Interpolation and Pulse Shaping

- **Prof. Brian L. Evans**
- **Dept. of Electrical and Computer Engineering**
- **The University of Texas at Austin**

Lecture 7

Discrete-to-Continuous Conversion

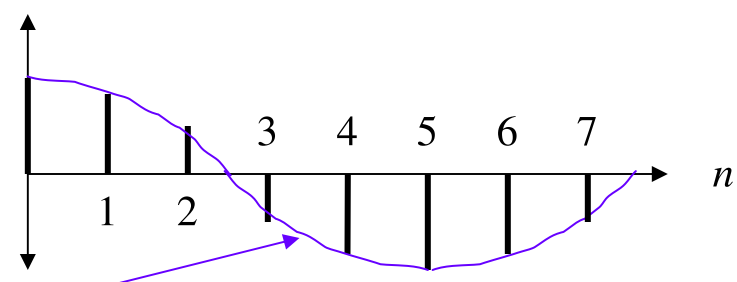
- **Interpolate a smooth continuous-time function through a sequence of samples ("connect the dots")**

If $f_0 < ½ f_s$, then

$$y[n] = A \cos(2\pi f_0 T_s n + \phi)$$

would be converted into

$$y(t) = A \cos(2\pi f_0 t + \phi)$$



Otherwise, aliasing has occurred, and the converter would reconstruct a cosine wave whose frequency is equal to the aliased positive frequency that is less than $½ f_s$

Discrete-to-Continuous Conversion

- **General form of interpolation is sum of weighted pulses**

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

Sequence $y[n]$ converted into continuous-time signal that is an approximation of $y(t)$

Pulse function $p(t)$ could be rectangular, triangular, parabolic, sinc, truncated sinc, raised cosine, etc.

Pulses overlap in time domain when pulse duration is greater than or equal to sampling period T_s

Pulses generally have unit amplitude and/or unit area

Above formula is discrete-time convolution for each value of t

Interpolation From Tables

- **Using mathematical tables of numeric values of functions to compute a value of the function**

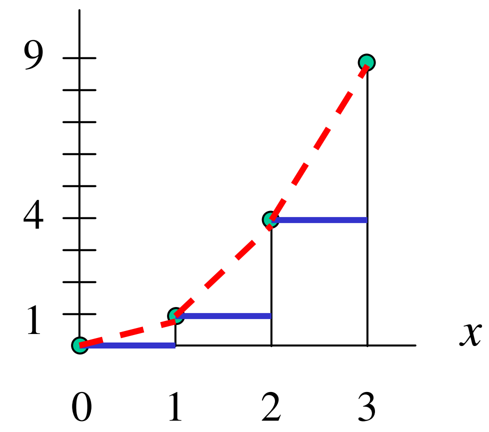
- **Compute $f(1.5)$ from table**

  Zero-order hold: take value to be $f(1)$ to make $f(1.5) = 1.0$ ("stairsteps")

  Linear interpolation: average values of nearest two neighbors to get $f(1.5) = 2.5$

  Curve fitting: fit the three points in table to function x^2 to compute $f(1.5) = 2.25$

x	$f(x)$
0	0.0
1	1.0
2	4.0
3	9.0

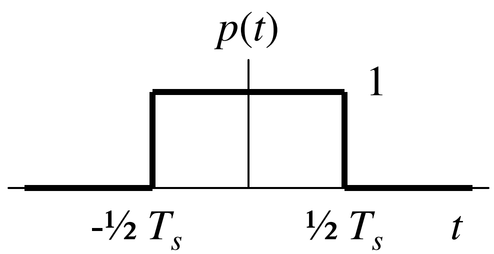


Rectangular Pulse

- **Zero-order hold**

  Easy to implement in hardware or software

$$p(t) = \text{rect}\left(\frac{t}{T_s}\right) = \begin{cases} 1 & \text{if } -\frac{1}{2}T_s < t \le \frac{1}{2}T_s \\ 0 & \text{otherwise} \end{cases}$$



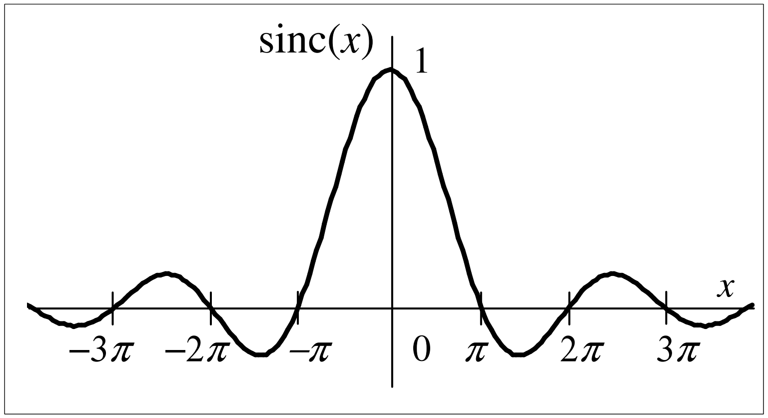
  The Fourier transform is

$$P(f) = T_s \,\text{sinc}(\pi\, f\, T_s) = T_s \frac{\sin(\pi\, f\, T_s)}{\pi\, f\, T_s} \quad \text{where } \text{sinc}(x) = \frac{\sin(x)}{x}$$

