spectral modeling

Michael Klingbeil
michael@klingbeil.com

CMC2005

why spectral models?

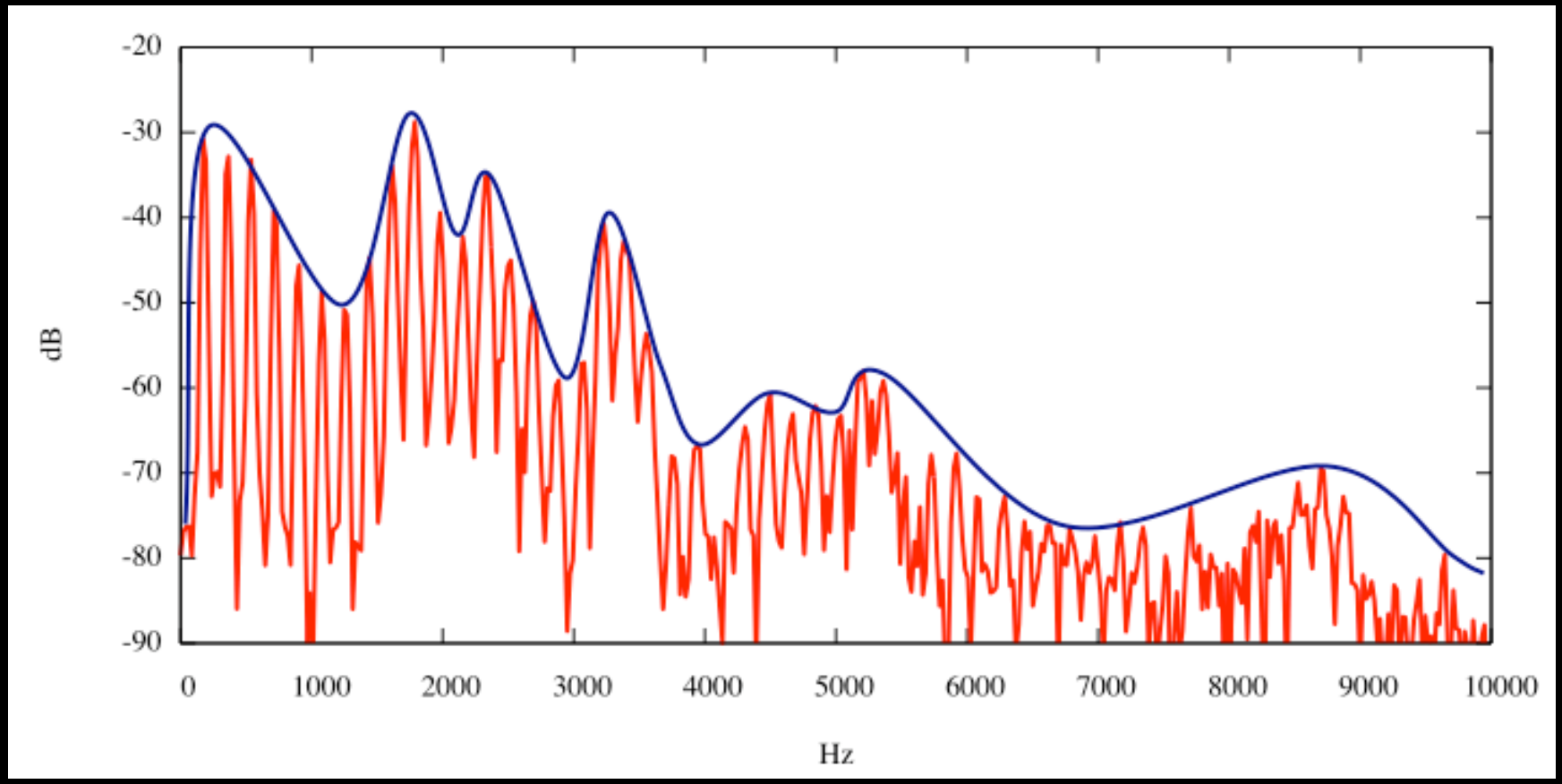
- manipulate and generate new sounds in terms of time varying *frequency* characteristics
- the human ear as spectrum analyzer
  - the cochlea of the inner ear separates sound into into (quasi) sinusoidal components
  - positions along the basilar membrane correspond to sinusoidal frequency
- sinusoidal motion is fundamental for objects that vibrate or resonate

spectral models: examples

- vocoder (spectral envelope)
- phase vocoder
- LPC – linear predictive coding (spectral envelope)
- sinusoidal modeling
- cepstrum (spectral envelope)
- wavelet transform
- matching pursuits

classes of spectral models

- spectral shape (envelope)
- spectral content (sinusoids)



sinusoidal modeling

- NB: we are making the assumption that sinusoids are a good way to model the signal
  - this may or may not be true!!

- sinusoid is specified in terms of its frequency *and* phase

$$x(t) = A\sin(2\pi ft + \phi)$$

A = amplitude

π = 3.14159… half the circumference of the unit circle

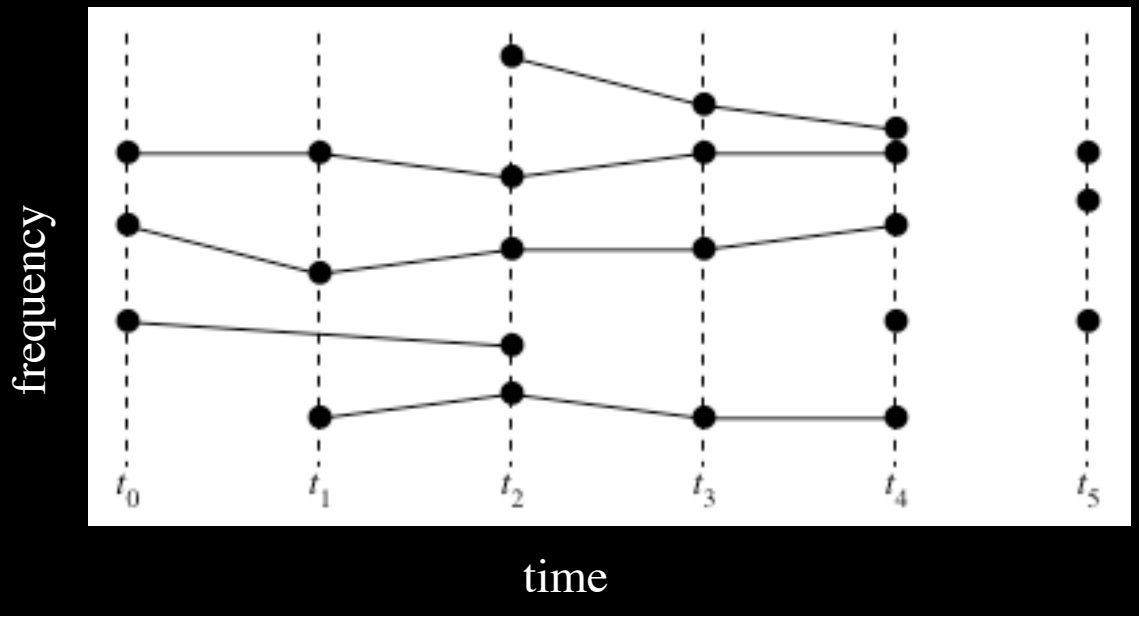
f = frequency in cycles per second (Hz)

ϕ = initial phase

- NB: information about sinusoidal frequency only (and not phase) is generally insufficient to reconstruct a signal precisely

sinusoidal modeling: overview

- use the Fourier transform to estimate the sinusoids (amplitude, frequency and phase) present at time t
  - averaged over some small number of samples, N
- advance time *t* by some amount h (the hop duration) and estimate the sinusoids present at time $t + h$
- try to connect the sinusoids at time t to the sinusoids at time $t + h$ into smooth frequency tracks



intuitive interpretation of the Fourier transform

- start with a waveform to be analyzed
- choose a set of reference sine and cosine waves at discrete frequencies
- “compare” the waveform to each reference sine and cosine wave by multiplying them point by point and adding up the values
- portions of the waveform that are like the reference sinusoid will result in larger sums
- the reference sinusoid will “resonate” with waveform components that are close in frequency and phase

sinusoidal modeling: some history

- Robert McAulay and Thomas Quatieri: Speech Analysis/Synthesis Based on a Sinusoidal Representation (1986)
- Julius Smith and Xavier Serra: PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation (1987)
- Xavier Serra: A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition (SMS) (1989)
  - SMSTools
- Robert Maher and James Beauchamp: SNDAN/MQAN (1988)
- Kelly Fitz, Lippold Haken, and Bryan Holloway: Lemur (1995)
- Kelly Fitz and Lippold Haken: The Reassigned Bandwidth-Enhanced Method of Additive Synthesis (1999)
  - Loris
- Juan Pampin: Analysis, Transformation, Synthesis (ATS) (1999)
- Sylvain Marchand and Robert Strandh: InSpect and ReSpect (1999)

