Unit 2: Symbol Mapping and TX Filtering

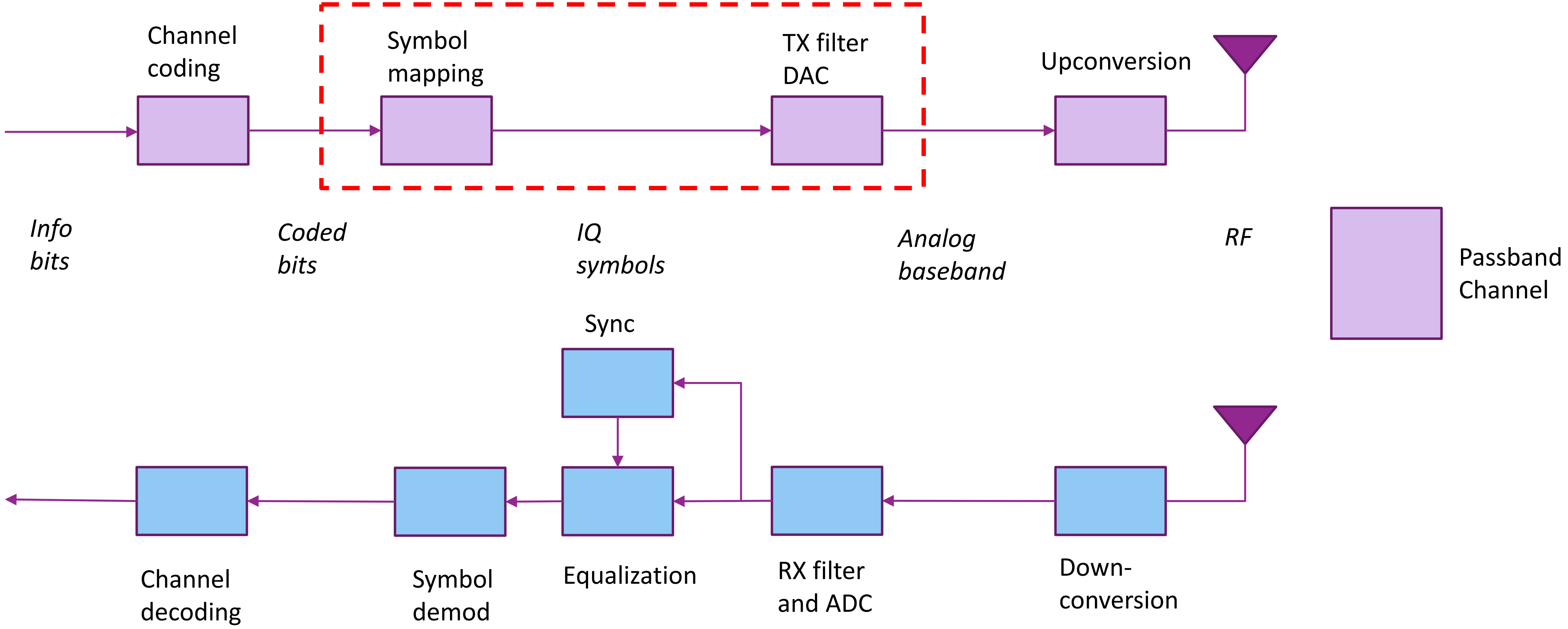
EL-GY 6013: DIGITAL COMMUNICATIONS

PROF. SUNDEEP RANGAN

Learning Objectives

- Describe the steps in symbol mapping and pulse shaping
- Describe the common modulation methods:
  - BPSK, QPSK, M-QAM.
  - For each, compute the minimum distance and symbol energy
- Compute the data rate as a function of the modulation and symbol rate
- Compute the TX spectrum given pulse shape and DTFT of the symbols
- Compute the PSD as a function of the pulse shape and symbol energy
- Specify TX filter requirements based on bandwidth and other requirements
- Describe the ideal sinc pulse in time domain and frequency domain
- Design a digital and analog filter given bandwidth constraints

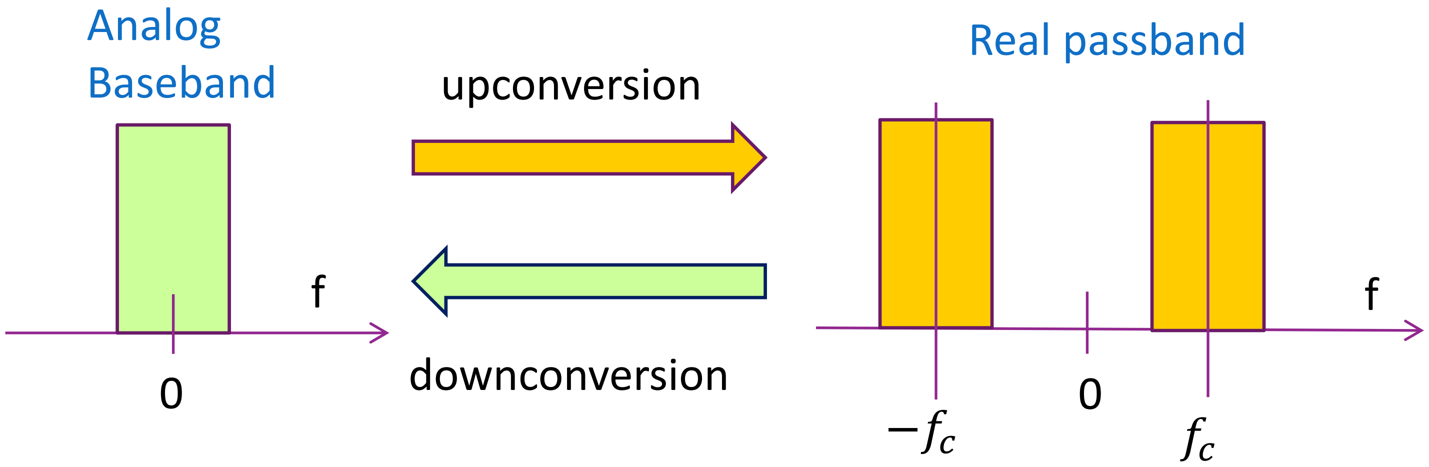
This Unit


Channel coding
Symbol mapping
TX filter DAC
Upconversion
Info bits
Coded bits
IQ symbols
Analog baseband
RF
Passband Channel
Sync
Channel decoding
Symbol demod
Equalization
RX filter and ADC
Down-conversion

Outline

- Symbol mapping
- DAC and pulse shaping
- Fourier analysis and bandwidth of TX filtering
- Power spectral density analysis
- Sinc pulse and Ideal low pass filtering
- Digitally implementing pulse shaping

Last Unit: Up- and Down-Conversion



- Upconversion in TX: Convert an analog baseband IQ to real passband
- Downconversion in RX: Convert real passband to analog IQ
- But, baseband signal is complex and analog
- How do we transmit digital information?

