ECE 3704 Lecture 5

Sampling and Reconstruction

- Recall that hybrid systems are ones with both CT & DT signals. Such systems require the ability to
  - convert from CT to DT: Sampling or analog to Digital Conversion (ADC)
  - convert from DT to CT: Reconstuction or Digital to Analog Conversion (DAC)

- Ideal Sampling Theory
  - impulse train $x_p(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT)$

    is a periodic function with period T. It has a CTFS representation $x_p(t) = \sum_{k=-\infty}^{\infty} a_k e^{j\frac{2\pi}{T}kt}$ where $a_k = \frac{1}{T}$

    Its Fourier transform is

    $$X_p(j\omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi}{T}k\right)$$

  - Given an arbitrary signal $x(t)$ we can represent sampling as multiplication by impulse train

    $$x_s(t) = x(t) \cdot x_p(t)$$
    $$= \sum_{n=-\infty}^{\infty} x(t)\,\delta(t-nT) = \sum_{n=-\infty}^{\infty} \underbrace{x(nT)}_{x[n]}\,\delta(t-nT)$$

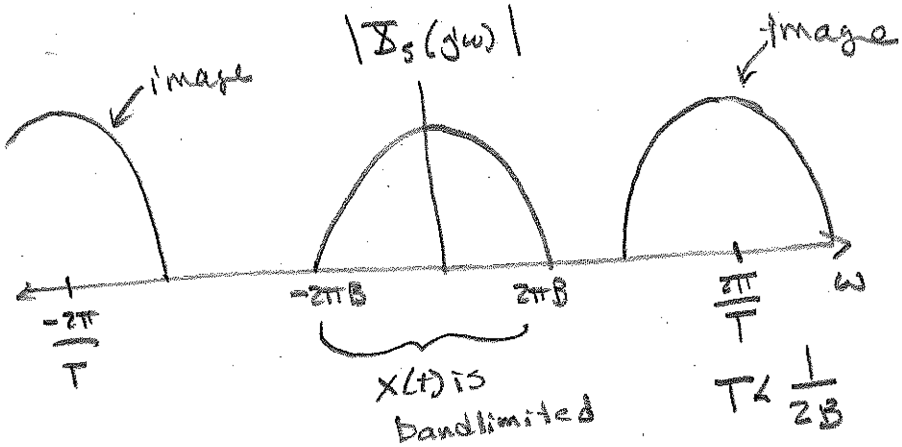
    In Frequency Domain, multiplication is convolution.

    $$X_s(j\omega) = \frac{1}{2\pi} X_p(j\omega) * X(j\omega)$$
    $$= \frac{1}{T} \sum_{n=-\infty}^{\infty} X\left(j\left(\omega - \frac{2\pi}{T}n\right)\right)$$

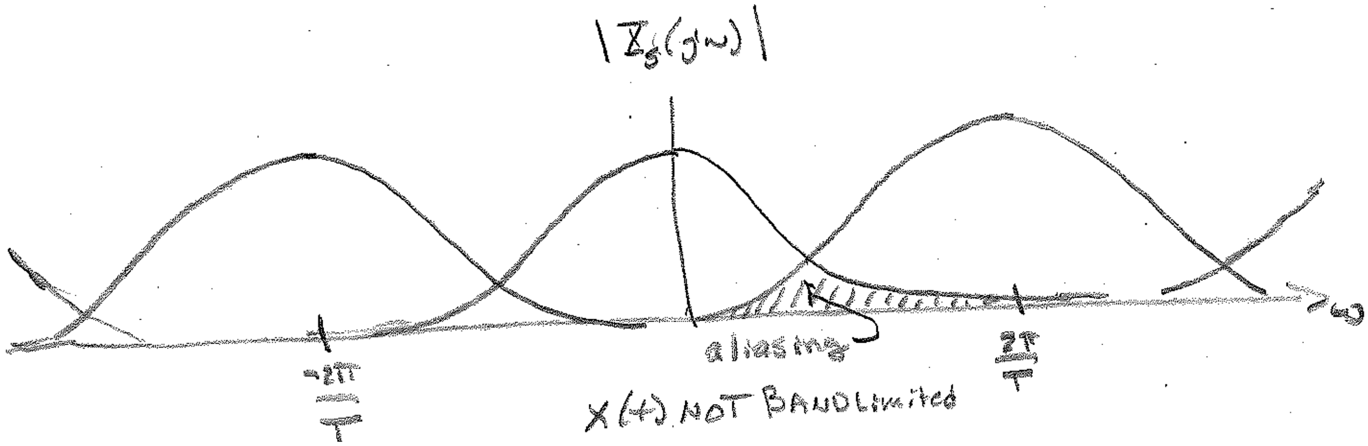
    The frequency content of the sampled signal consists of copies of the fregency content of original signal called <u>images</u>

