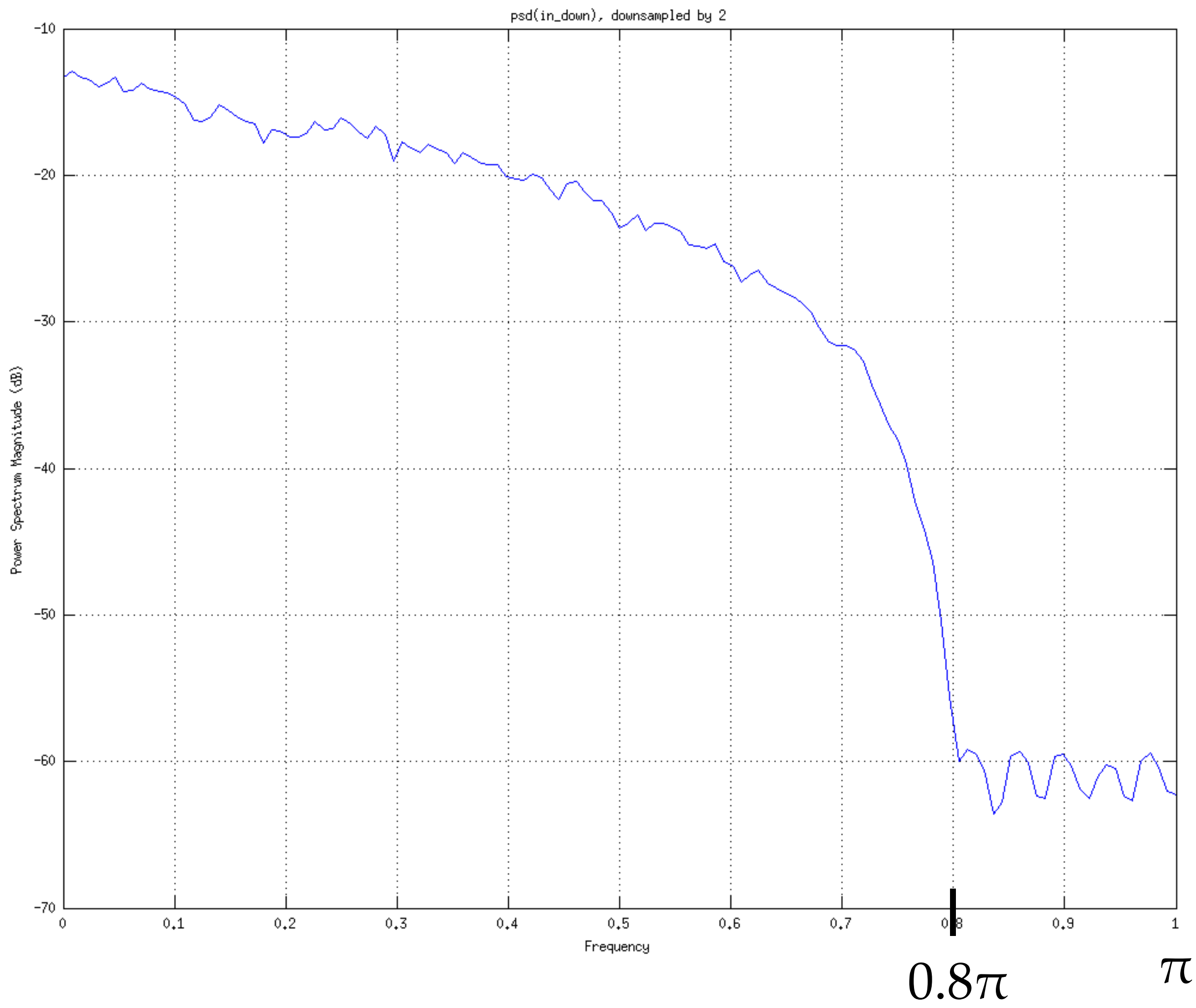
- Decimated by 2
- Spectral slope now from 0–0.8π
- Plotted with abs(fft(•))



downsamp2.m

Decimation Example 2

- Signal now from 0–0.8π though sloping shape is still not visible with dB vertical scale
- Plotted with psd(•)