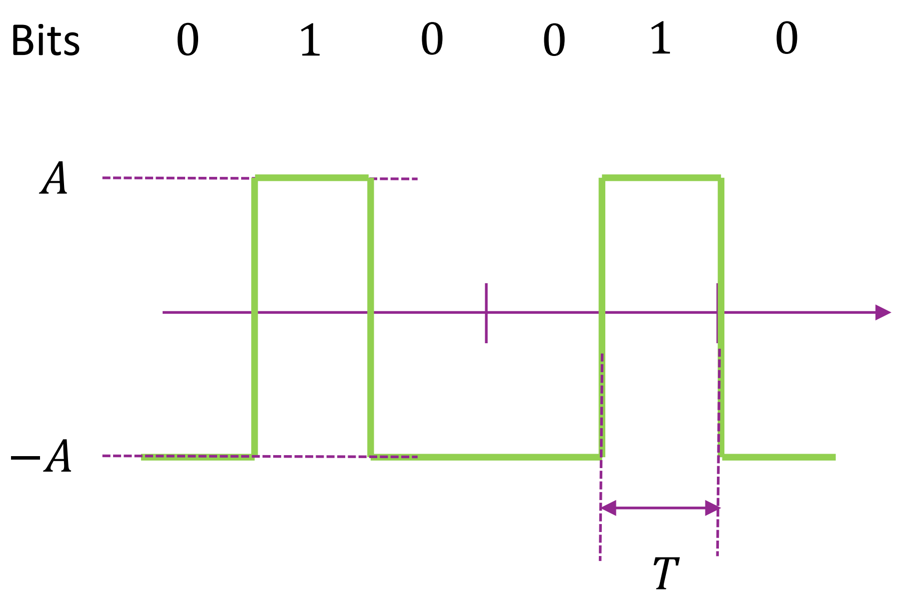
Simple Idea

- How do we transmit digital information over an analog channel?

- Simple idea: At the transmitter
  - Take a sequence of bits $b[k] \in \{0,1\}$ e.g. $010010 \ldots$
  - Divide time into intervals T
  - For $t \in [kT, (k+1)T)$:

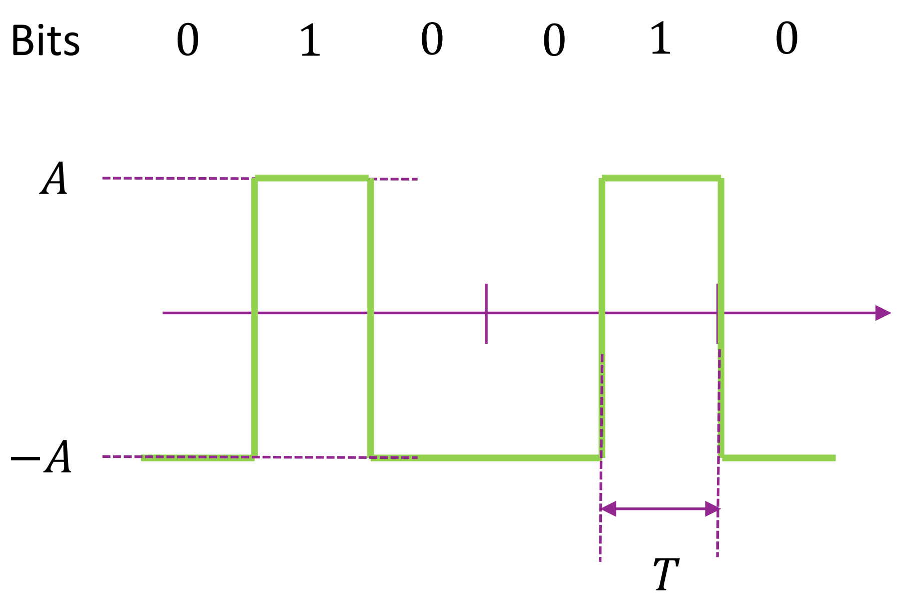
$$u(t) = \begin{cases} A & \text{if } b_k = 1 \\ -A & \text{if } b_k = 0 \end{cases}$$

- At the receiver:
  - Measure $u(t)$ in interval $[kT, (k+1)T)$
  - Determine if $b[k] = 0$ or 1

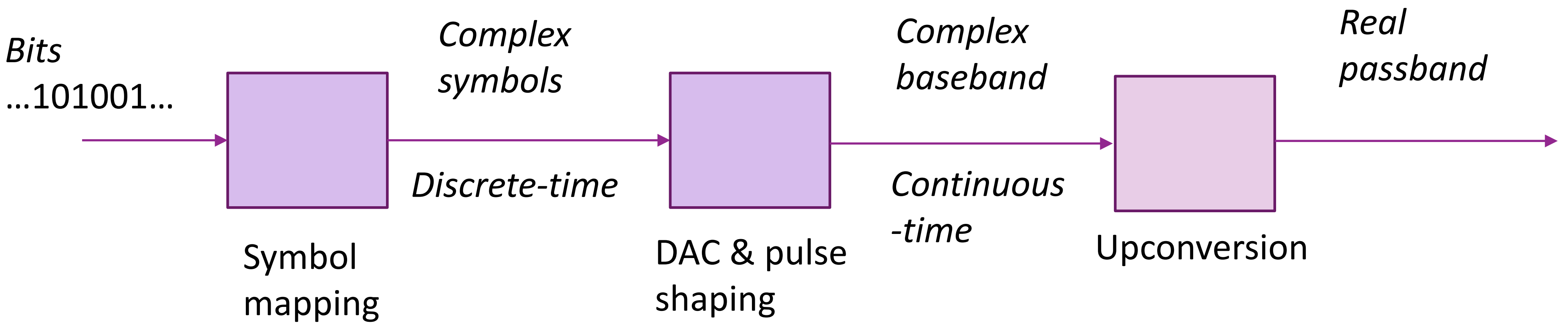


Simple Idea: Continued

- Simple idea exhibits three key steps:
- Step 1. Map bits to symbols:
  - $s[n] = \begin{cases} A & \text{if } b[n] = 1 \\ -A & \text{if } b[n] = 0 \end{cases}$
- Step 2. Modulate to a pulse
  $u(t) = s[n] \quad \text{for } t \in [nT, (n+1)T)$
- Step 3. Upconvert

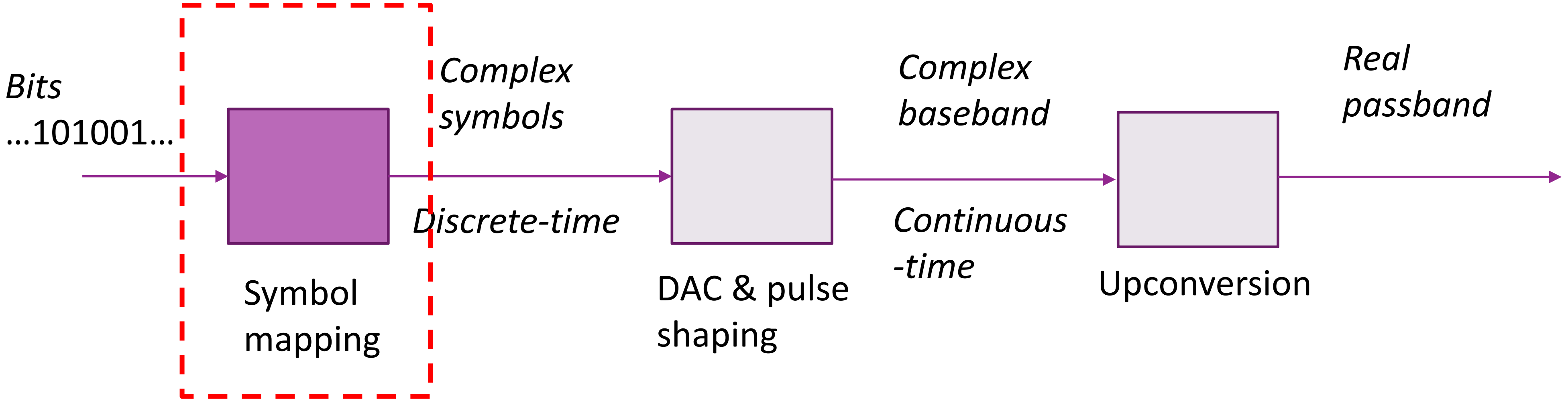


Digital Modulation General Procedure



- Most communication systems follow the same three steps
  - Step 1: Bit to symbol map
  - Step 2: Pulse shaping
  - Step 3: Upconversion (done in last class)

