Lecture 8: Interpolation

Mark Hasegawa-Johnson

ECE 401: Signal and Image Analysis

Outline

How to sample a continuous-time signal

Suppose you have some continuous-time signal, $x(t)$, and you'd like to sample it, in order to store the sample values in a computer. The samples are collected once every $T_s = \frac{1}{F_s}$ seconds:

$$x[n] = x(t = nT_s)$$

Outline

How can we get $x(t)$ back again?

We've already seen one method of getting $x(t)$ back again: we can find all of the cosine components, and re-create the corresponding cosines in continuous time.

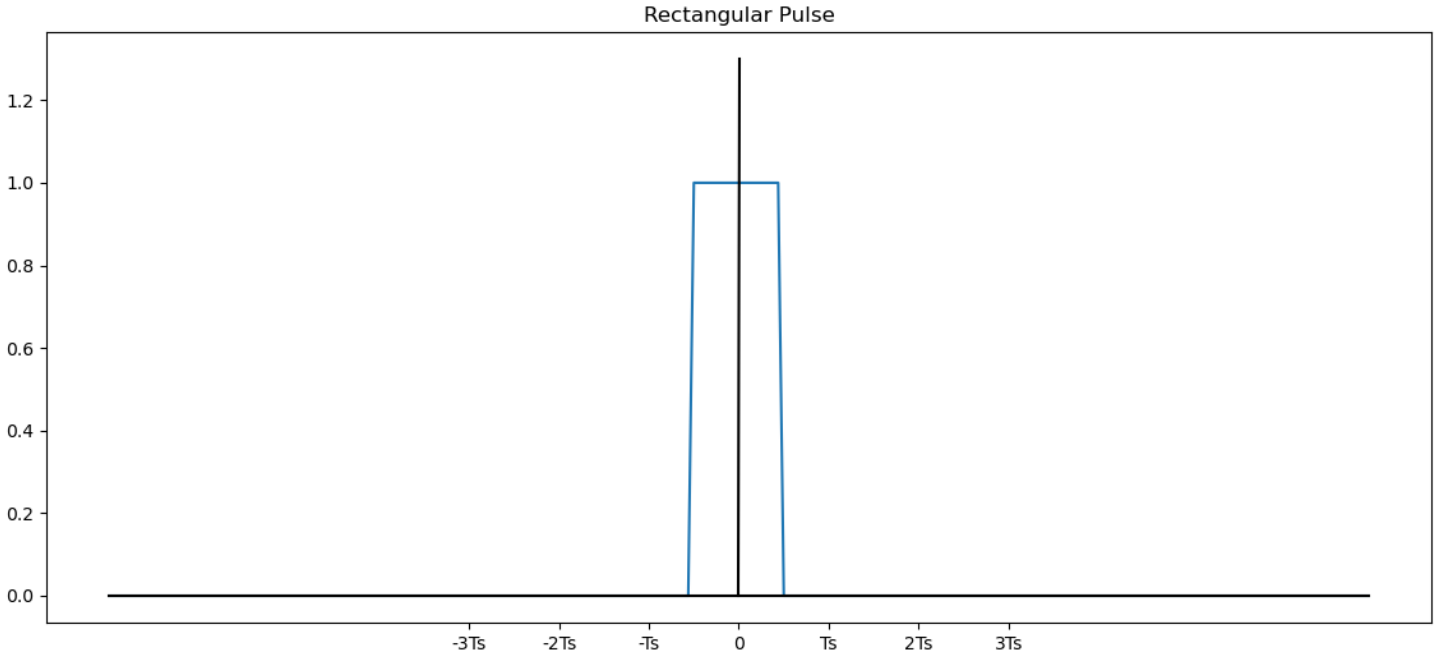
There is a more general method, that we can use for any signal, even signals that are not composed of pure tones. It involves multiplying each of the samples, $x[n]$, by a short-time pulse, $p(t)$, as follows:

$$y(t) = \sum_{n=-\infty}^{\infty} y[n]p(t - nT_s)$$

Rectangular pulses

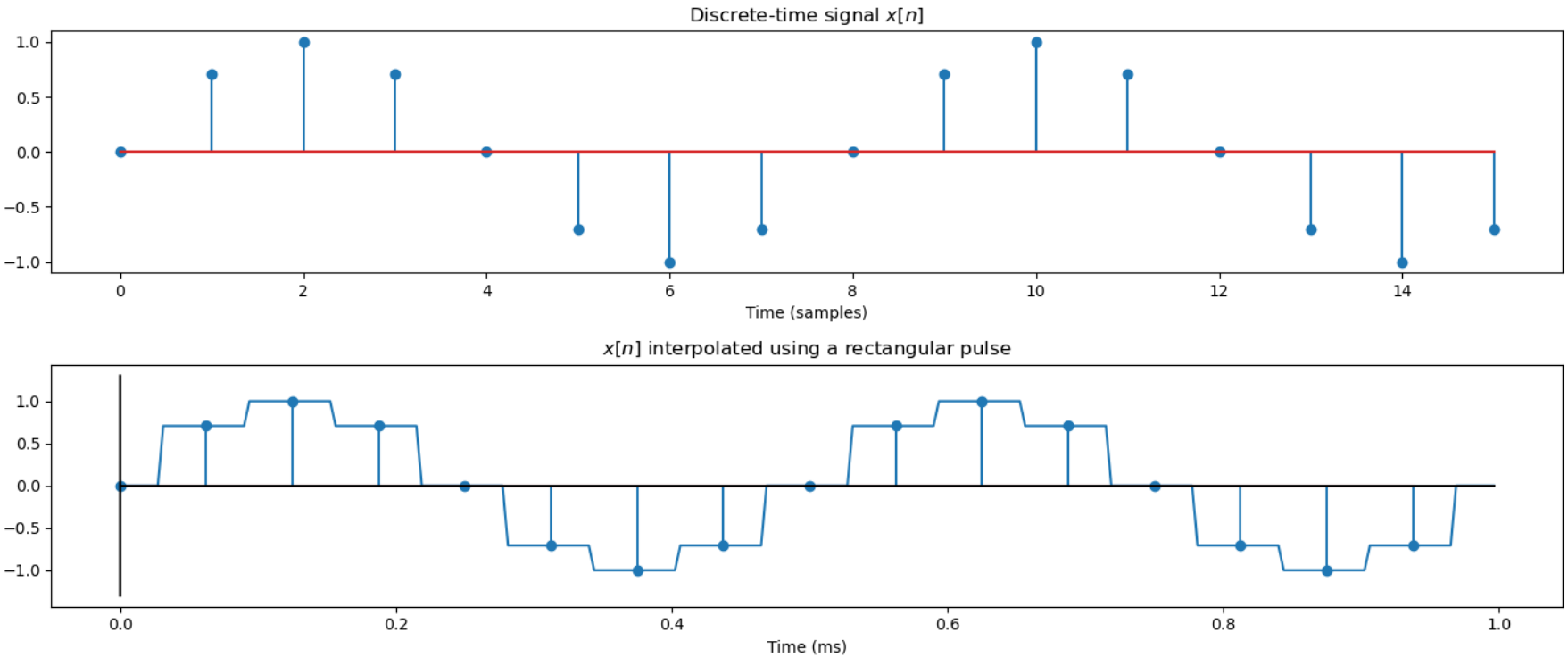
For example, suppose that the pulse is just a rectangle,

$$p(t) = \begin{cases} 1 & -\frac{T_S}{2} \leq t < \frac{T_S}{2} \\ 0 & \text{otherwise} \end{cases}$$



Rectangular pulses = Piece-wise constant interpolation

The result is a piece-wise constant interpolation of the digital signal:



Although this has roughly the right values, it is discontinuous. The discontinuities make an image "blocky," and add high-frequency noise to an audio signal.

Aliasing caused by imperfect interpolation

For example, suppose we start with $x[n] = \cos\left(\frac{n\pi}{8}\right)$, and interpolate using a rectangular pulse with $T_s = \frac{1}{8000}$ of a second. We wind up with