  In time domain, no overlap between $p(t)$ and adjacent pulses $p(t - T_s)$ and $p(t + T_s)$

  In frequency domain, sinc has infinite two-sided extent; hence, the spectrum is not bandlimited

Sinc Function



$$\operatorname{sinc}(x) = \frac{\sin(x)}{x}$$

How to compute sinc(0)?

As $x \to 0$, numerator and denominator are both going to 0. How to handle it?

Even function (symmetric at origin)

Zero crossings at $x = \pm\pi, \pm 2\pi, \pm 3\pi, \ldots$

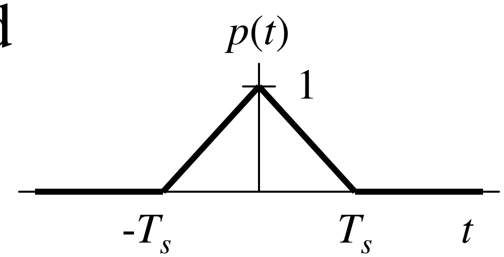
Amplitude decreases proportionally to 1/x

Triangular Pulse

- **Linear interpolation**

  It is relatively easy to implement in hardware or software, although not as easy as zero-order hold

$$p(t) = \Delta\left(\frac{t}{T_s}\right) = \begin{cases} 1 - \frac{|t|}{T_s} & \text{if } T_s < t \le T_s \\ 0 & \text{otherwise} \end{cases}$$



  Overlap between $p(t)$ and its adjacent pulses $p(t - T_s)$ and $p(t + T_s)$ but with no others

- **Fourier transform is** $P(f) = T_s \,\mathrm{sinc}^2(f\, T_s)$

  How to compute this? *Hint:* The triangular pulse is equal to 1 / T_s times the convolution of rectangular pulse with itself

  In frequency domain, sinc^2 has infinite two-sided extent; hence, the spectrum is not bandlimited

Sinc Pulse

- **Ideal bandlimited interpolation**

$$p(t) = \operatorname{sinc}\left(\frac{\pi}{T_s} t\right) = \frac{\sin\left(\frac{\pi}{T_s} t\right)}{\frac{\pi}{T_s} t} \Leftrightarrow P(f) = \frac{1}{T_s} \operatorname{rect}\left(\frac{f}{T_s}\right) \qquad W = \frac{1}{2\,T_s}$$

  In time domain, infinite overlap between other pulses

  Fourier transform has extent $f \in [-W, W]$, where

  $P(f)$ is ideal lowpass frequency response with bandwidth W

  In frequency domain, sinc pulse is bandlimited

- **Interpolate with infinite extent pulse in time?**

  Truncate sinc pulse by multiplying it by rectangular pulse

  Causes smearing in frequency domain (multiplication in time domain is convolution in frequency domain)

Raised Cosine Pulse: Time Domain

- **Pulse shaping used in communication systems**

$$p(t) = \underbrace{\mathrm{sinc}\left(\frac{t}{T_s}\right)}_{\text{ideal lowpass filter impulse response}} \; \underbrace{\frac{\cos(2\pi\,\alpha\,W\,t)}{1 - 16\,\alpha^2\,W^2\,t^2}}_{\text{Attenuation by } 1/t^2 \text{ for large } t \text{ to reduce tail}}$$