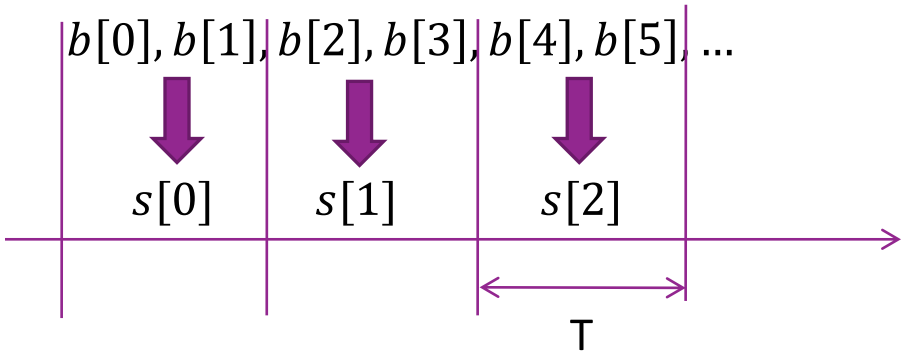
Step 1: Symbol Mapping



- Generally done in three steps:
  - Step 1: Bit to symbol map
  - Step 2: Pulse shaping
  - Step 3: Upconversion (done in last class)

Step 1: Bit to Symbol Mapping

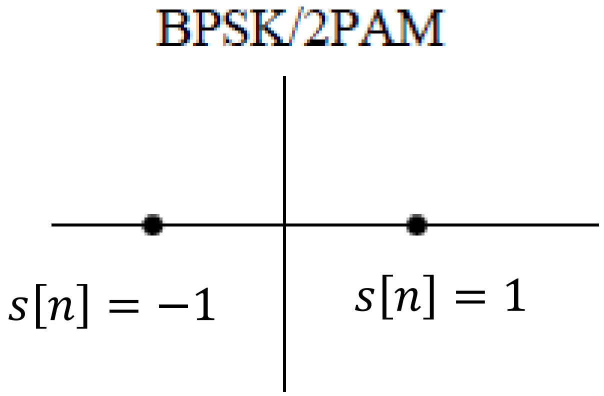
- $b[k] \in \{0,1\}$ = sequence of bits.
- $x[n] \in \{0,1,\ldots,M-1\}$ = sequence of symbol indices
- $s[n] \in \{s_1,\ldots,s_M\}$ = sequence of complex symbols
- Modulation rate: $R_{mod} = \log_2 M$ bits per symbol
- Symbol period: One symbol every T seconds.
- Bit rate of $R = R_{mod}/T$ bits per second



Ex. with M=4 symbols
R_{mod}=2 bits per symbol

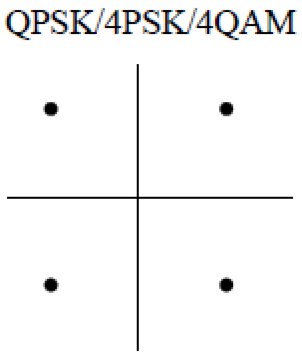
Example: BPSK

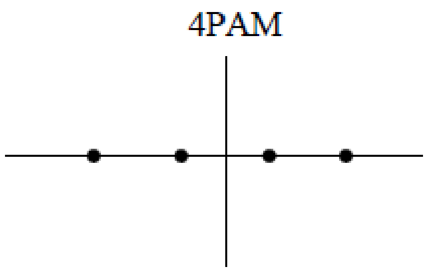
- 1 bit per symbol
- $s[n] = \begin{cases} 1 & x[n] = 1 \\ -1 & x[n] = 0 \end{cases}$
- Symbol is always real



Example 2: 4-PAM and QPSK

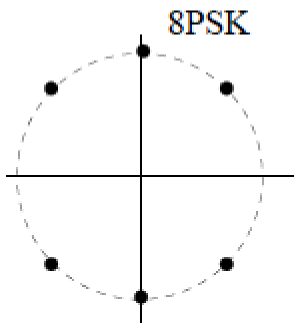
- Two bits per symbol
- 4-PAM: Symbols are multi-level real.
- QPSK: Symbol is complex
  - $s[n] = s_c[n] + js_s[n]$
  - Has I and Q parts
- Draw bit to symbol mapping table on board

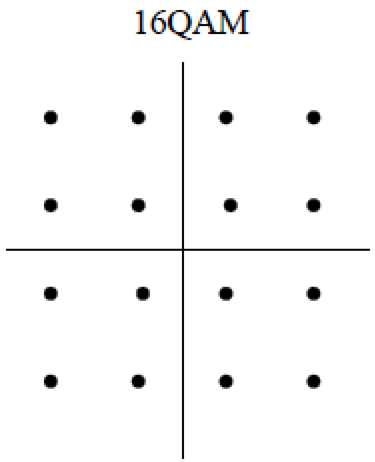




Higher-Order Modulation

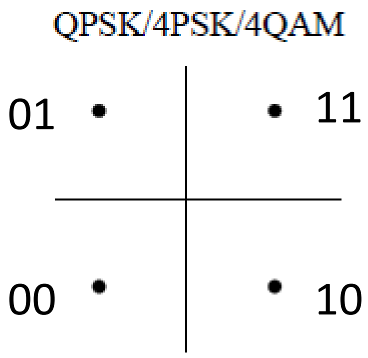
- Constellations go up to 1024 in wireline communications
- Wireless is typically limited to 64-QAM (6 bits per symbol)
- High order modulation:
  - Will see need very low noise to detect high order modulation correctly





Example Problem

- Given bit sequence: $b = (1,0,0,1,1,1, \dots)$
- What are the first 3 symbols under the QPSK mapping
- Suppose the symbol rate is $f_{sym} = 1/T = 20\ Msym/s$.
- What is the data rate?



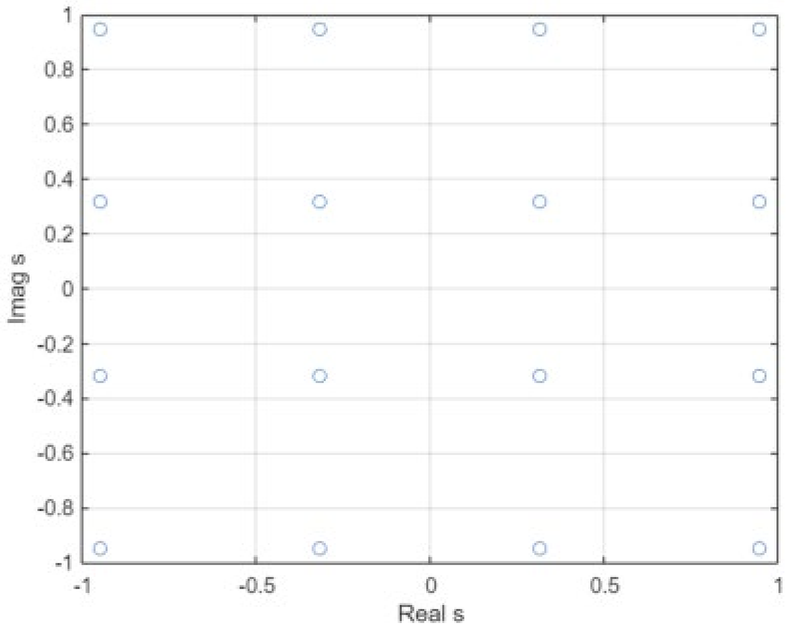
In Class Exercise

Symbol Mapping and TX Filter Design. In-Class Problems

Creating QAM Symbols

The communication toolbox has excellent tools to create QAM symbols. Suppose we want to modulate `nbits`.   Generate the random bits with the `randi` command.