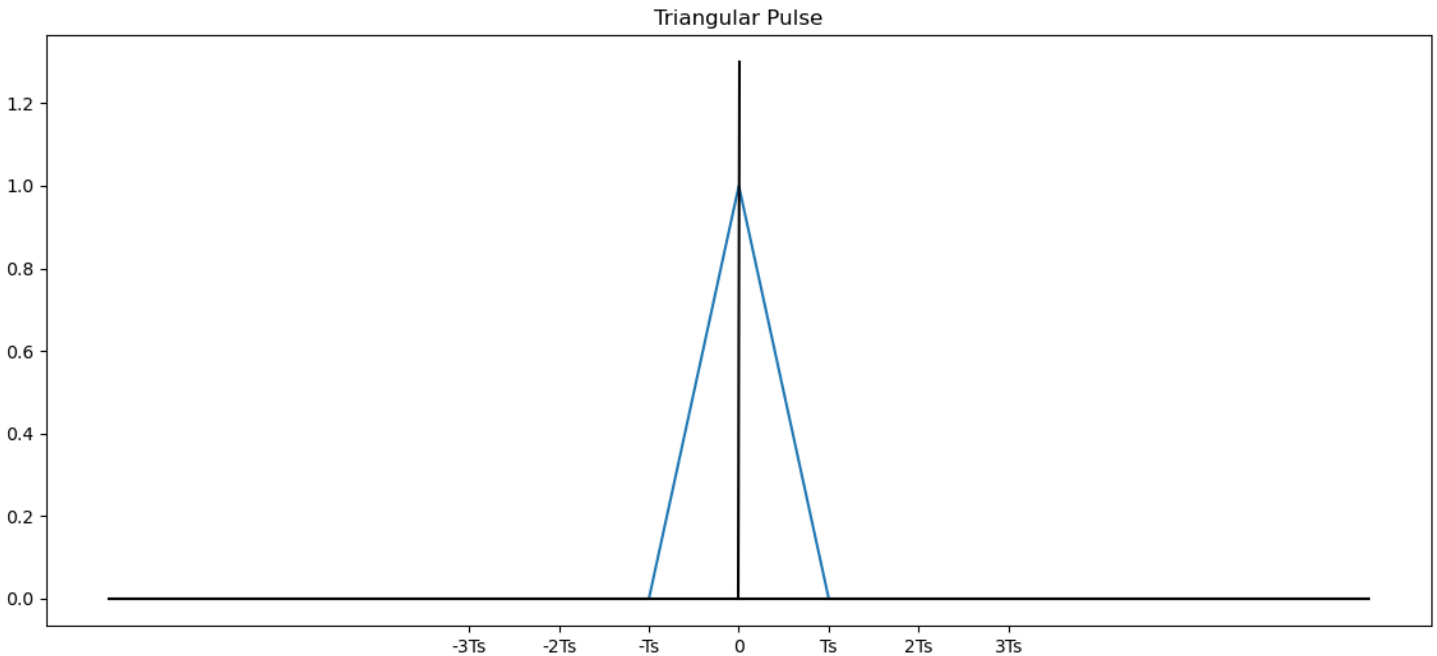
$$y(t) = a\cos(2\pi 1000n) + b\cos(2\pi 7000n) + c\cos(2\pi 9000n) + \cdots,$$

where $a \approx 1$, and $b \approx 0$ and $c \approx 0$, but not exactly. Since b and c are not zero, we can hear tones at those aliased frequencies.

Triangular pulses

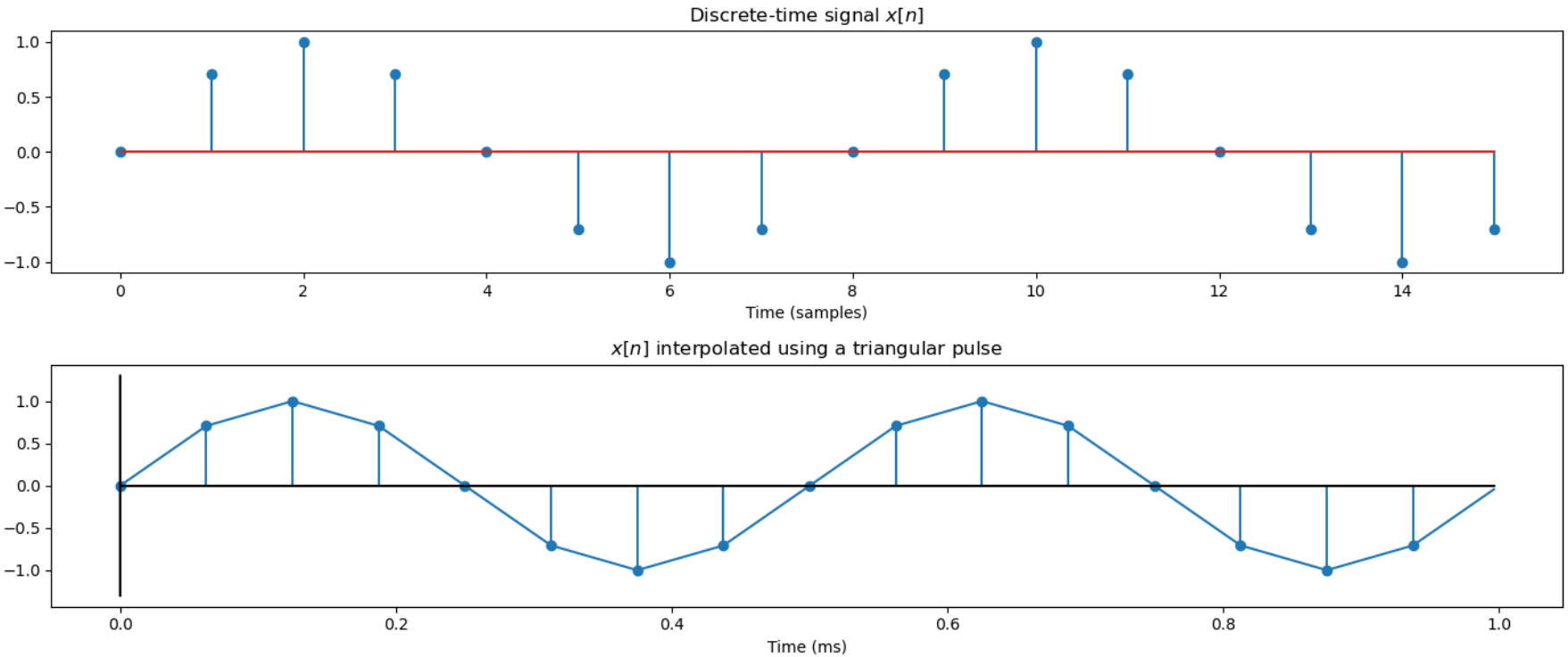
The rectangular pulse has the disadvantage that $y(t)$ is discontinuous. We can eliminate the discontinuities by using a triangular pulse:

$$p(t) = \begin{cases} 1 - \frac{|t|}{T_S} & -T_S \leq t < T_S \\ 0 & \text{otherwise} \end{cases}$$



Triangular pulses = Piece-wise linear interpolation

The result is a piece-wise linear interpolation of the digital signal:



Although the signal is continuous, its derivatives are discontinuous. Images look pretty good like this, but audio signals still sound noisy (the ear is more sensitive to high frequencies than the eye).

Aliasing caused by imperfect interpolation

For example, suppose we start with $x[n] = \cos\left(\frac{n\pi}{8}\right)$, and interpolate using a triangular pulse with $T_s = \frac{1}{8000}$ of a second. We wind up with

$$y(t) = a\cos(2\pi 1000n) + b\cos(2\pi 7000n) + c\cos(2\pi 9000n) + \cdots,$$

where b and c are much smaller than they were for the rectangular pulse, but still not zero.

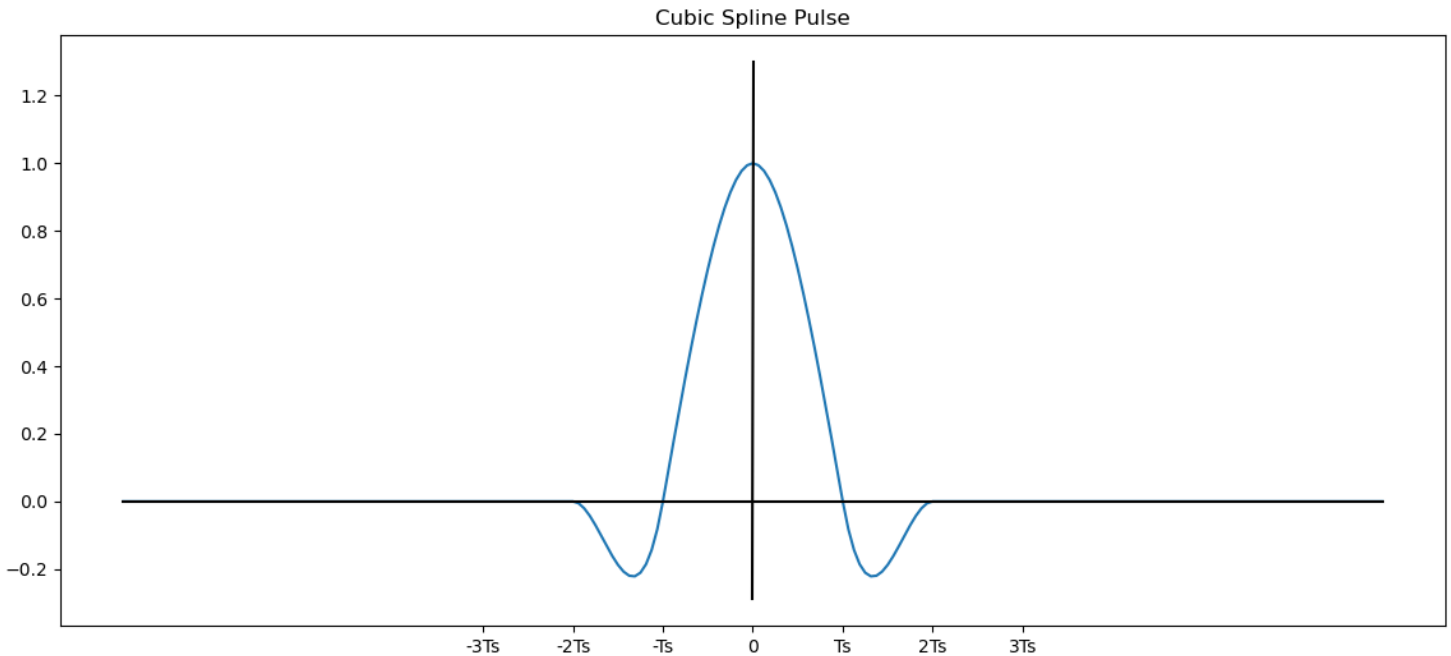
Cubic spline pulses

The triangular pulse has the disadvantage that, although $y(t)$ is continuous, its first derivative is discontinuous. We can eliminate discontinuities in the first derivative by using a cubic-spline pulse:

$$p(t) = \begin{cases} 1 - \frac{3}{2}\left(\frac{|t|}{T_S}\right)^2 + \frac{1}{2}\left(\frac{|t|}{T_s}\right)^3 & 0 \leq |t| \leq T_S \\ -\frac{3}{2}\left(\frac{|t|-2T_s}{T_S}\right)^2\left(\frac{|t|-T_s}{T_S}\right) & T_S \leq |t| \leq 2T_S \\ 0 & \text{otherwise} \end{cases}$$

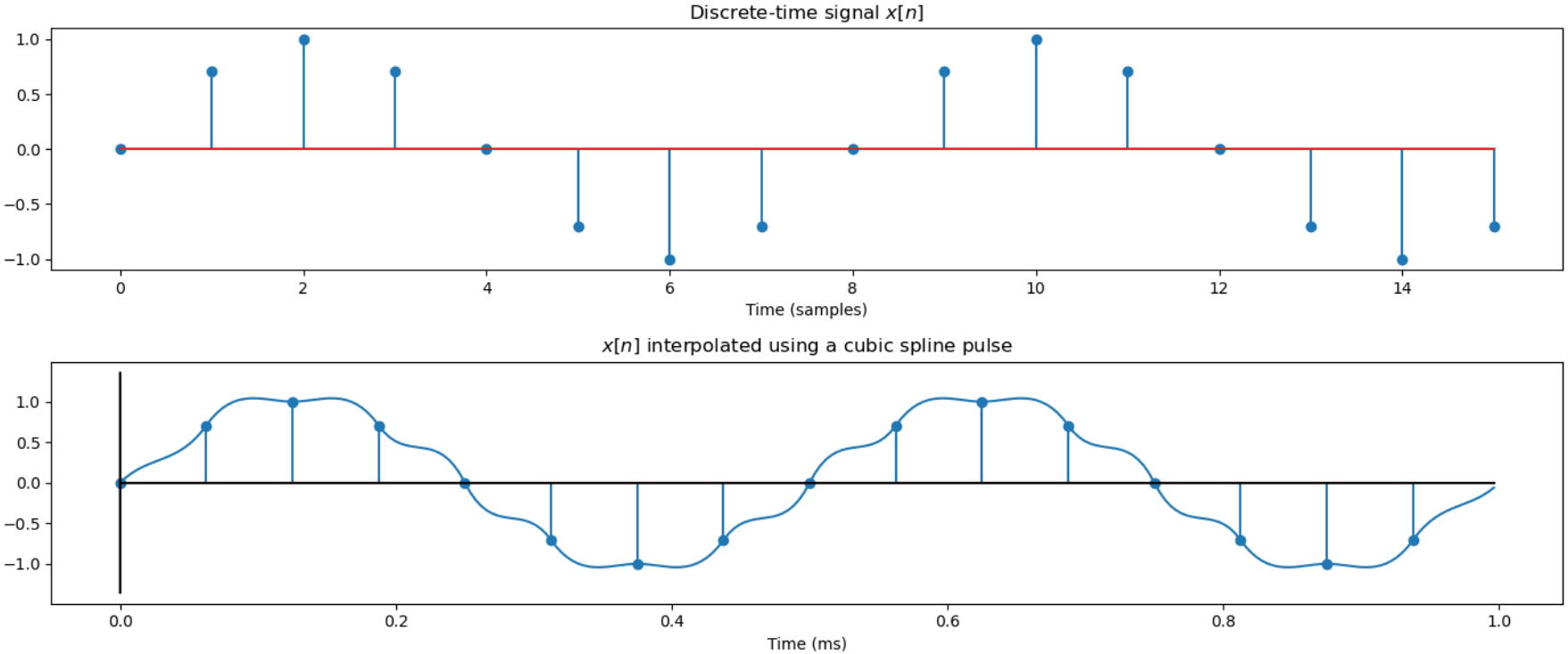
Cubic spline pulses

The triangular pulse has the disadvantage that, although $y(t)$ is continuous, its first derivative is discontinuous. We can eliminate discontinuities in the first derivative by using a cubic-spline pulse:



Cubic spline pulses = Piece-wise cubic interpolation

The result is a piece-wise cubic interpolation of the digital signal:



Aliasing caused by imperfect interpolation

For example, suppose we start with $x[n] = \cos\left(\frac{n\pi}{8}\right)$, and interpolate using a cubic-spline pulse with $T_s = \frac{1}{8000}$ of a second. We wind up with

$$y(t) = a\cos(2\pi 1000n) + b\cos(2\pi 7000n) + c\cos(2\pi 9000n) + \cdots,$$

where b and c are much smaller than they were for the triangular pulse, but still not zero.

Sinc pulses

The cubic spline has no discontinuities, and no slope discontinuities, but it still has discontinuities in its second derivative and all higher derivatives. Can we fix those?
The answer: yes! If we keep smoothing $p(t)$, we can find a signal such that:

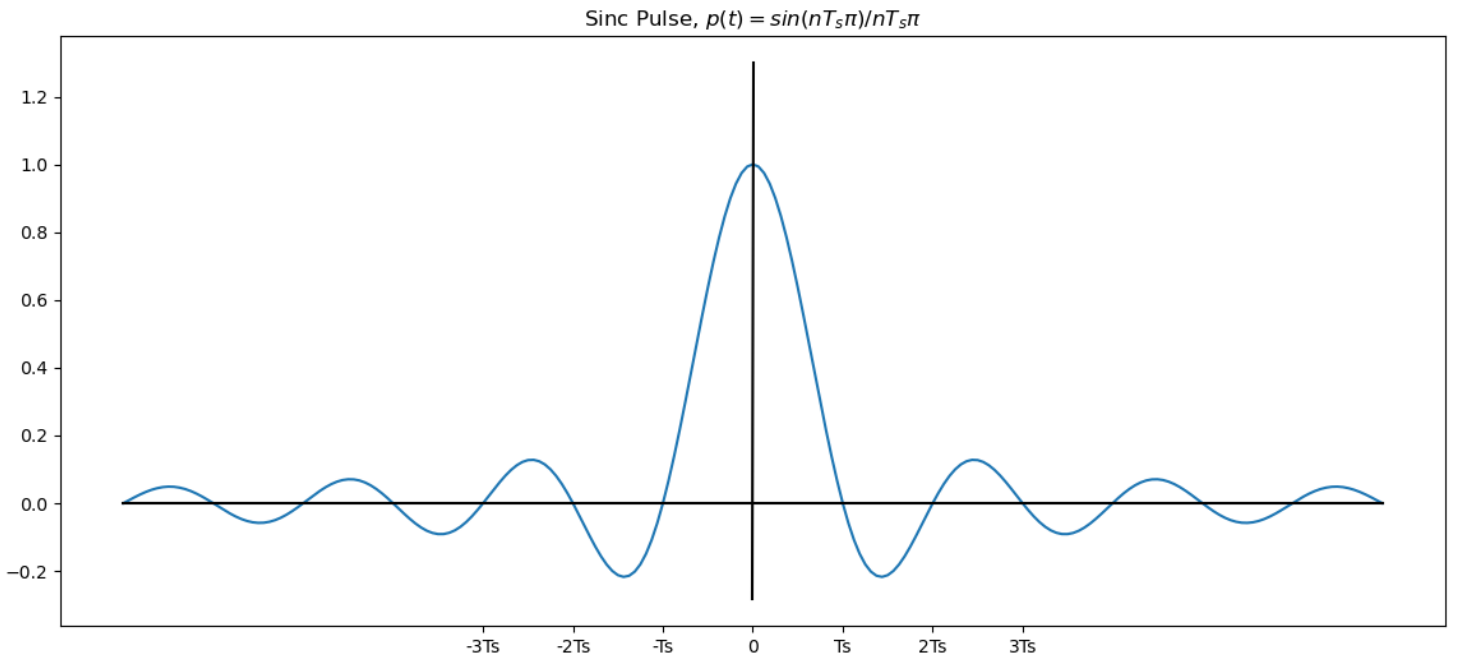
- $p(0) = 1$
- $p(nT_s) = 0$ for all integers $n \neq 0$
- All of the derivatives of $p(t)$ are continuous everywhere

The resulting signal is called a "sinc function."