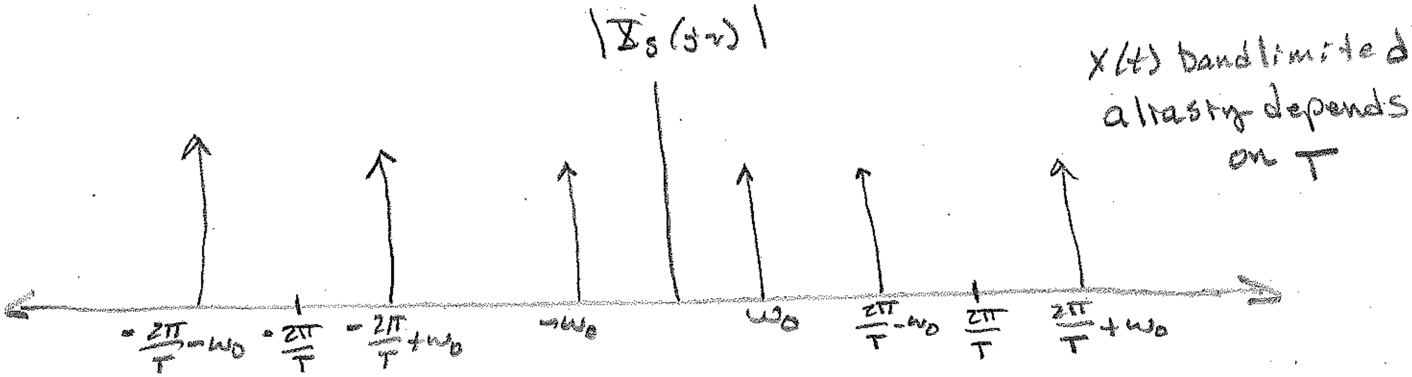
#5 2

- To ensure there is no distortion in the signal we require that $T < \frac{1}{2B}$, where B is the bandwidth of $x(t)$. This is the Nyquist Criteria.
- If B is finite, then $x(t)$ is bandlimited and we can choose T to meet the Nyquist rate.
- If $x(t)$ is not bandlimited, or $T > \frac{1}{2B}$ then we will get aliasing.



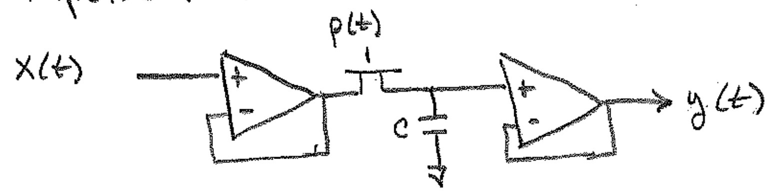
No aliasing





— Practical Sampling. While mathematically simple and intuitively appealing, ideal sampling cannot be implemented in practice because we cannot generate ideal impulses.

- we can generate short pulses however to approximate the impulse train. This is called Sample & Hold.



where $p(t)$ is the pulse train with pulse width $P < T$ $\quad p(t) = \sum_{n=-\infty}^{\infty} u(t-nT) - u(t-P-nT)$

- we can model this as well since $p(t) = x_p(t) * [u(t) - u(t-P)]$

and $\quad x_s(t) = \{x_p(t) * [u(t) - u(t-P)]\} \cdot x(t)$

In frequency domain $\quad X_s(j\omega) = \{X_p(j\omega) \cdot \mathcal{F}\{u(t) - u(t-P)\}\} * X(j\omega)$

— To store the output of the Sample & Hold, $y(t)$ we need to quantize it into N bits, e.g. $N = 8, 12, 24$. The mathematical model for this process is outside the scope of 3704 (take DSP) so we will assume $x(nT) \in \mathbb{R}$. See Successive Approximation.