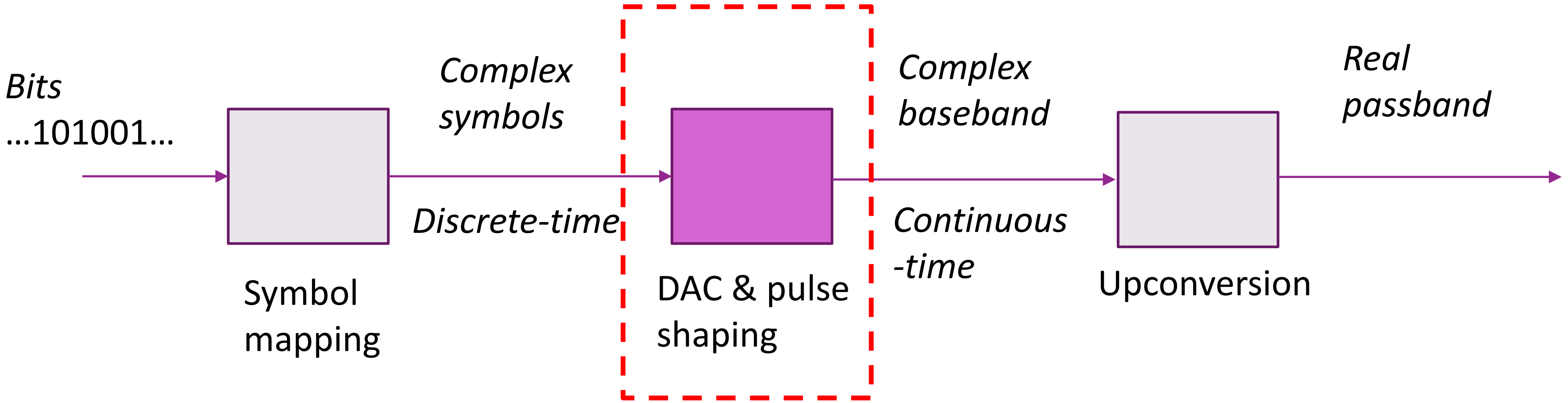
```
nbits = 1024;

% TODO.  Generate the random bits
%    b = randi(...)
b = randi([0,1], nbits, 1);
```



Outline

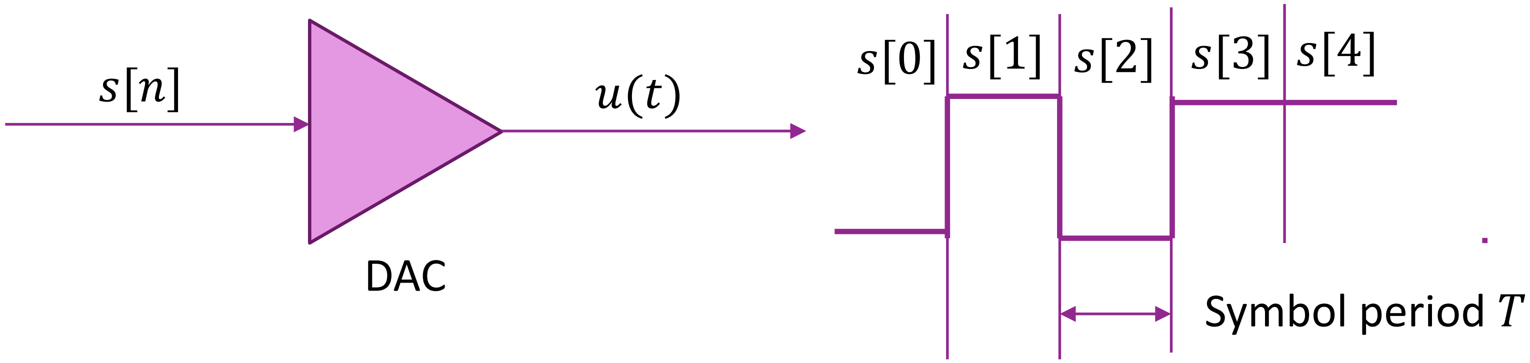
- ❑Symbol mapping
- ❑DAC and pulse shaping
- ❑Fourier analysis and bandwidth of TX filtering
- ❑Power spectral density analysis
- ❑Sinc pulse and Ideal low pass filtering
- ❑Digitally implementing pulse shaping

Step 2: DAC and Pulse Shaping



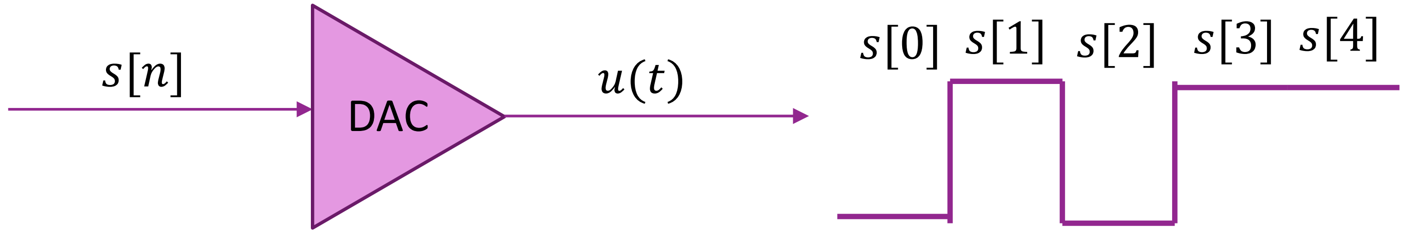
- Generally done in three steps:
  - Step 1: Bit to symbol map
  - Step 2: Pulse shaping
  - Step 3: Upconversion (done in last class)

Digital-Analog Conversion (DAC)



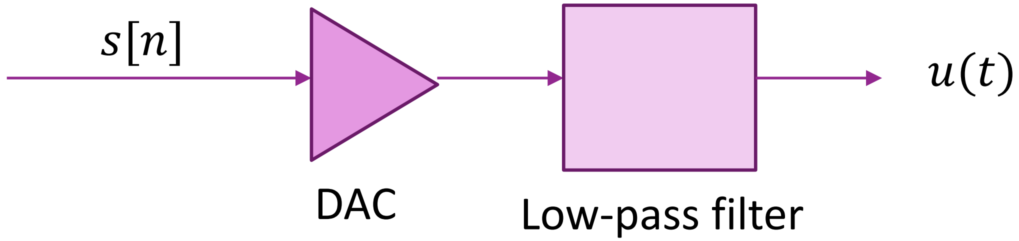
- Simplest idea for generating baseband signal:
- Send $s[n]$ during symbol n: $u(t) = s[n],\ t \in [nT, (n+1)T)$
- Use DAC converter: Sometimes called zero-order-hold
- Symbol rate = $1/T$
- For complex symbols, use two DACs (one for I, one for Q)
  - Then upconvert in analog

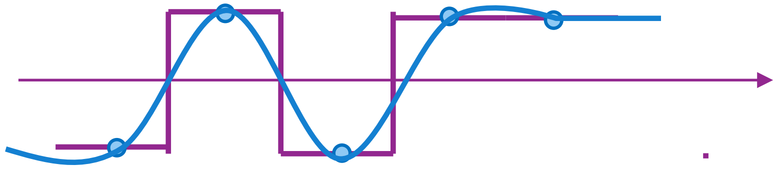
Problem with DAC only solution



- Benefits of using a DAC for modulation
  - Simple to implement
  - Easy to detect symbols at receiver (just sample in middle of symbol period)
  - Used in many examples: e.g. digital signals in circuits. Modulate bits 0,1 to voltages 0, V
- But, problems:
  - Signal $u(t)$ requires high bandwidth due to fast transitions
  - Not acceptable for bandlimited transmissions

