Lecture 4 Sampling and Aliasing

March 9, 2026

Notes by Jenna May

Exam Information

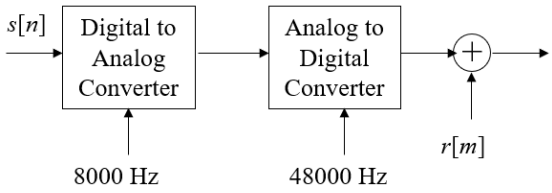
- Split between classroom and lab room - room assignments based on last name posted on Canvas
- Open book, open note, open laptop
- NO networking, AI, phones, etc
- Charge devices beforehand
- Download course reader and notes

Review of Previous Lecture

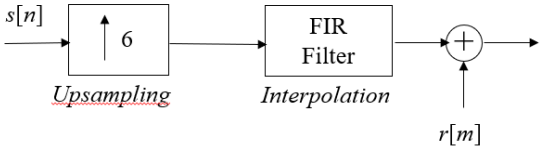
- Sampling a continuous-time signal x(t) at sampling rate fs
    - Slide 4-3 Time domain: Sampling modeled as a switch closing and opening instantaneously at integer multiples of sampling time Ts where Ts = 1/ / fs
    - Slide 4-4: Frequency domain: sampling causes the spectrum of x(t) to pass through (scaled by 1/Ts) and produces copies of the original spectrum at offsets of integer multiples of fs from -infinity to infinity (periodic spectrum)
- Reconstruction of the original signal from sampled signal
    - Slide 4-4: Apply lowpass filtering on sampled signal to attenuate copies. The lowpass filter would pass frequencies (-½) fs < f < (½).
    - Slide 4-5: Reconstruction is only accurate over frequencies (-½) fs to (½) fs due to the Sampling Theorem that says fs > 2 fmax which means fmax < (½) fs in non-negative frequencies, which means (-½) fs < f < (½) fs over all frequencies.
    - Slide 4-6: Sample-and-hold reconstruction: equivalent to using a lowpass FIR averaging filter which has a rectangular pulse for the impulse response and a two-sided sinc pulse for the frequency domain
        - Convolution in the time domain of the sampled signal and pulse shape results in holding magnitude of last sample until next sample occurs
        - Low run-time complexity
        - Would apply a lowpass continuous-time IIR filter after sample-and-hold to attenuate frequencies outside (-½) fs < f < (½) introduced by the sample-and-hold operation which in the time domain would round off (smooth out) corners
    - Linear Interpolation: triangular pulse shape with width of 2Ts in time domain
        - Pulse shape could be causal, 0 to 2Ts, or centered at origin, -Ts to Ts
        - Double the run-time implementation complexity of sample-and-hold
- Aliasing
    - When sampling, copies of the original signal will include aliased components in the range -½ fs to ½ fs
    - We will see these copies when we reconstruct, resulting in a our reconstructed signal differing from our original

Increasing Sampling Rates

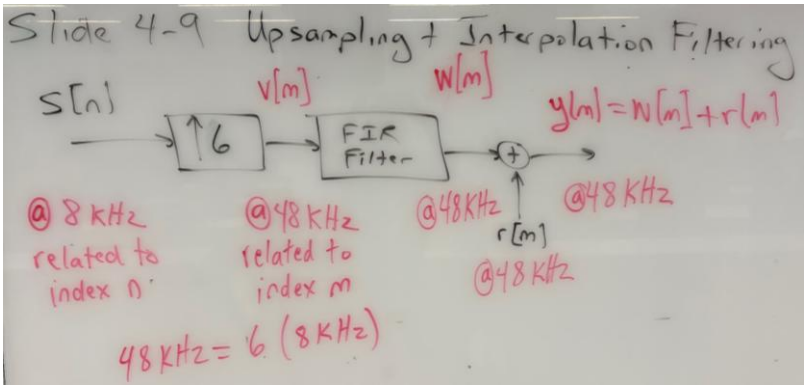
- Slide 4-9: May want to change sample rates of signal
    - Ex: combine speech signal sampled at 8kHz with audio signal sampled at 48 kHz
- Slide 4-9: Using a cascade of a D/A and an A/D converter to change the sampling rate takes a lot of power (proportional to the sampling rate times 2^bits) and is inefficient.