SPEAR (2004)

- asynchronous partials (not locked to frames)
- realtime resynthesis
- open file formats
- cross platform
- nice looking GUI

SPEAR: text file formats

- **Text Resampled Frames**

```
par-text-frame-format
point-type index frequency amplitude
partials-count 38
frame-count 283
frame-data
0.000000 22 21 845.108887 0.005091 20 1091.096191 0.011639 . . .
0.010000 36 35 523.937317 0.012844 21 786.535583 0.027642 20
1066.157227 0.077848 . . .
0.020000 38 35 521.226868 0.028354 21 775.809753 0.051166 20
1062.829590 0.146942 . . .
. . .
```

SPEAR: text file formats

- **Text Partials**

```
par-text-partials-format
point-type time frequency amplitude
partials-count 38
partials-data
0 24 0.000000 0.127778
0.000000 13225.231445 0.000048 0.005556 13221.041016 0.000094 0.011111
13221.066406 0.000118 . . .
1 36 0.000000 0.194444
0.000000 12151.894531 0.000113 0.005556 12139.592773 0.000225 0.011111
12136.164062 0.000288 . . .
2 60 0.000000 0.333333
0.000000 11688.482422 0.000162 0.005556 11674.118164 0.000329 0.011111
11669.075195 0.000446 . . .
. . .
```

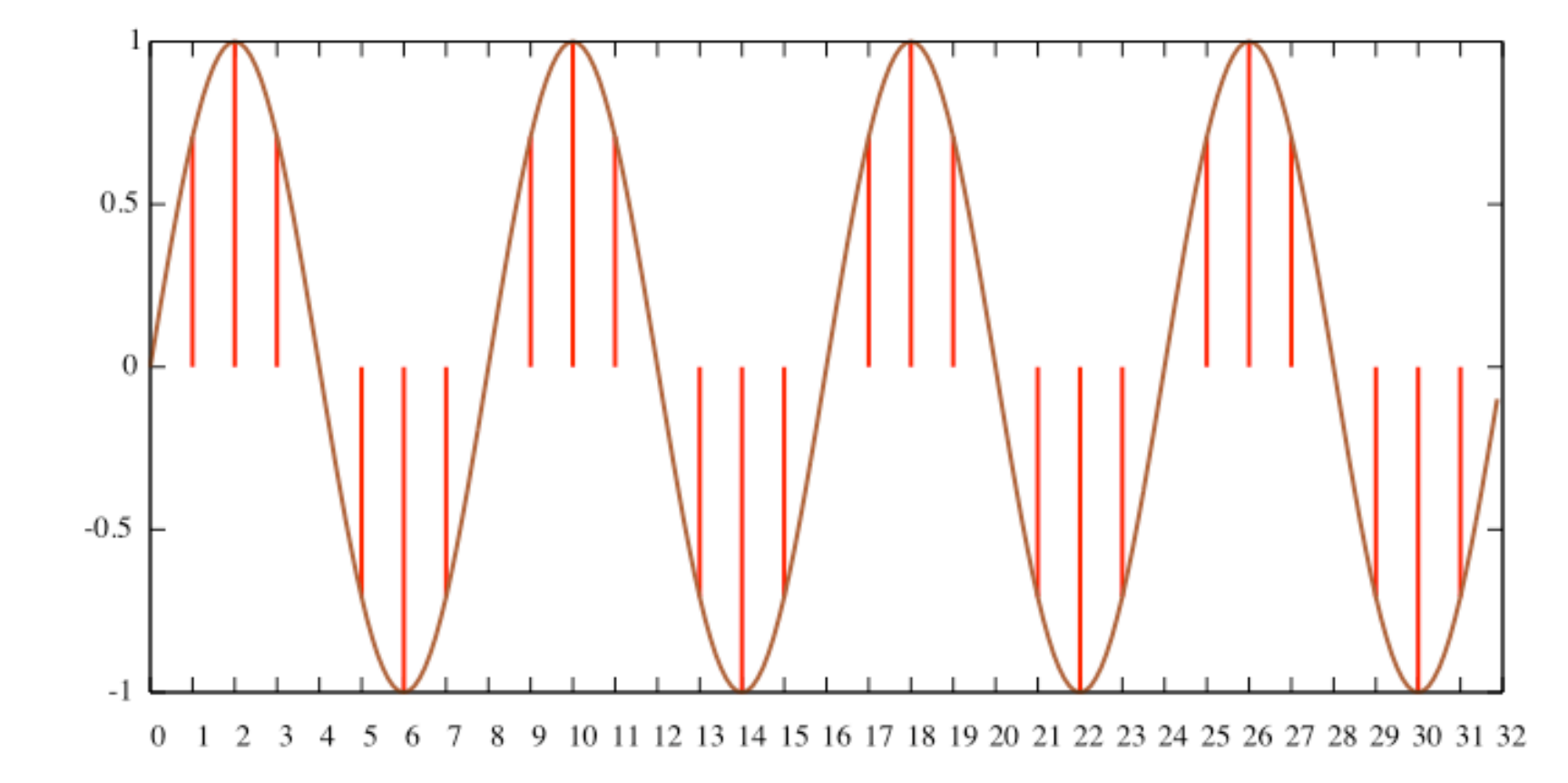
the partial tracking problem

- how to connect sinusoids from one frame to the next
- how to determine whether the set of connections over n frames (not *just* from frame i to $i+1$) is in some sense optimal
- how to compute the optimal connection efficiently

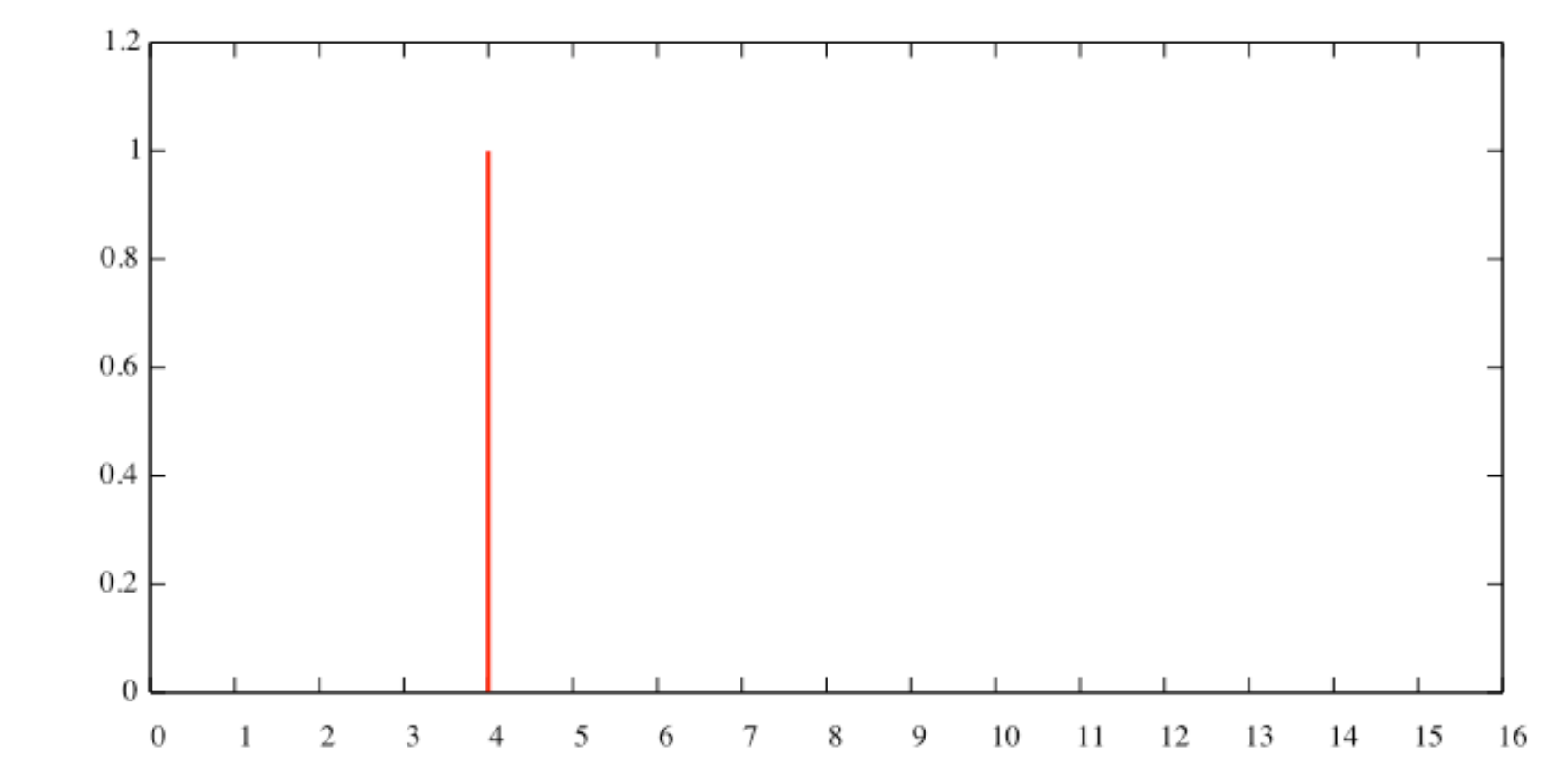
the time frequency resolution problem

- high time resolution => low frequency resolution
  - “grainy” artifacts
- high frequency resolution => low time resolution
  - time smearing