DAC + Filtering

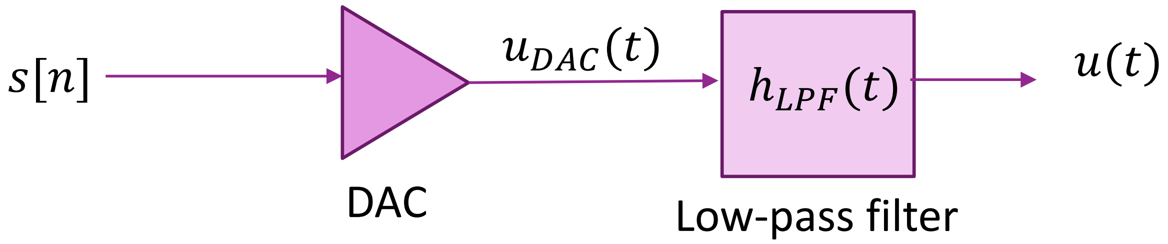




DAC output before filtering
Filtered signal

- Solution: Add low-pass filter to DAC output.
- Removes high frequency components
- Questions:
  - Can we still recover $s[n]$ from signal $u(t)$?
  - How do we measure the bandwidth of the signal
  - What is the effect of the filter on the bandwidth

Infinite Pulse Series Representation



- Can write DAC output as:

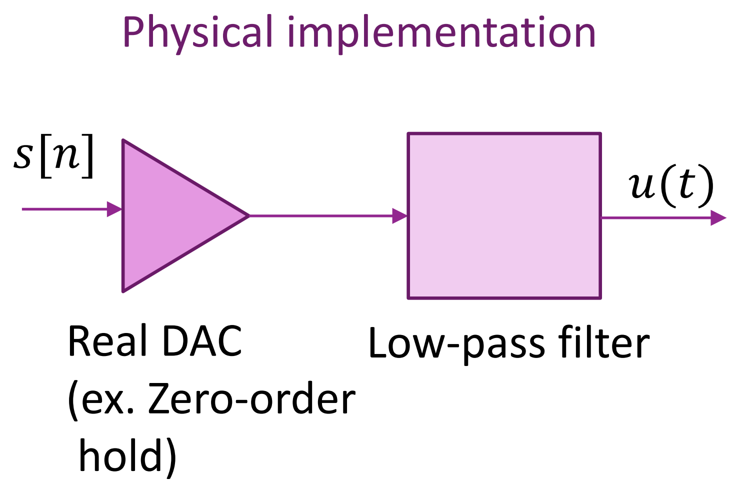
$$u_{DAC}(t) = \sum_{n=-\infty}^{\infty} s[n] h_{DAC}(t - nT), \qquad h_{DAC}(t) = Rect(t/T)$$

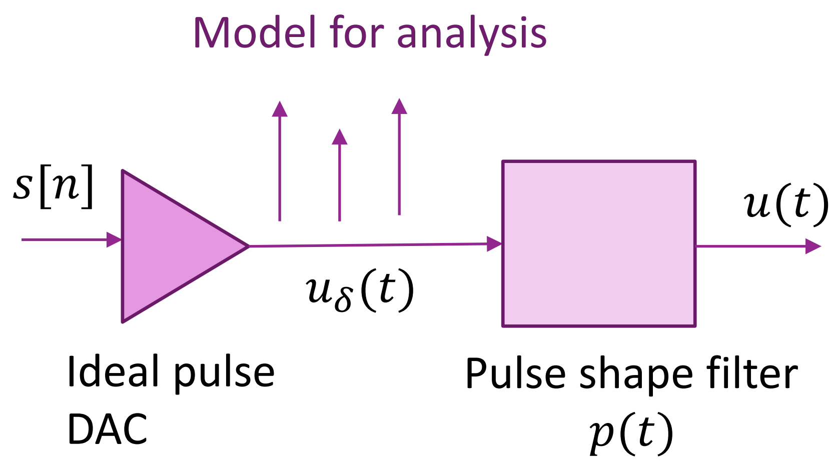
- Then filtered output is:

$$u(t) = h_{LPF}(t) * u_{DAC}(t) = \sum_{n=-\infty}^{\infty} s[n] p(t - nT), \qquad p(t) = h_{DAC}(t) * h_{LPF}(t)$$

- Pulse shape: $p(t) = h_{DAC}(t) * h_{LPF}(t)$

Theoretical Pulse Shape Model





- Can model DAC + LPF via filtered pulse train
- $u_\delta(t) = \sum_n s[n]\delta(t - nT)\,,\quad u(t) = p(t) * u_\delta(t) = \sum_n s[n]p(t - nT)$

Zero ISI Pulse

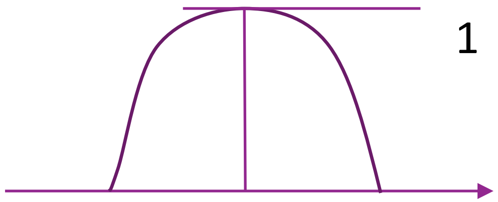
- Consider linear modulation: $u(t) = \sum_n s[n]p(t - nT)$
- Question: Can we recover $s[n]$ from $u(t)$?
- Definition: A pulse $p(t)$ is a zero ISI pulse if $p(0) = 1$ and $p(nT) = 0$ for all $n \neq 0$
  - ISI = intersymbol interference
- If $p(t)$ is a zero ISI then: $s[n] = u(nT)$
- Design idea: Find a zero ISI pulse, then recover symbols $s[n]$ by sampling $u(nT)$

Simple Pulse Shapes

- Pictures on board
- Rectangular pulse: Leads to zero-order-hold
- Triangular pulse: Leads to linear interpolation
- Zero ISI condition

Sample Problem

- Suppose the complex symbols are: $s[n] = (1+j, 1-j, -1+j)$
- Given pulse $p(t)$ as shown to right
- Draw the real and imaginary components of $u(t)$
- Where would you sample $u(t)$ to recover $s[n]$?



Units in Linear Modulation

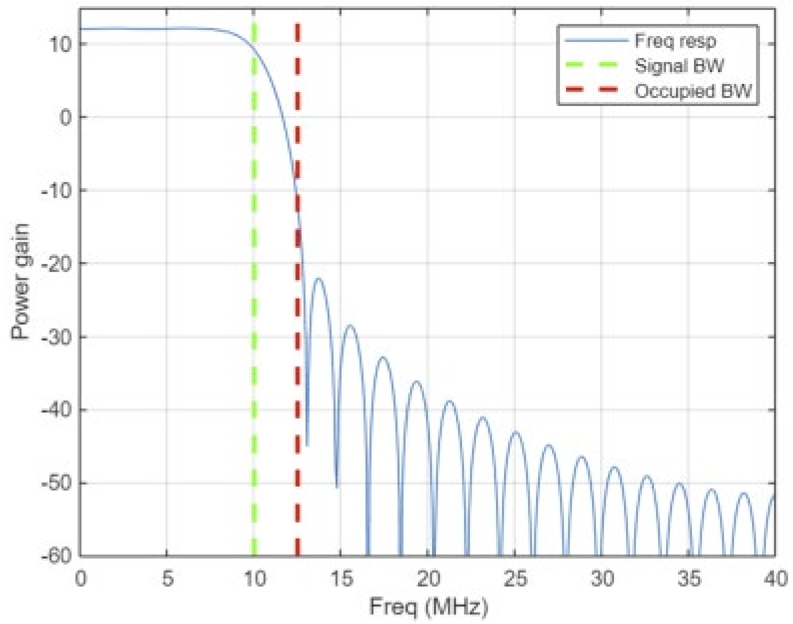
- Suppose $u(t) = \sum_n s[n]p(t - nT)$
- Units for $u(t)$:
  - $|u(t)|^2$ represents instantaneous power
  - So, typically in this class $|u(t)|^2$ in W or mW
  - But $u(t)$ could also be in volts, volts/m (electric field).
  - In these cases, $|u(t)|^2$ is proportional to power.
- Units for $s[n]$ and $p(t)$:
  - Many possible units
  - Convention in this class: $|s[n]|^2$ will have units of energy per sample (e.g., J or mJ/sample)
  - $|p(t)|^2$ will have the units of samples per second (e.g., Hz, MHz, …)
  - Then product $|s[n]|^2|p(t)|^2$ will have units energy/time=power

In Class Exercise

Pulse Shaping with a Raised Cosine Filter

In this section, we will see next perform pulse shaping of the symbols. We will use a widely-used raised cosine filter. Complete the code below to create the filter and plot its impulse response.