Sinc pulses

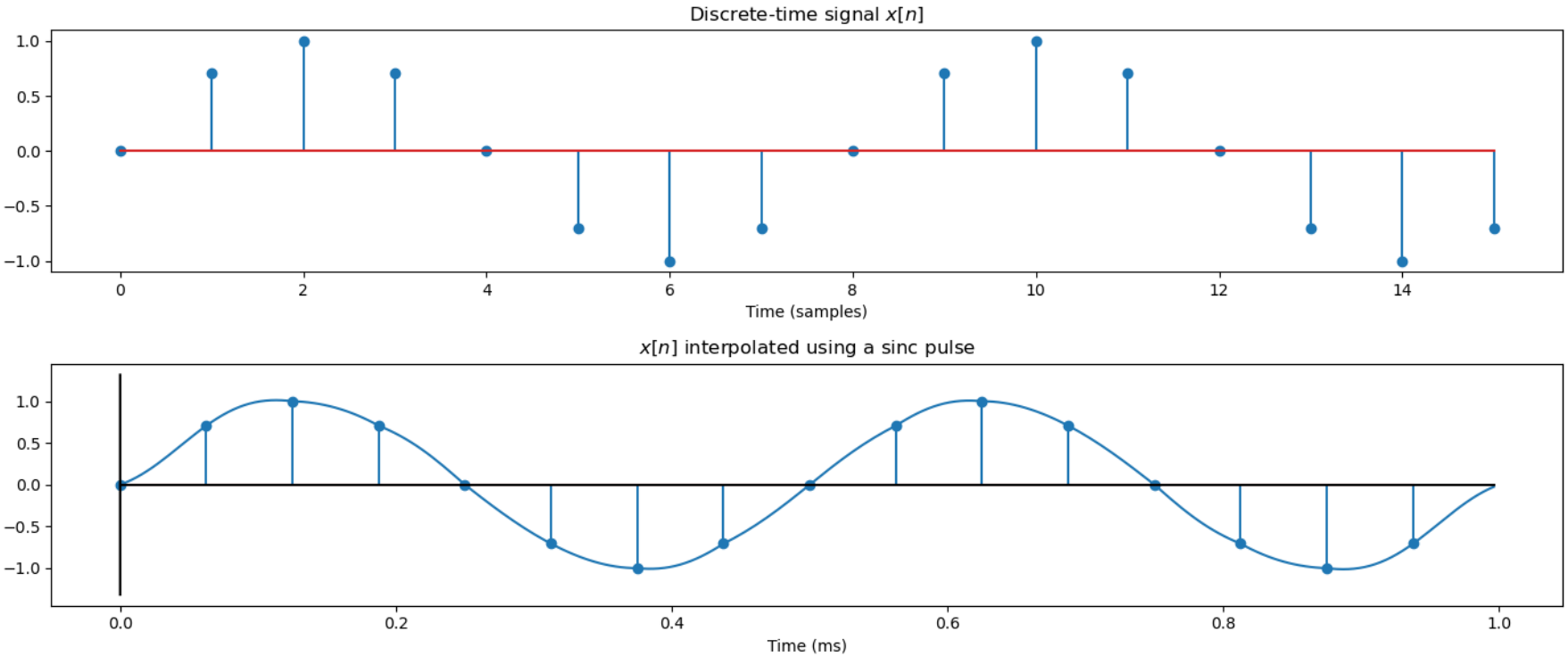
We can reconstruct a signal that has no discontinuities in any of its derivatives by using an ideal sinc pulse:

$$p(t) = \frac{\sin(\pi t/T_S)}{\pi t/T_S}$$



Sinc pulse = ideal bandlimited interpolation

The resulting signal has no high-frequency noise: it contains only frequencies below the Nyquist frequency. We call this an "ideal bandlimited interpolation."



Sinc interpolation = no aliasing!

For example, suppose we start with $x[n] = \cos\left(\frac{n\pi}{8}\right)$, and interpolate using a sinc function with $F_s = 8000$ samples/second. We wind up with exactly:

$$y(t) = \cos\left(1000\pi n\right)$$

Perfect!

Try the quiz!

Go to the course web page, and try today's quiz!

Outline

Changing the sampling rate of a signal

Suppose we have an audio signal ($x[n]$) sampled at 11025 samples/second, but we really want to play it back at 44100 samples second. We can do that by creating a new signal, $y[n]$, at $M = 4$ times the sampling rate of $x[n]$:

$$y[n] = \begin{cases} x[n/M] & n = \text{integer multiple of } M \\ \text{interpolated value} & \text{otherwise} \end{cases}$$

Upsampling

We split this process into two steps. First, **upsampling** means that we just insert zeros between the samples of $x[n]$:

$$u[n] = \begin{cases} x[n/M] & n = \text{integer multiple of } M \\ 0 & \text{otherwise} \end{cases}$$

Interpolation

Second, we generate the missing samples by interpolation:

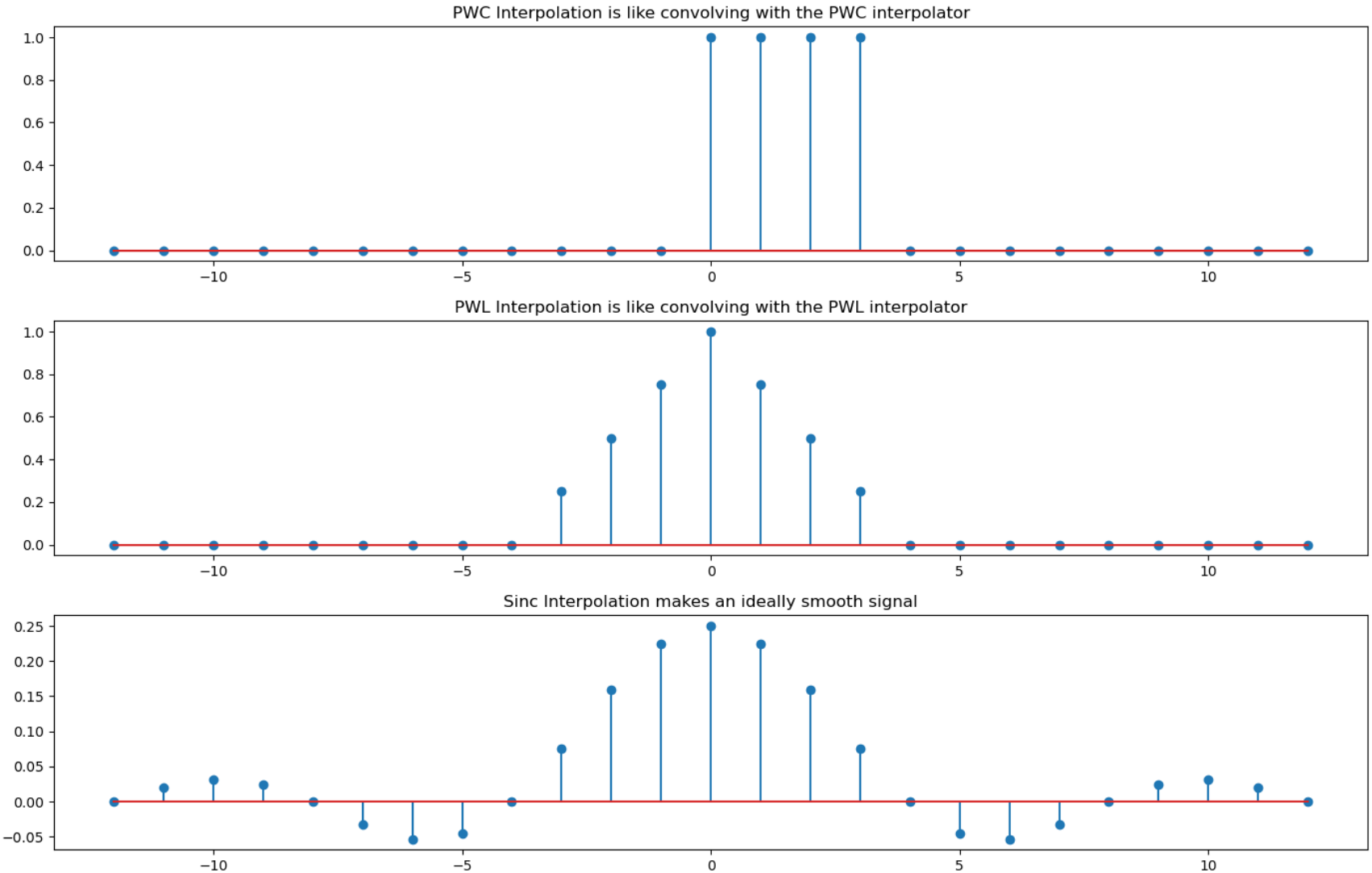
$$y[n] = \sum_{m=-\infty}^{\infty} u[m]p[n-m]$$
$$= \begin{cases} x[n/M] & n = \text{integer multiple of } M \\ \text{interpolated value} & \text{otherwise} \end{cases}$$

The second line of the equality holds if

$$p[n] = \begin{cases} 1 & n = 0 \\ 0 & n = \text{nonzero integer multiple of } M \\ \text{anything} & \text{otherwise} \end{cases}$$

Interpolation Kernels

All of these interpolation kernels satisfy the condition on the previous slide:



Outline

Summary

- Piece-wise constant interpolation = interpolate using a rectangle
- Piece-wise linear interpolation = interpolate using a triangle
- Cubic-spline interpolation = interpolate using a spline
- Ideal interpolation = interpolate using a sinc