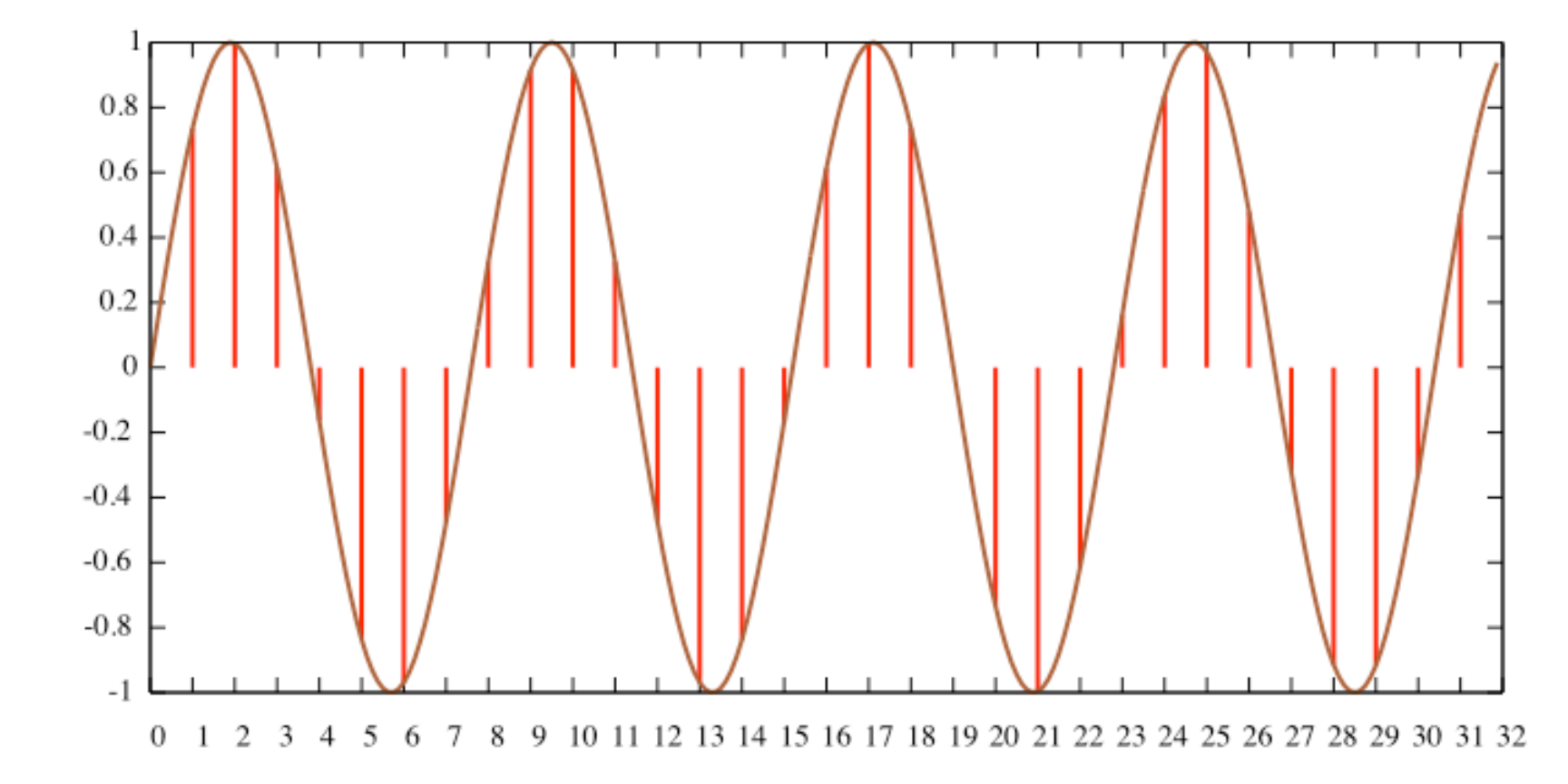
- the discrete Fourier transform estimates the sinusoids present over some short time interval (N discrete samples)
- consider sampled input signal $x(n)$, $N = 32$



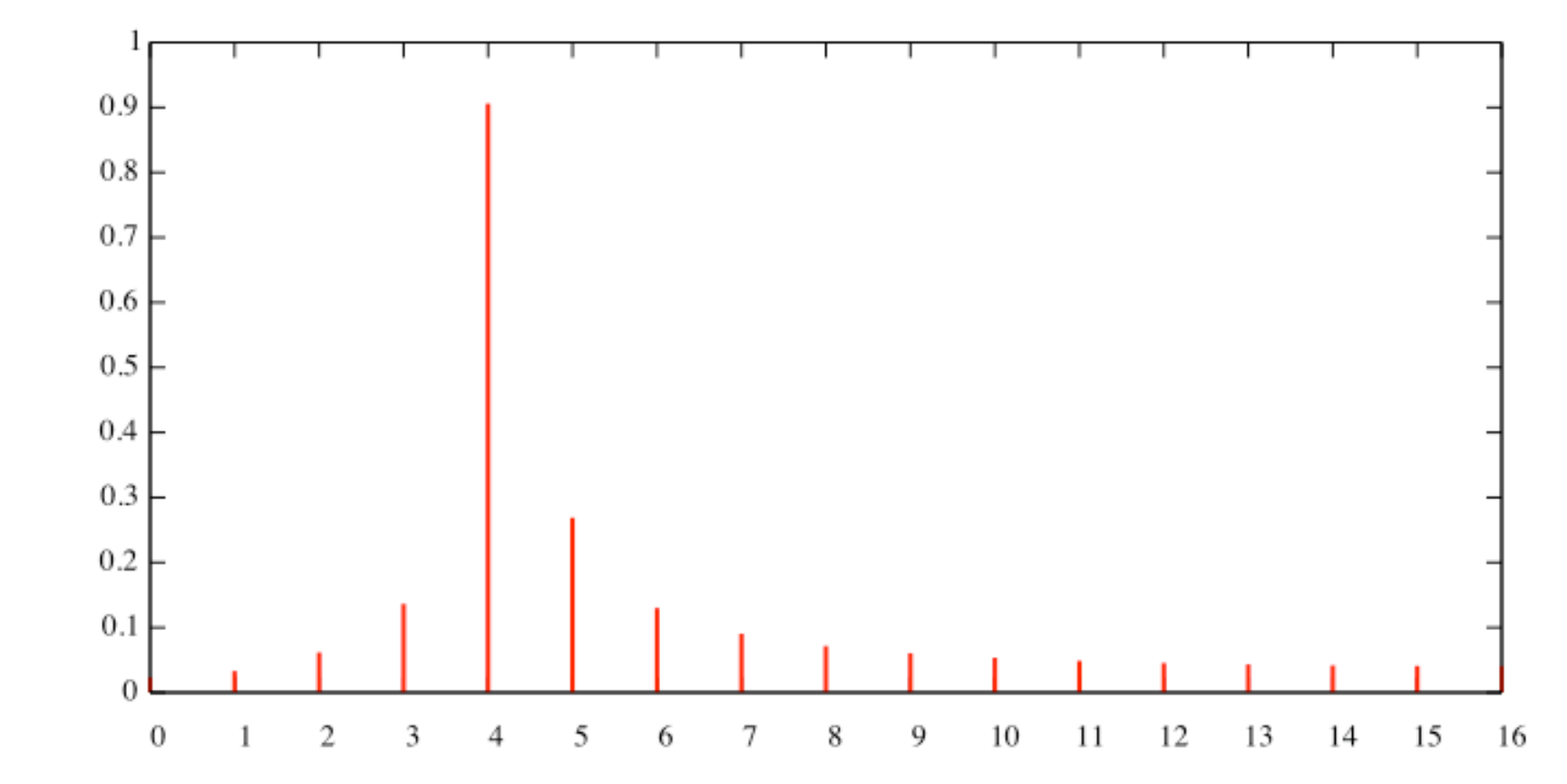
- **compute the DFT of $x(n)$, $N = 32$**
- **here is the magnitude spectrum:**