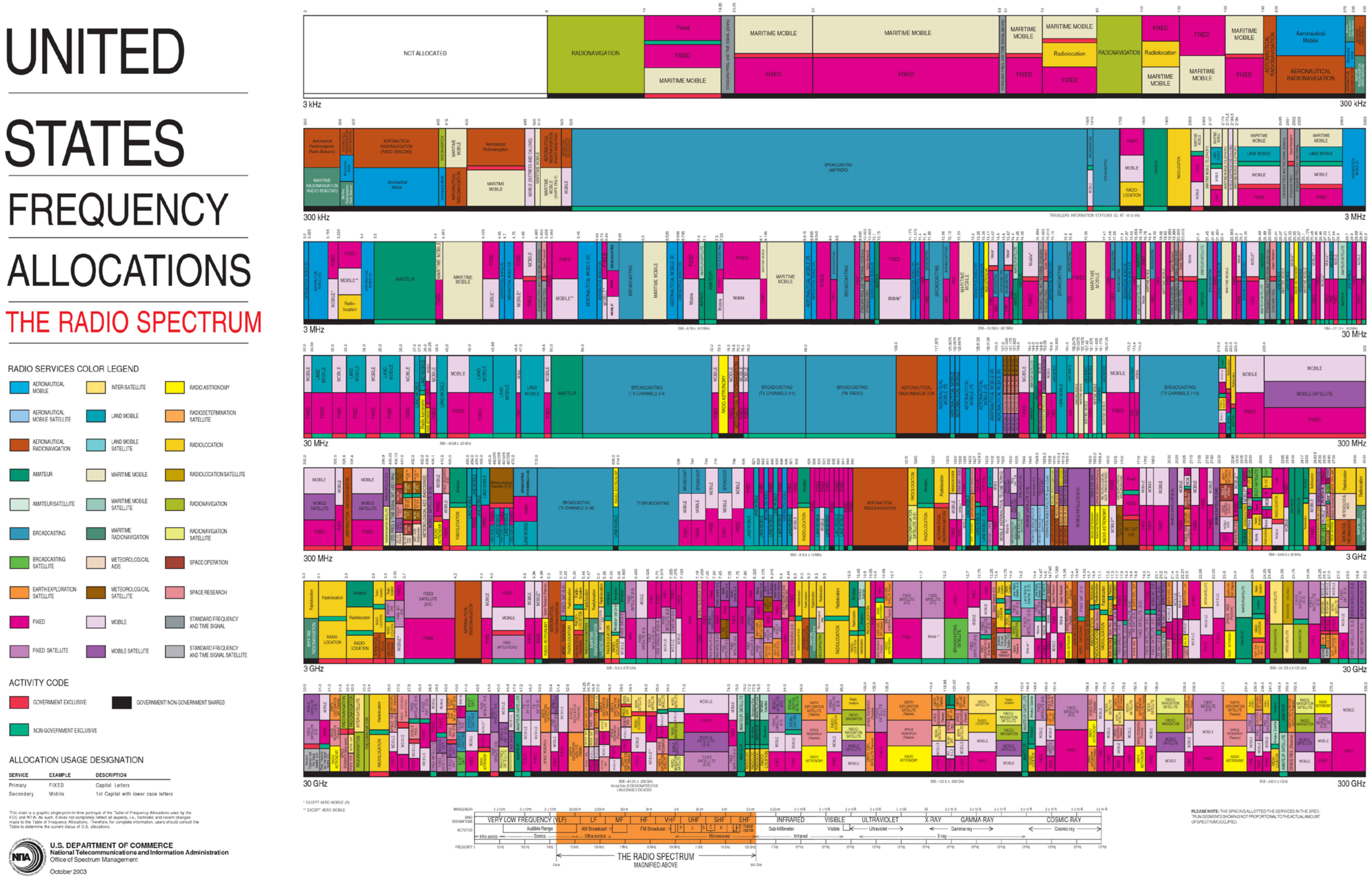
```
fsym = 20e6;  % Symbol rate
beta = 0.25;  % Rolloff factor
span = 10;    % Filter length in symbols
sampsPerSym = 4;   % Number of samples per symbol
b = rcosdesign(beta,span,sampsPerSym);
```



Outline

- Symbol mapping
- DAC and pulse shaping
- Fourier analysis and bandwidth of TX filtering
- Power spectral density analysis
- Sinc pulse and Ideal low pass filtering
- Digitally implementing pulse shaping

Bandwidth and US Spectral Allocations

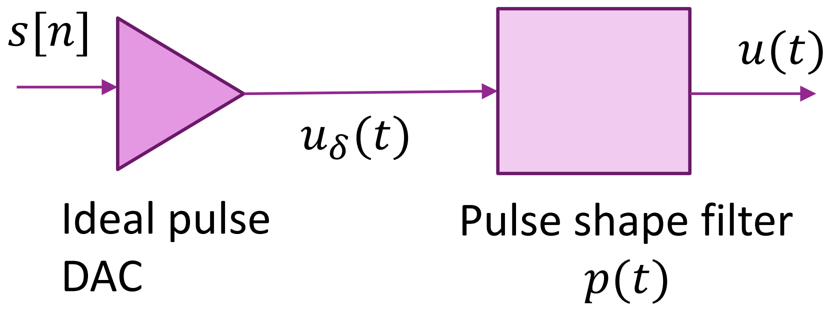


Bandwidth: A Basic Resource

- Limited by:
  - Nature of the medium. Most channels can transmit only limited range of frequencies
  - Ownership / allocations
- We will see that data rate is proportional to bandwidth
  - Assuming power per unit bandwidth is constant
- Basic questions:
  - How do we measure bandwidth?
  - What is the bandwidth of linearly modulated signals?

Fourier Transform of Modulated Signal

- Want to measure occupied bandwidth of $u(t)$
- Look at FT $U(f)$
- Problem: How do we compute FT of $U(f)$?
- Depends on two factors:
  - DTFT of $s[n]$
  - Pulse shape filter response $P(f)$



Review of DTFT

- Given discrete-time sequence $s[n]$
- DTFT: $S(\Omega) = \sum_n s[n] e^{-j\Omega n}$
- Inverse DTFT: $s[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\Omega) e^{j\Omega n} d\Omega$
- Note $S(\Omega)$ is always a 2π periodic signal
- Ω is the discrete frequency. Units is radians per sample.