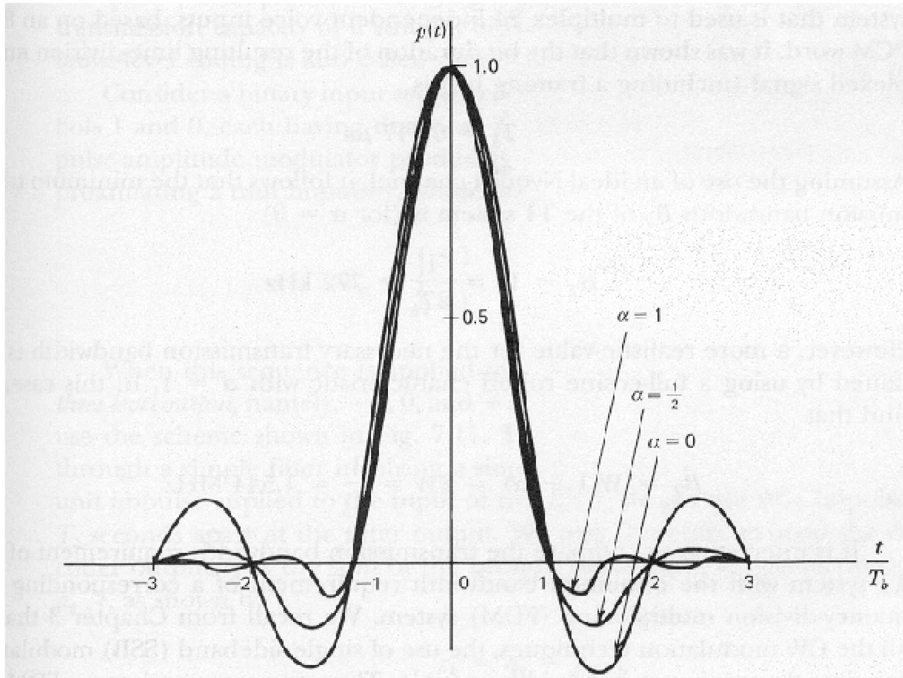
ideal lowpass filter impulse response

Attenuation by $1/t^2$ for large t to reduce tail

W is bandwidth of an ideal lowpass response

$\alpha \in [0, 1]$ rolloff factor

Zero crossings at $t = \pm\, T_s\,, \pm\, 2\,T_s\,, \ldots$



- **See handout G in reader on raised cosine pulse**

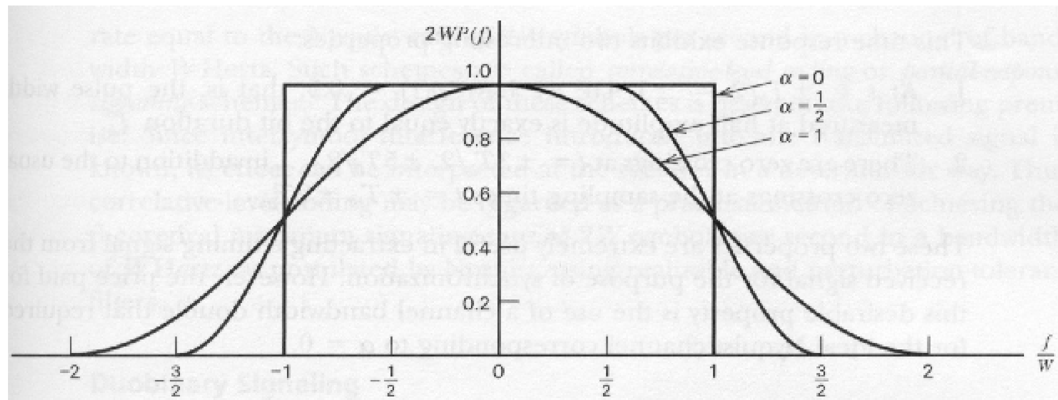
Raised Cosine Pulse Spectra

- **Pulse shaping used in communication systems**

Bandwidth:

$(1 + \alpha)\, W = 2\, W - f_1$

f_1 transition begins from ideal lowpass response to zero



$$P(f) = \begin{cases} \dfrac{1}{2\mathrm{W}} & \text{if } 0 \le |f| < f_1 \\ \dfrac{1}{4\mathrm{W}}\left(1 - \sin\left(\dfrac{\pi(|f| - W)}{2W - 2f_1}\right)\right) & \text{if } f_1 \le |f| < 2W - f_1 \\ 0 & \text{otherwise} \end{cases}$$

$$W = \frac{1}{2\,T_s}$$

$$\alpha = 1 - \frac{f_1}{W}$$

Full Cosine Rolloff

- **When α = 1**

$$p(t) = \frac{\operatorname{sinc}(4\,W\,t)}{1 - 16\,W^2\,t^2} \quad \Leftrightarrow \quad P(f) = \begin{cases} \frac{1}{4\mathrm{W}}\left(1 + \cos\left(\frac{\pi\, f}{2\,W_1}\right)\right) & \text{if } 0_1 \le |\, f\,| < 2W \\ 0 & \text{otherwise} \end{cases}$$

At $t = \pm \frac{1}{2}\, T_s = \pm\, 1 / (4\, W)$, $p(t) = \frac{1}{2}$, so that the pulse width measure at half of the maximum amplitude is equal to T_s

Additional zero crossings at $t = \pm\, 3/2\, T_s$, $\pm\, 5/2\, T_s$, …

- ***Advantages in communication systems?***

Easier for receiver to extract timing signal for synchronization

- ***Drawbacks in communication systems?***

Transmitted bandwidth doubles over sinc pulse

Bandwidth generally scarce in communications systems

DSP First Demonstration

- **Web site:** http://users.ece.gatech.edu/~dspfirst
- **Sampling and aliasing demonstration (Chapter 4)**

  Sample sinusoid $y(t)$ to form $y[n]$

  Reconstruct sinusoid using rectangular, triangular, or truncated sinc pulse $p(t)$ by

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

- **Which pulse gives the best reconstruction?**
- **Sinc pulse is truncated to be four sampling periods long. Why is the sinc pulse truncated?**
- **What happens as the sampling rate is increased?**