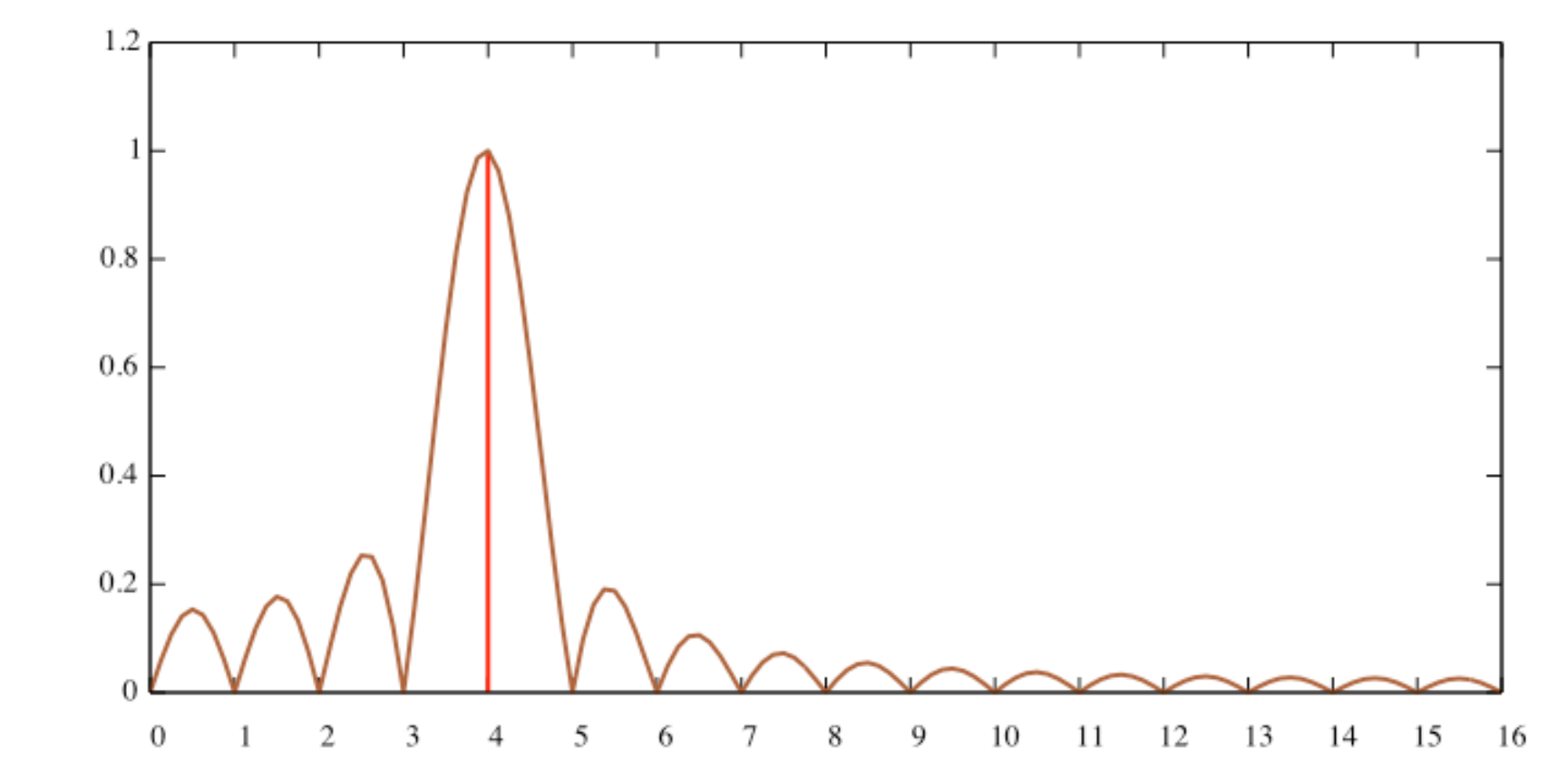
- now consider input signal $x(n)$
- the frequency is 4.21 cycles per N samples:



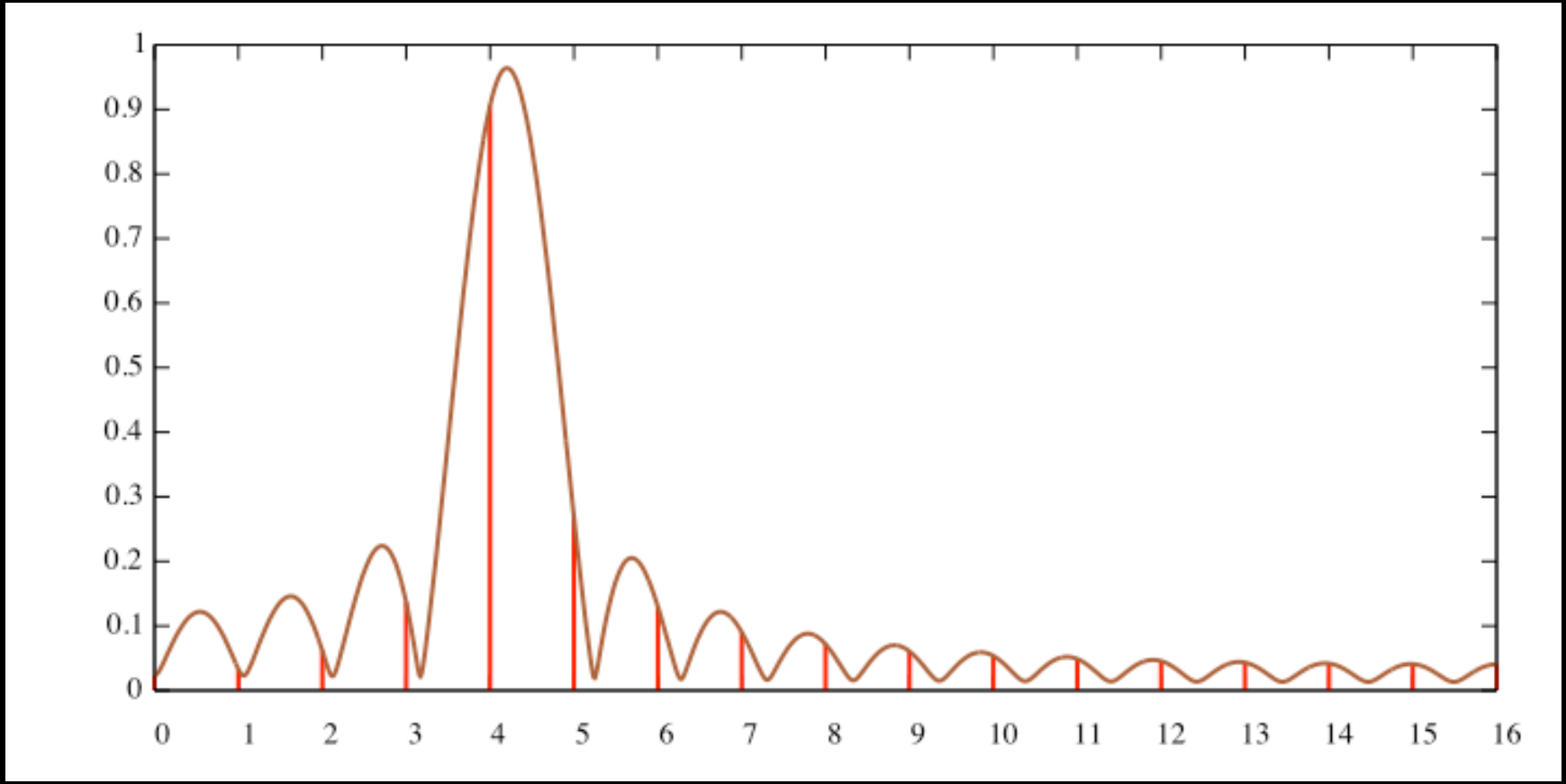
- compute the DFT of $x(n)$, $N = 32$
- here is the magnitude spectrum:
- what happened?



- just as $x(n)$ is a discrete sampling of a continuous signal…
- the DFT gives a sampling of a continuous spectrum
- when the frequency is an integer multiple of the DFT size, the spectral samples align with zeros in the continuous spectrum:



- the underlying spectrum becomes evident when the frequency is no longer an integer multiple of the DFT size
- the continuous spectrum is shifted with its peak aligned at the sinusoid frequency (in this case 4.21 cycles)
- the samples remain fixed at integer values:



intuitive explanation

- why are there “ripples” in the continuous spectrum?