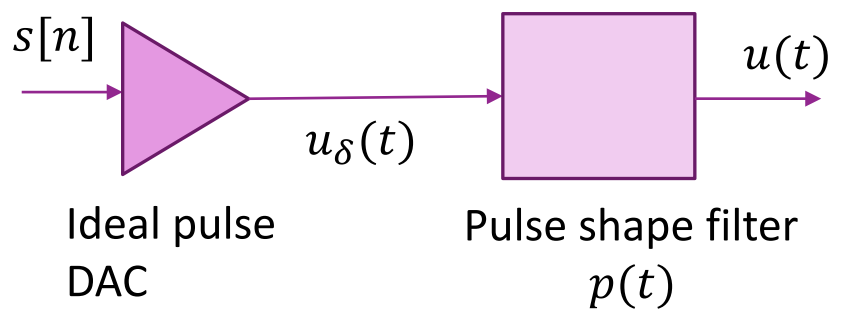
Common DTFT Pairs

Time domain $x[n]$	Frequency domain $X_{2\pi}(\omega)$
$\delta[n]$	$X_{2\pi}(\omega) = 1$
$\delta[n-M]$	$X_{2\pi}(\omega) = e^{-i\omega M}$
$\sum_{m=-\infty}^{\infty} \delta[n-Mm]$	$X_{2\pi}(\omega) = \sum_{m=-\infty}^{\infty} e^{-i\omega Mm} = \frac{2\pi}{M} \sum_{k=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi k}{M}\right)$ $X_o(\omega) = \frac{2\pi}{M} \sum_{k=-(M-1)/2}^{(M-1)/2} \delta\left(\omega - \frac{2\pi k}{M}\right)$ odd M $X_o(\omega) = \frac{2\pi}{M} \sum_{k=-M/2+1}^{M/2} \delta\left(\omega - \frac{2\pi k}{M}\right)$ even M
$u[n]$	$X_{2\pi}(\omega) = \frac{1}{1-e^{-i\omega}} + \pi \sum_{k=-\infty}^{\infty} \delta(\omega - 2\pi k)$ $X_o(\omega) = \frac{1}{1-e^{-i\omega}} + \pi \cdot \delta(\omega)$
$a^n u[n]$	$X_{2\pi}(\omega) = \frac{1}{1-ae^{-i\omega}}$
e^{-ian}	$X_o(\omega) = 2\pi \cdot \delta(\omega + a),\quad -\pi < a < \pi$ $X_{2\pi}(\omega) = 2\pi \sum_{k=-\infty}^{\infty} \delta(\omega + a - 2\pi k)$
$\cos(a \cdot n)$	$X_o(\omega) = \pi\left[\delta(\omega - a) + \delta(\omega + a)\right],$ $X_{2\pi}(\omega) \triangleq \sum_{k=-\infty}^{\infty} X_o(\omega - 2\pi k)$
$\sin(a \cdot n)$	$X_o(\omega) = \frac{\pi}{i}\left[\delta(\omega - a) - \delta(\omega + a)\right]$
$\operatorname{rect}\left[\frac{n - M/2}{M}\right]$	$X_o(\omega) = \frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-\frac{i\omega M}{2}}$
$\operatorname{sinc}(W(n+a))$	$X_o(\omega) = \frac{1}{W} \operatorname{rect}\left(\frac{\omega}{2\pi W}\right) e^{ia\omega}$
$\operatorname{sinc}^2(Wn)$	$X_o(\omega) = \frac{1}{W} \operatorname{tri}\left(\frac{\omega}{2\pi W}\right)$

- See Wikipedia

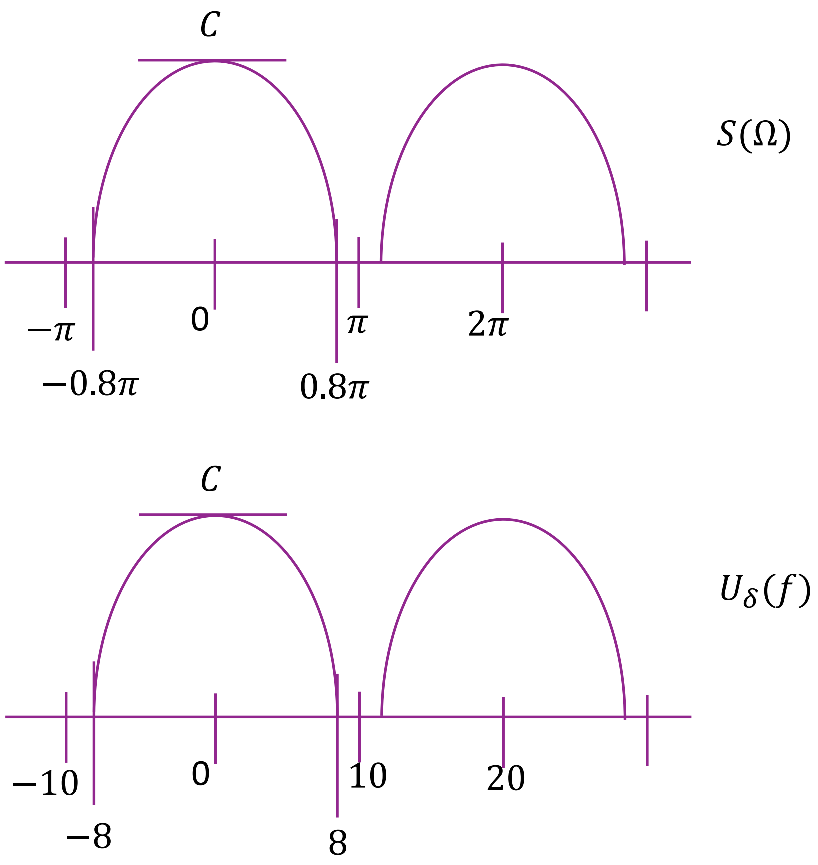
Fourier Analysis of Modulation

- Computing $U(f)$ follows three steps:
- Compute $S(\Omega)$. This is 2π periodic
- Compute $U_\delta(f) = S(2\pi f T) = S\left(\frac{2\pi f}{f_s}\right)$
  - Vertical scale is unchanged
  - Digital frequency Ω mapped to $f = \frac{\Omega}{2\pi T} = \frac{\Omega f_s}{2\pi}$
  - This is periodic with period $\frac{1}{T} = f_s$
- Compute $U(f) = P(f)U_\delta(f)$



Example Problem: Part 1

- Given $S(\Omega)$ as shown
- Suppose $f_s = \frac{1}{T} = 20 \text{ MHz}$
- Draw $U_\delta(f)$
- Solution: $U_\delta(f) = S(2\pi f T)$
  - $U_\delta(f)$ has period $f_s = 20 \text{ MHz}$
  - Same vertical scale as $S(\Omega)$
  - $\Omega = 0.8\pi$ maps to $f = \frac{0.8\pi}{2\pi}(20) = 8 \text{ MHz}$
  - $\Omega = \pi$ maps to $f = \frac{\pi}{2\pi}(20) = 10 \text{ MHz}$