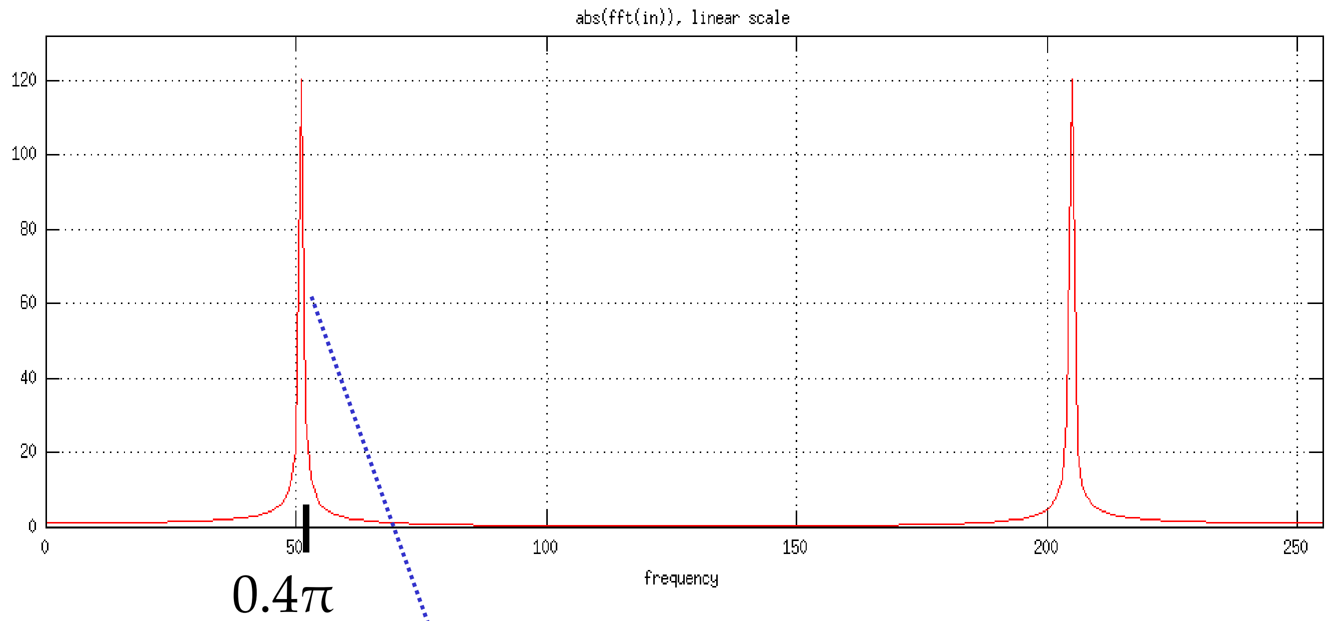


downsamp2.m

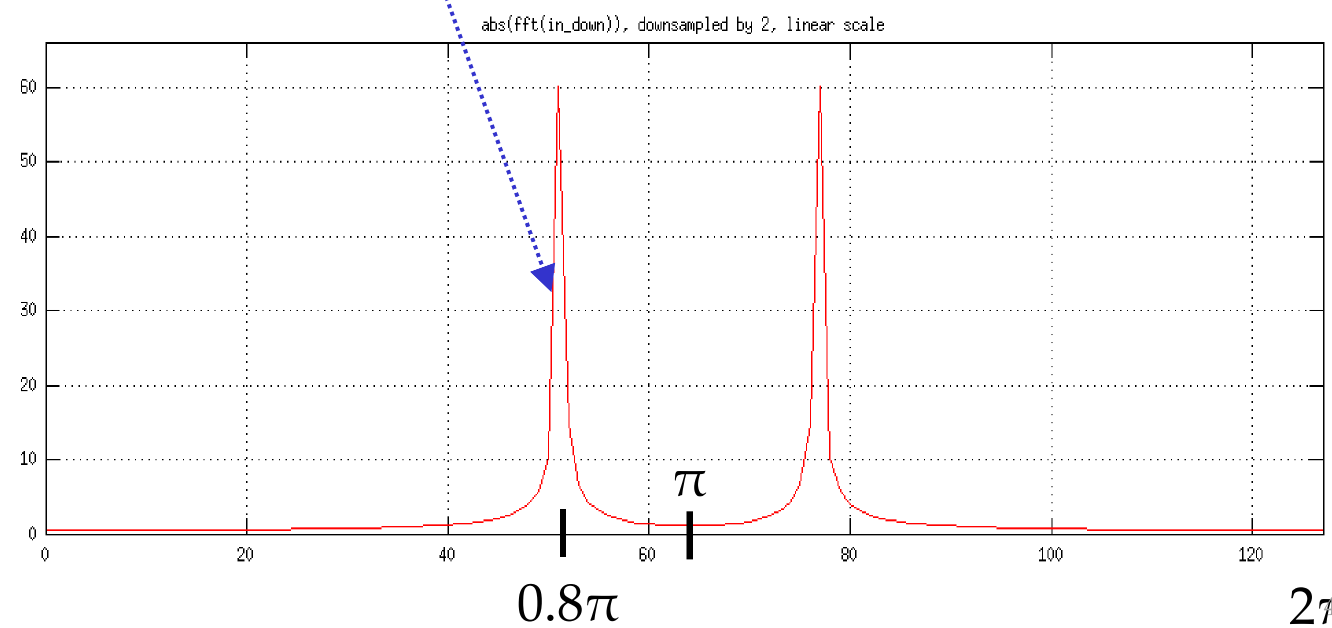
Decimation

- Signals above ½ the Nyquist frequency *alias* to lower frequencies when downsampled
  - Normally this is bad
  - In rare cases, this can be exploited to some benefit