- recall: the spectrum is defined in the frequency domain only
- the time variable of the Fourier transform extends from $-\infty$ to $+\infty$

- however, we are looking at the spectrum of a signal which only lasts for N samples
- its value is zero everywhere else

- these extra frequency components are the frequency domain manifestation of the signal “turning off” after a duration of N samples

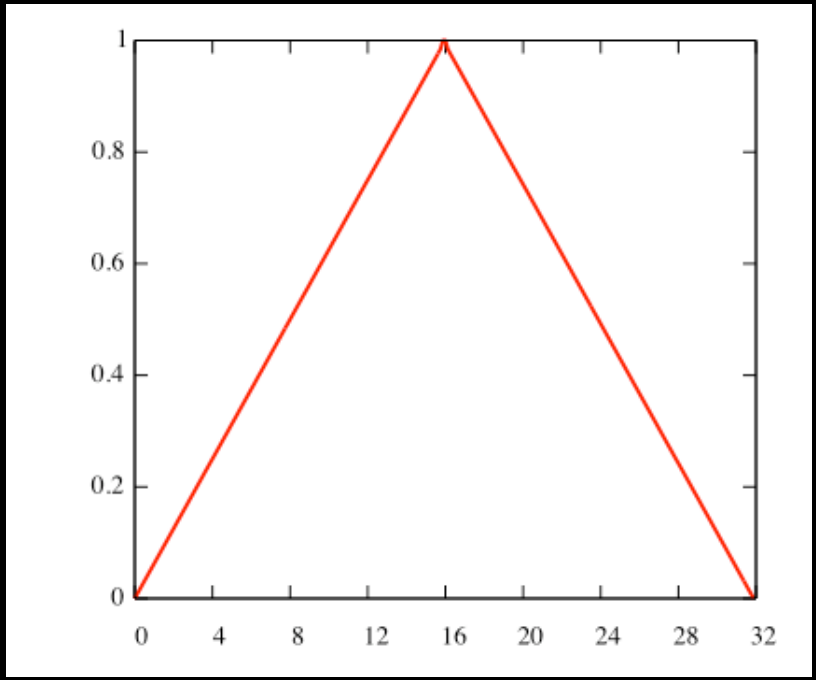
definition of the Fourier transform

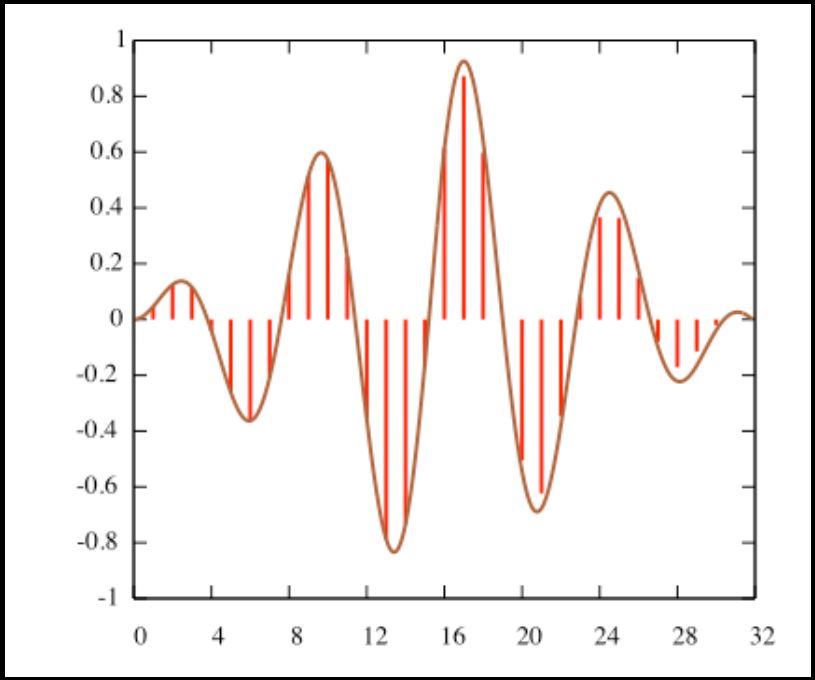
- forward transform

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi i f t} dt$$

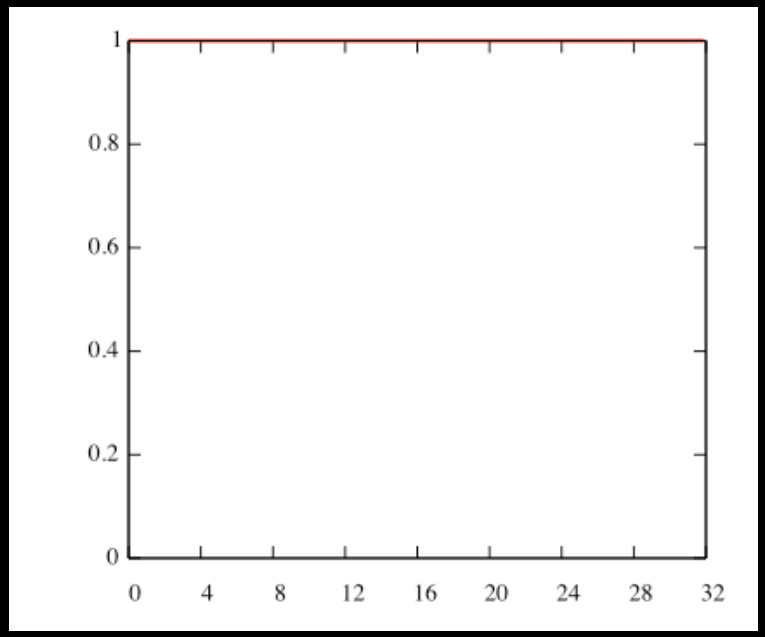
windowing — a solution

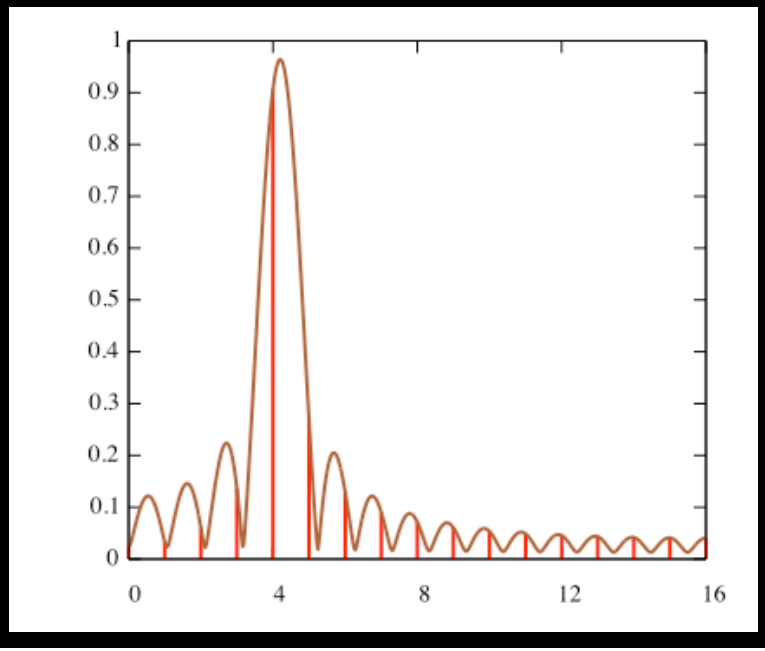
- reduce the frequency domain effects of the discontinuity at the boundaries of the analysis signal segment
- multiply the time domain signal by a smoothing window