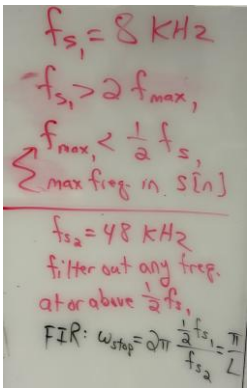


- Slide 4-9: More efficient: to upsample by L (which copies each input sample s[n] to the output then followed by L-1 zeros) and then put upsampled result through an FIR interpolation filter (which can be implemented in discrete-time in software or hardware)



- Upsampling by L
    - Frequency domain: introduces L-1 copies of our signal's spectrum at offsets equal to integer multiples of the original sampling rate before upsamplinge
    - Need FIR filter to interpolate to fill in the zero values with meaningful values that connect the values of s[n] in v[m] and remove high frequencies artificially introduced by the upsampling operation
    - The FIR filter has a stopband frequency of pi/L to attenuate artificial high frequencies above (½) fs1 introduced by upsampling
    - Similar to interpolation in reconstruction





Interpolation Demo

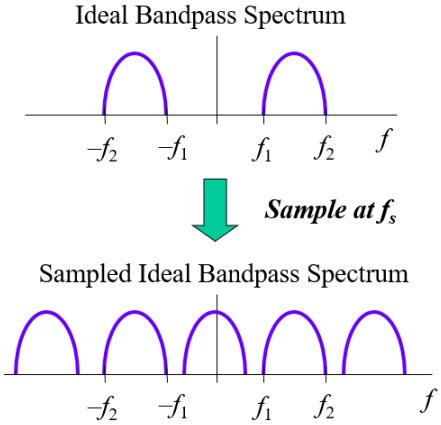
- Slide 4-10: Upsampling sinusoidal signal and applying three different interpolation filters
- Question #1: Which of the four upsampled signals sounds the best?
- Question #2: Which of the four upsampled signals sounds the closest to the original sinusoidal signal prior to upsampling?

Bandpass sampling (Slide 4-11)

- Can reduce power consumption and simulation speed for bandpass signals
- Intentionally use aliasing when sampling a bandpass signal to reduce the sampling rate and hence lower run-time complexity (power consumption proportional to sampling rate)
- Want good bandpass filter before sampling to remove out-of-band noise and interference (like homework problem 3.1)
- Sample rate must be greater than bandpass bandwidth (extent in positive frequency)

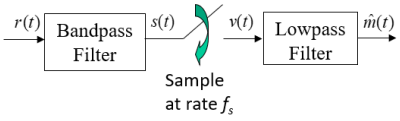
  $f_s > f_2 - f_1$
- For replica to be centered at origin after sampling,

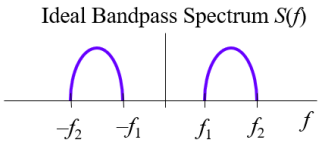
  $f_{center} = \frac{1}{2}(f_1 + f_2) = k f_s$
- After sampling, use a lowpass filter to extract the baseband signal



- Practical issues of sampling clock tolerance, BP and LP filter design, noise effects

Bandpass sampling example (Slide 4-12)





Bandpass sampling

Bandwidth $f_2 - f_1$

$f_s > f_2 - f_1$

$f_{center} = \frac{1}{2}(f_1 + f_2) = k f_s$

Extract IEEE 802.11a Wi-Fi 2.4 GHz Channel 1

$f_1 = 2.401$ GHz

$f_2 = 2.423$ GHz

Bandwidth = 0.022 GHz

$f_c = 2.412$ GHz

Sampling theorem

$f_s > 2 f_2$ e.g. $f_s = 4.86$ GHz

Bandpass sampling

$f_s = 0.036$ GHz with $k = 67$

135x more efficient

Software Defined Radio (Slides 4-13 and 4-14)

- Within unlicensed 2.4GHz band, multiple different usages, including Bluetooth Low Energy (BLE), Wi-Fi, etc., use different frequencies
- Shift bandpass spectra down to touch at the origin NOT centered at the origin.

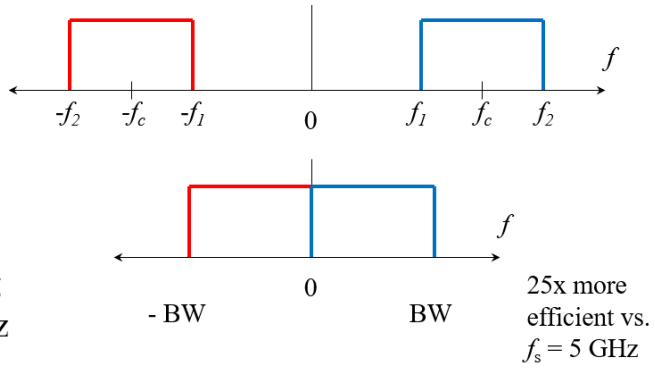
Extract band for processing

$f_1 = 2.4$ GHz

$f_2 = 2.5$ GHz

BW = 0.1 GHz

Bandpass sampling with $f_s = 0.2$ GHz



- Then extract band(s) of interest
- Up to 25x more efficient