Original

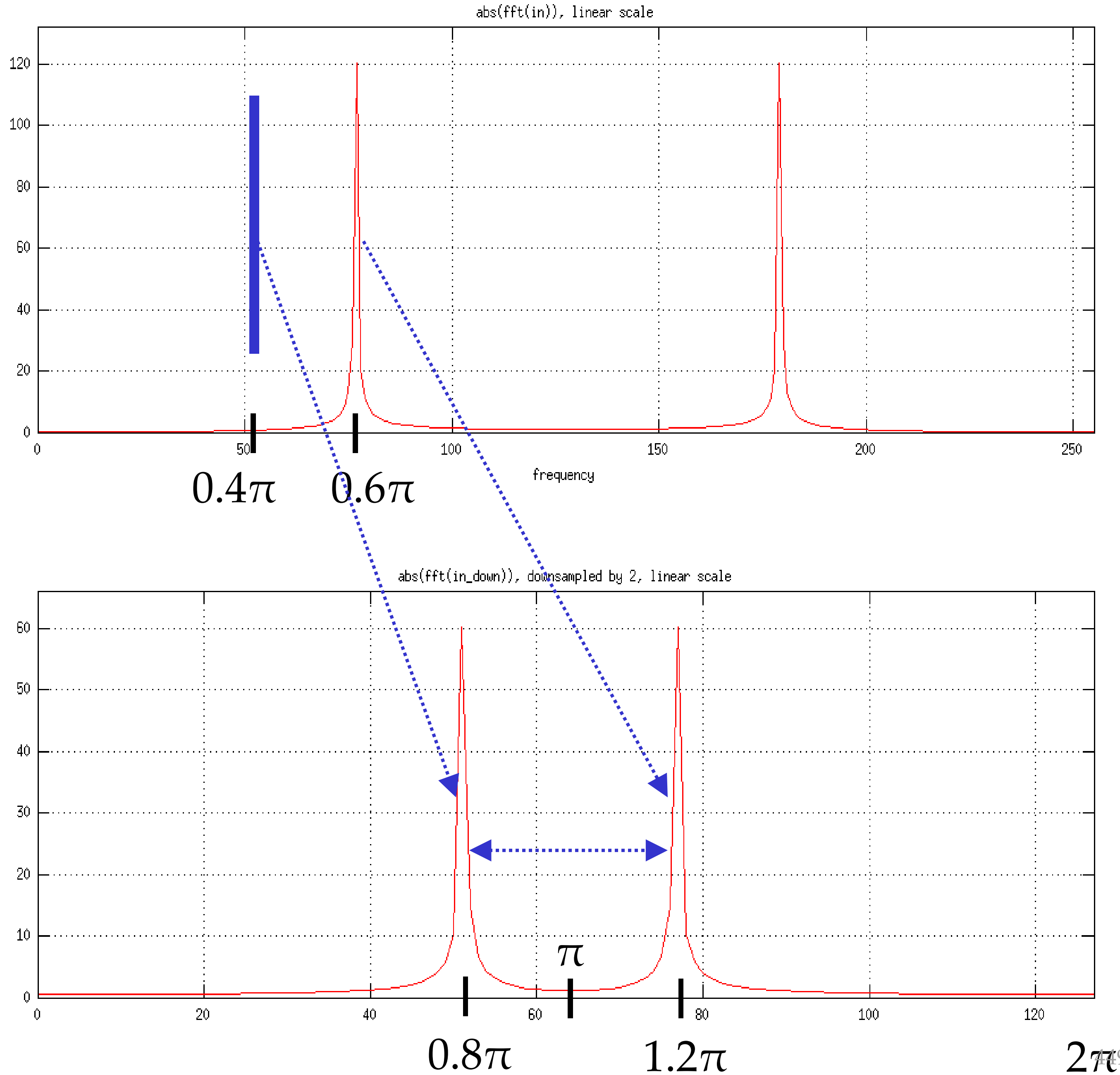


Down-
sampled
by 2



Original

Down-
sampled
by 2



Anti-alias filtering

- Normally we will want to filter the original signal with a filter before decimation to attenuate signal components which would alias into the desired signal band(s)
- This filter also has its own name: an *anti-alias filter*

Decimation Example

- Example
  - 1 MSample/sec signal, interested in 0–100 KHz portion only, decimate as much as possible
  - Need to watch "head room" above signal
  - Decimate by three → new f_s at 333 KHz
    - Leaves 67 KHz between highest freq components and $f_s/2$
  - Decimate by four → new f_s at 250 KHz
    - Leaves 25 KHz between highest freq components and $f_s/2$
  - Decimate by five → new f_s at 200 KHz
    - Leaves 0 KHz between highest freq components and $f_s/2$
    - Some aliasing will occur