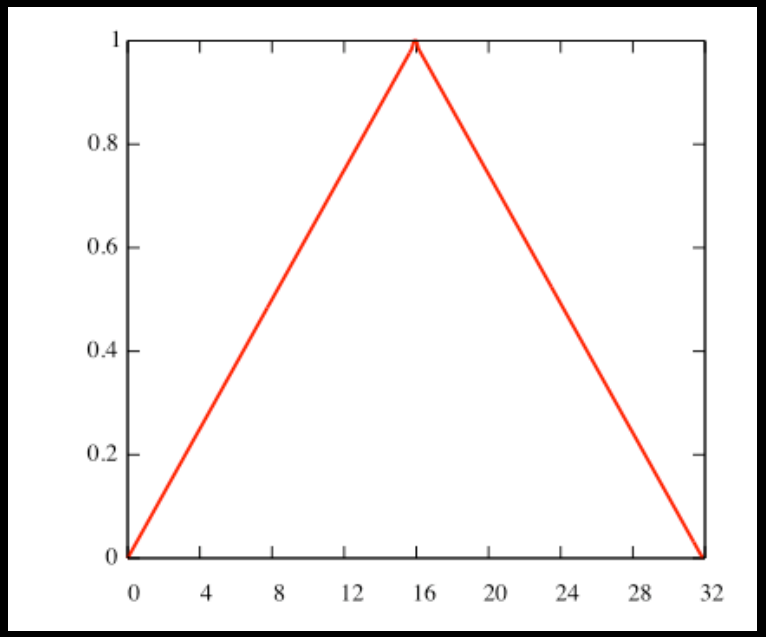


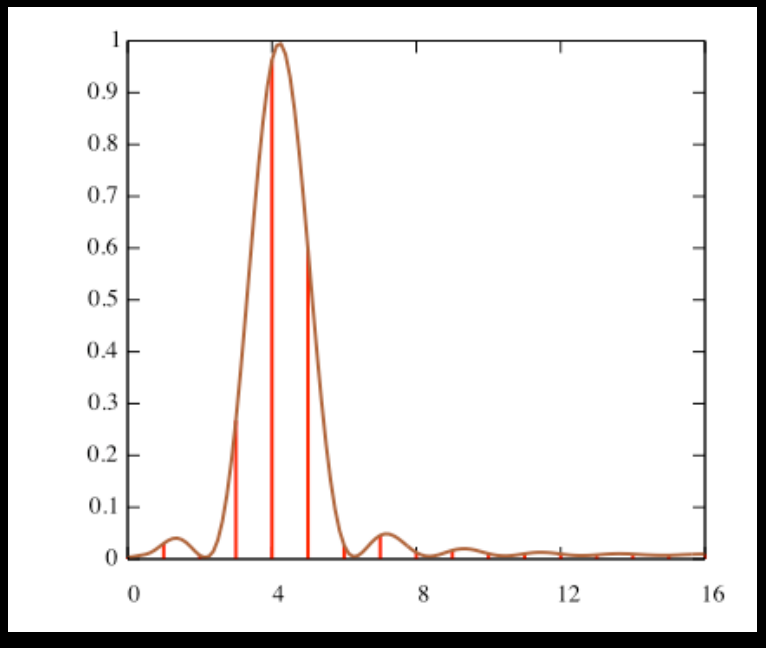
rectangular window



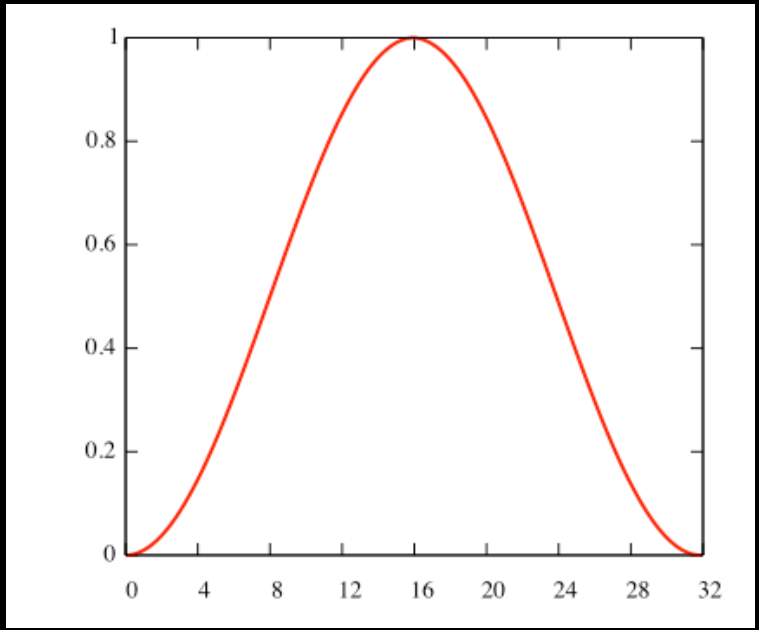


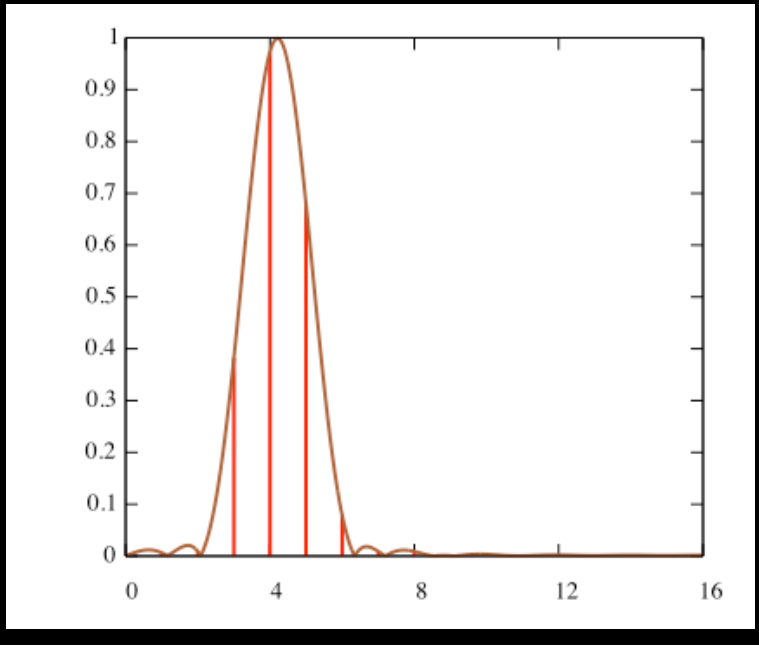
triangular window



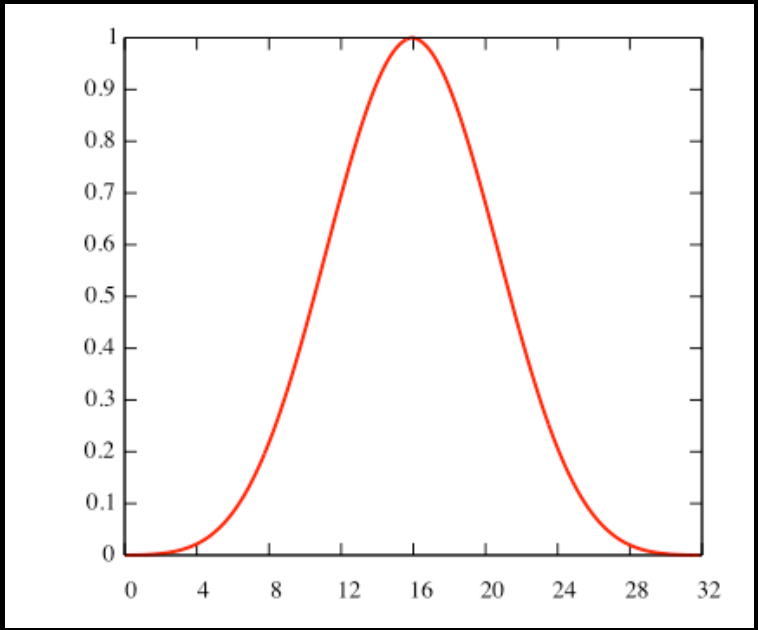


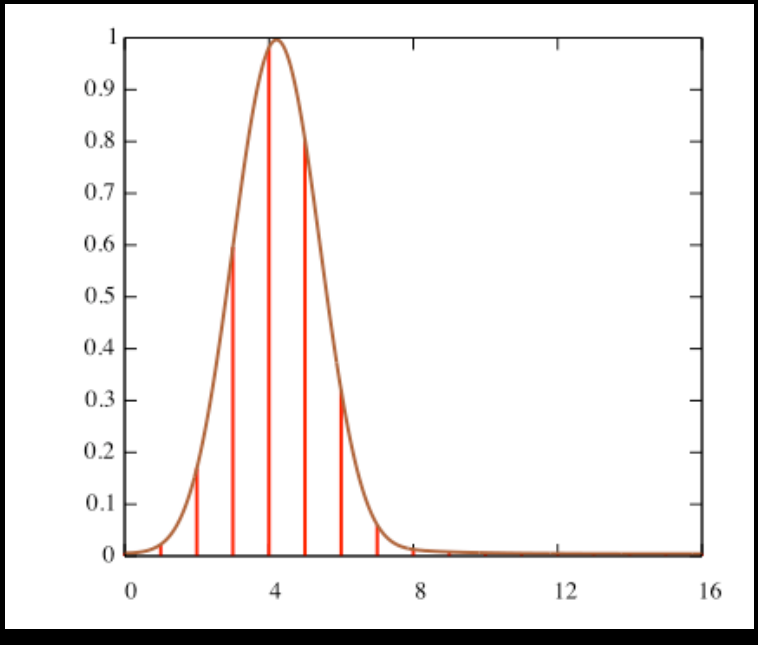
Hann window





Blackman 3 term window



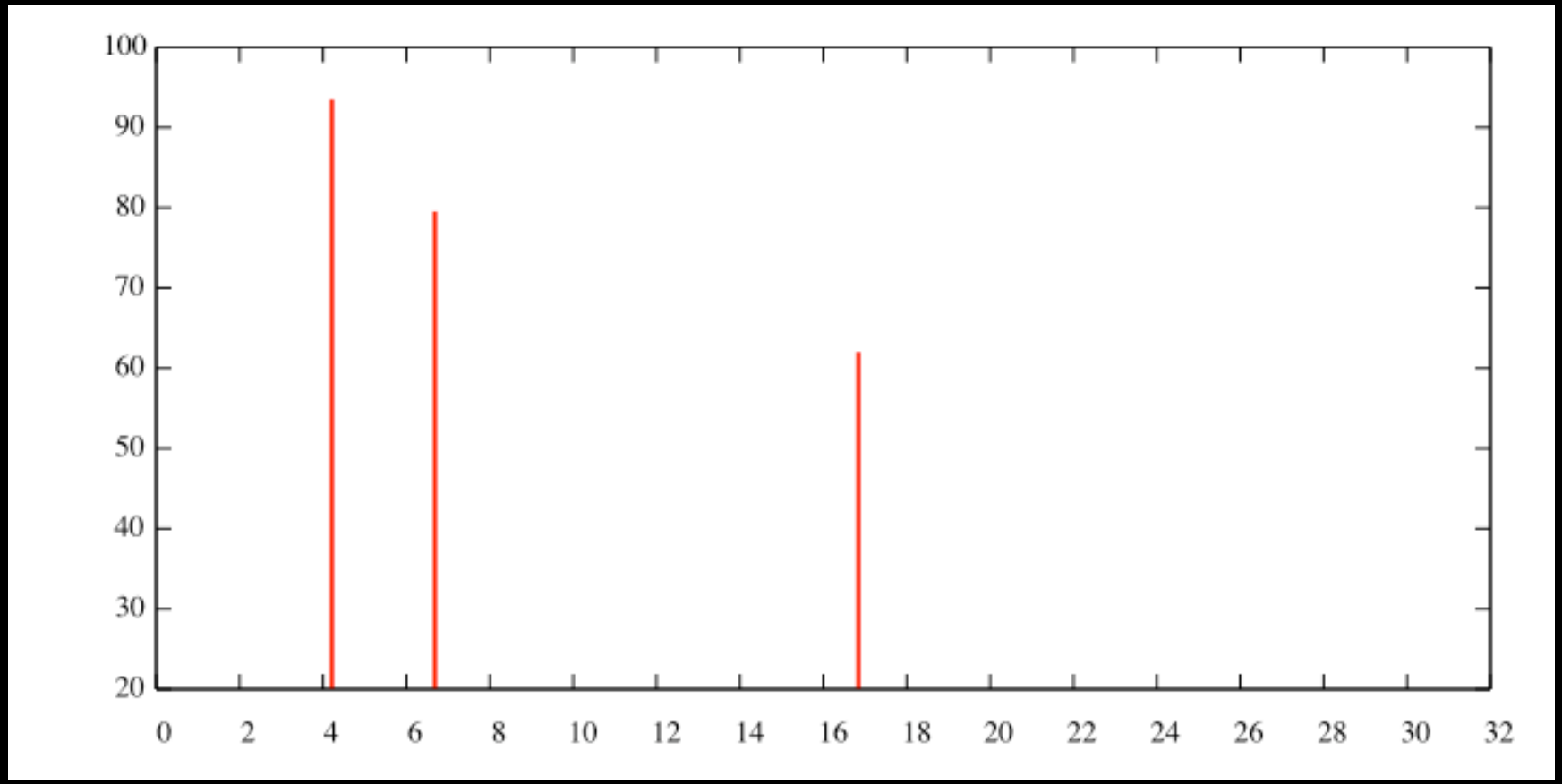


windowing tradeoffs

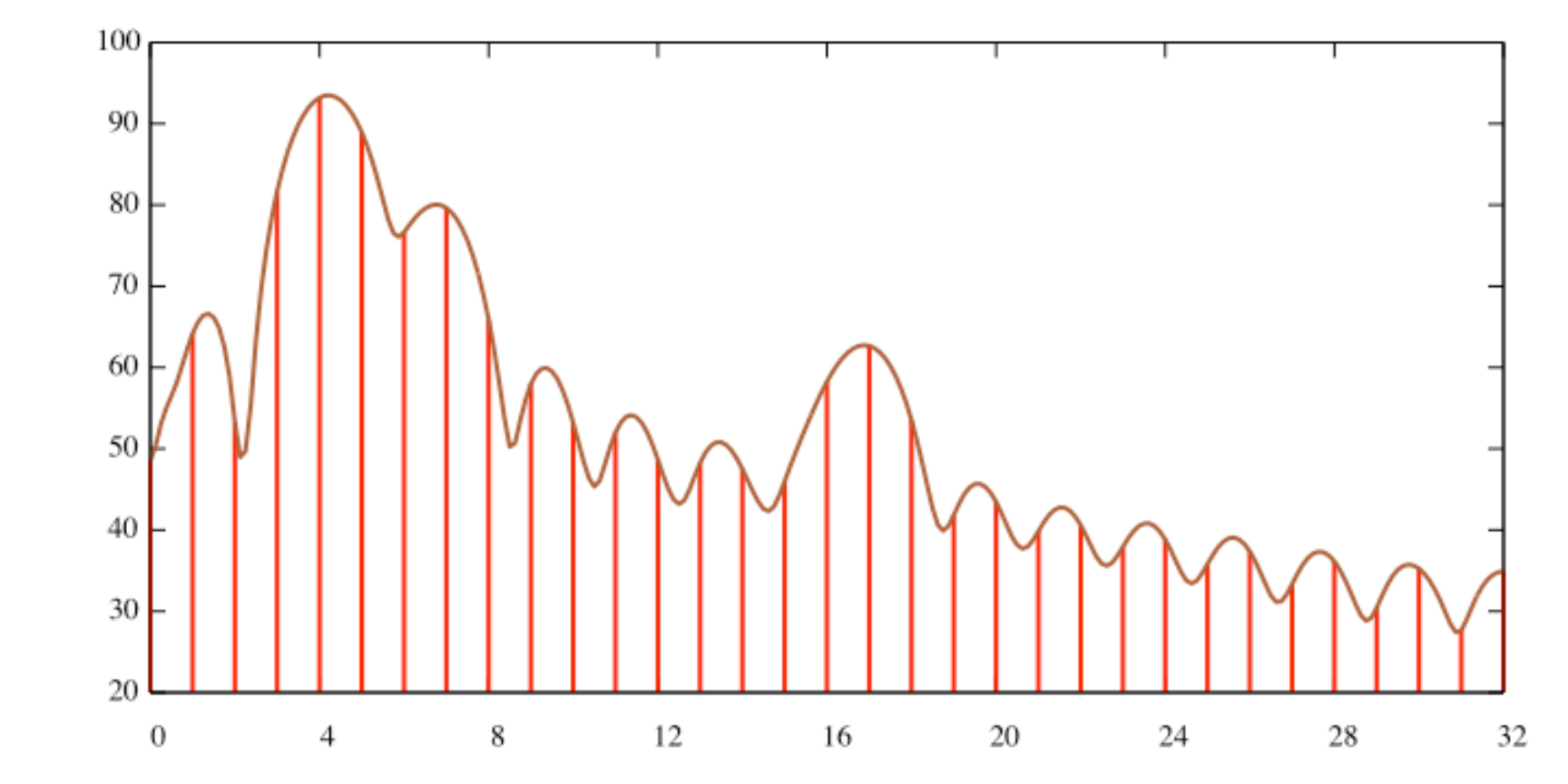
- narrow peak in the time domain
  - low side lobe levels
  - wide spectrum main lobe
  - poor frequency resolution
- wide peak in the time domain
  - high side lobe levels
  - narrow spectrum main lobe
  - improved frequency resolution

windowing tradeoffs for complex signals

- consider an input signal x(n) made up of a sum of sinusoids
  - 4.21 cycles at amplitude 0.75 (-2.5dB)
  - 6.68 cycles at amplitude 0.15 (-16.5 dB)
  - 16.84 cycles at amplitude 0.02 (-33.9 dB)