— If we have no control over the input $x(t)$ and have to fix the sample time, we need to force $x(t)$ to be approximately bandlimited using a lowpass filter called an <u>antialiasing</u> filter.

$$X(\omega) \longrightarrow \boxed{H_{Lp}(\omega)} \longrightarrow H_{Lp}(\omega) X(\omega) \longrightarrow \boxed{ADC} \longrightarrow x[n].$$

where the cutoff ω_c is relative to $\frac{2\pi}{T}$.

— A DAC must do the opposite of ADC

1. Convert N-bit representation of $x[n]$ to $x(nT) \in \mathbb{R}$
2. remove the images from $x(nT)$ by low pass filtering aka. reconstruction Filter.

— two common approaches are to use PWM and Resistor ladder followed by Filter.

- PWM generates pulses whose widths relative to T are fractions of $x[n]$, e.g. $N=8$ bits.

uses 1 pin

$x[n] = 0$ pulse width $= 0$

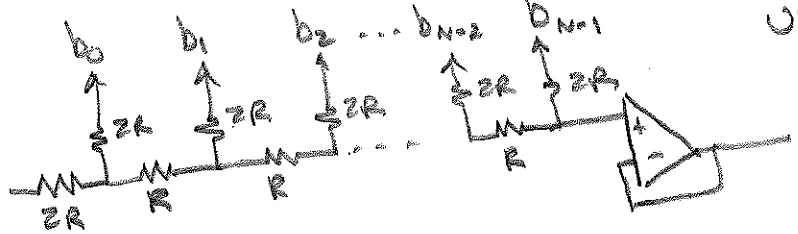
$x[n] = 128$ pulse width $= \frac{T}{2}$

$x[n] = 256$ pulse width $= T$

0, 128, 256, 128, 0 …

uses N pins.

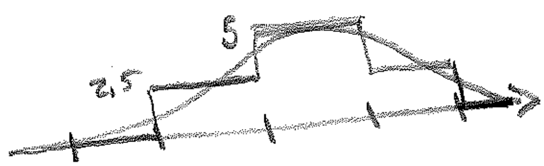
- R-2R ladder



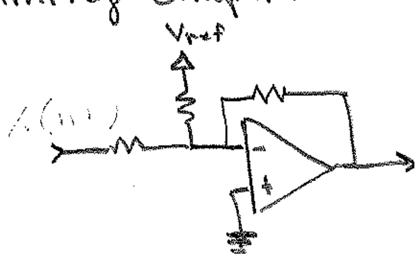
$V_o = V_{ref} \frac{V}{2^N}$ $V = b_{N-1}, b_{N-2} \cdots b_2\, b_1\, b_0\,{}_2$ (base 2 / binary)

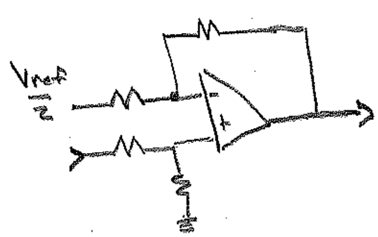
0, 128, 256, 128, 0

$V_{ref} = 5V$

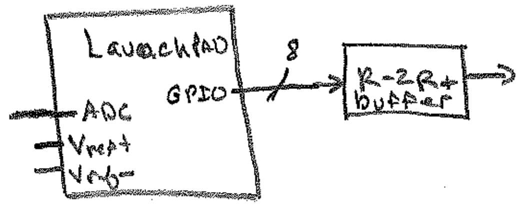


— IN both cases the output is positive. It can be followed by a summing or differential amplifier to shift up/down and scale.





— LAB #1. The goal of Lab I is to build a sampling and reconstuction system using protoboard and MCU, and implement a DT identity (allpass) system.



1. Configure the MCU to use one pin (P5.5) in repeat mode using DriverLib, with pin P5.6 and P5.7 as Vref+ and Vref− respectively. Use automatic iteration with interrupt.

2. Configure the MCU to use a GPIO port for DAC (P4) i.e. as output

3. Write an interrupt rootine to copy sample to internal 16bit variable. Then copy that to output P4. This implements an identity system.

4. Characterize the system with respect to aliasing using AD2 to generate sinusoidal input, compare to DAC output.

- See how fast you can get this system to sample.
- We will build on this in LAB 2 and LAB 3.