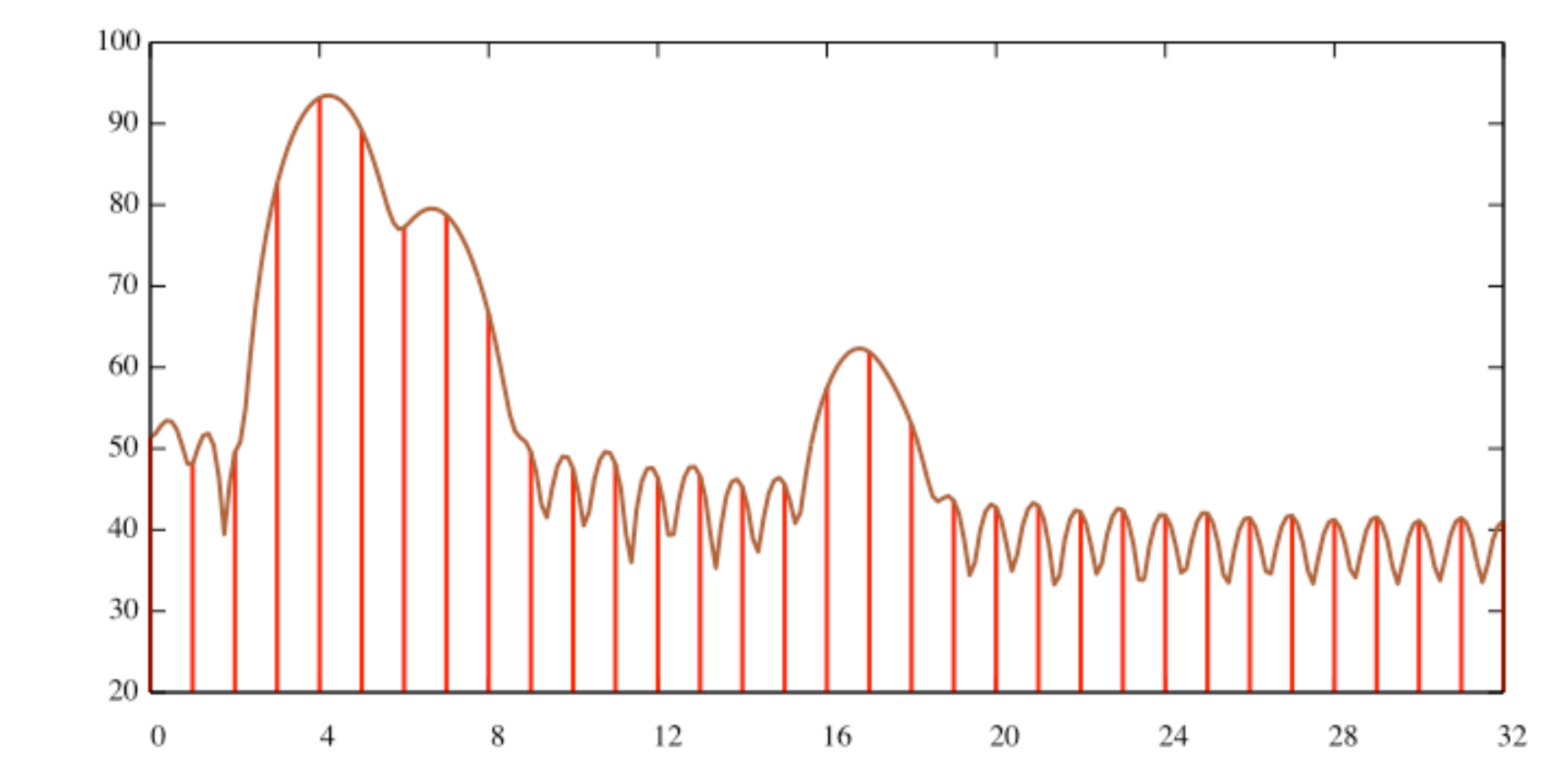


triangular window



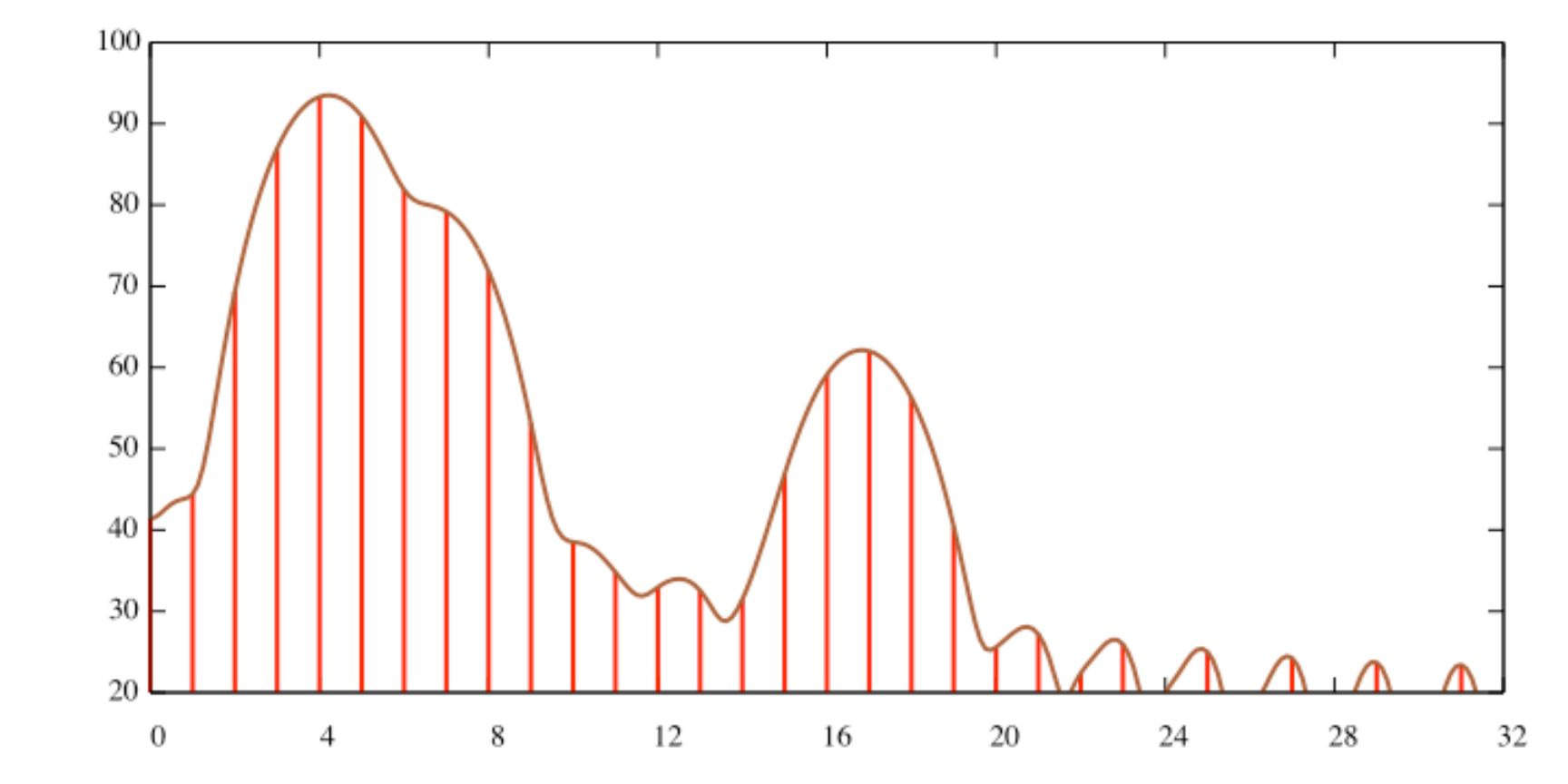
- cannot distinguish sinusoidal components from side lobe ripple

Hamming window



- side lobe levels considerably reduced
- individual sinusoidal components are resolved

Blackman window (2 term)



- reduced frequency resolution due to wide main lobe
- individual sinusoidal components are *not* all resolved

choose windows carefully

- short duration window
  - good time resolution
  - poor frequency resolution
- longer duration window
  - better frequency resolution
  - can resolve low frequencies and more closely spaced sinusoids
  - poor time resolution
- “smoother” window edges
  - lower side lobe levels
  - less chance of spurious sinusoids
  - decreased frequency resolution
  - can be compensated for by increasing window length
  - but this reduces time resolution!

we need better methods for spectral modeling

- time/frequency reassignment
- transient detection
- noise modeling
- multiresolution analysis
- wavelet transform
- high resolution matching pursuits
- what else?