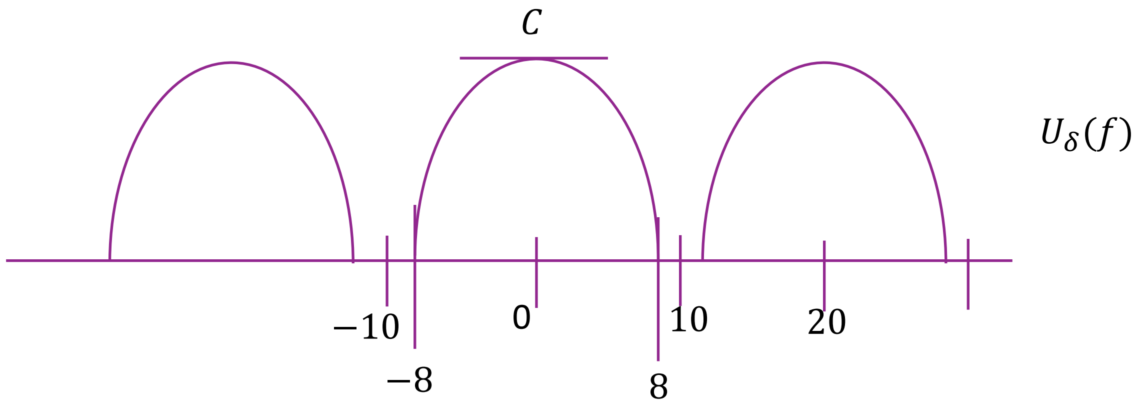


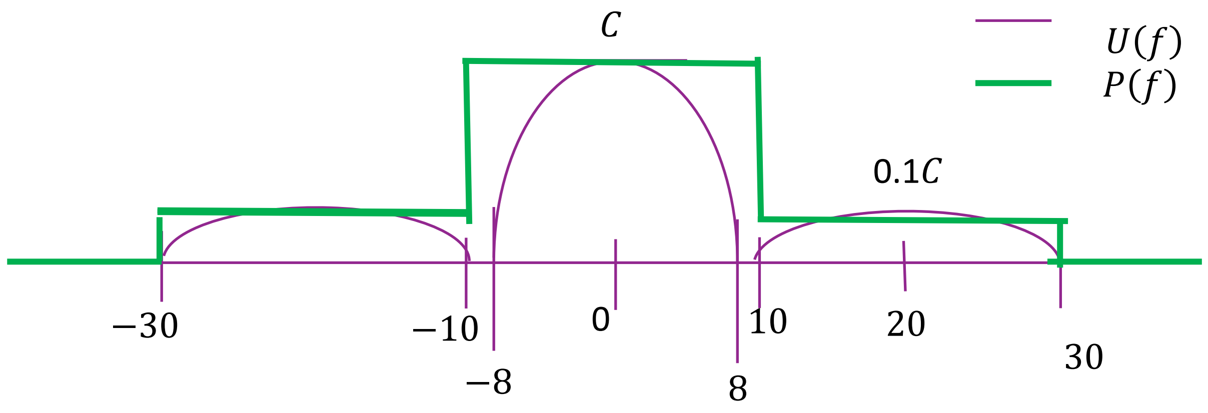
Example Problem: Part 2

- Suppose filter is:

$$P(f) = \begin{cases} 1 & |f| < 10 \\ 0.1 & |f| \in [10,30) \\ 0 & \text{else} \end{cases}$$

- Draw $P(f)$ and $U(f)$
- Solution
  - Use equation to draw $P(f)$
  - Get $U(f)$ from $U(f) = P(f)U_\delta(f)$
  - In this case, filter attenuates two sidelobes



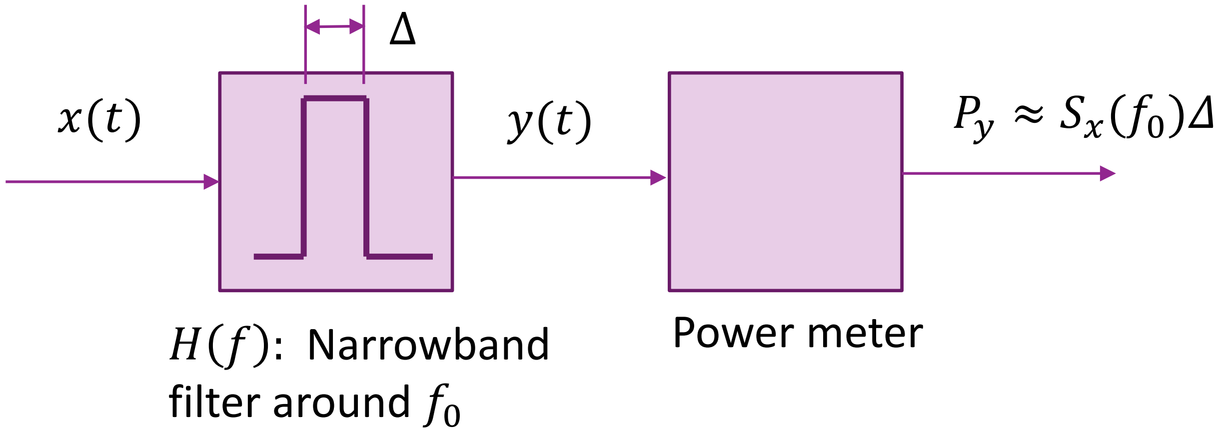


Outline

- Symbol mapping
- DAC and pulse shaping
- Fourier analysis and bandwidth of TX filtering
- Power spectral density analysis
- Sinc pulse and Ideal low pass filtering
- Digitally implementing pulse shaping

Review: PSD of a Continuous-Time Signal

- Let $x(t)$ be a power signal
- Select frequency f_0 to measure PSD
- Filter with narrowband filter
  - $y(t) = h(t) * x(t)$
  - $H(f) = 1$ for $|f - f_0| \leq \Delta/2$
- Measure power P_y



- PSD at f_0 is defined as

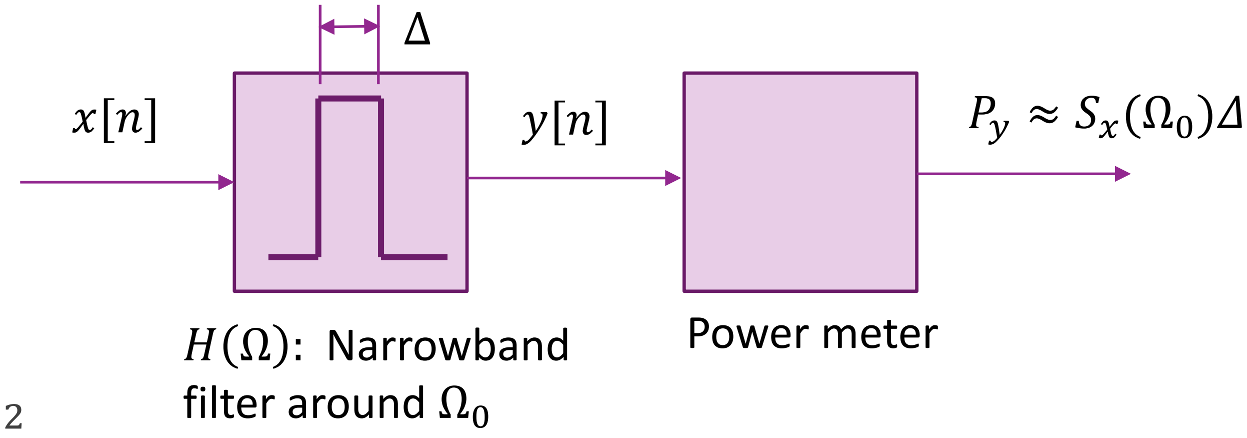
$$S_x(f_0) := \lim_{\Delta \to 0} \frac{1}{\Delta} P_y$$

- Can show this is equivalent to window definition
- Reveals how much power is in a certain frequency

PSD of a Discrete-Time Signal

- Can define PSD of a discrete-time power signal similarly
- Let $x[n]$ be a discrete-time signal
- Select frequency Ω_0 to measure PSD
- Filter with narrowband filter
  - $y[n] = h[n] * x[n]$
  - $H(\Omega) = 1$ for $|\Omega - \Omega_0| \leq \Delta/2$
- Measure power $P_y = \lim_{N} \frac{1}{2N} \sum_{n=-N}^{N} |y[n]|^2$
- PSD at Ω_0 is defined as

$$S_x(\Omega_0) \coloneqq \lim_{\Delta \to 0} \frac{1}{\Delta} P_y$$



Units of Discrete-Time PSD

- Recall, by convention: $|s[n]|^2$ has units of energy (e.g. Joules)
- Power $P = \lim_{N} \frac{1}{2N} \sum_{n=-N}^{N} |s[n]|^2$
  - Units are energy per sample
  - Or, simply energy (e.g . Joules)
- Discrete-time PSD:
  - $S_s(\Omega) = \lim_{\delta} \frac{1}{\delta} \text{Power in freq bin}$
  - Units are energy per sample per radian
- In dB scale: dBJ / radian or dBmJ per radian

Symbol Mean and Energy

- Consider a linear modulated signal:

$$u(t) = \sum_{n=-\infty}^{\infty} s[n]p(t - nT)$$

- What is its PSD?
- Assume $s[n] \in \{s_1, \dots, s_M\}$. M constellation points
- Define symbol mean and symbol energy:

$$\bar{s} = \frac{1}{M}\sum_{m=1}^{M} s_m, \qquad E_s = \frac{1}{M}\sum_{m=1}^{M} |s_m - \bar{s}|^2$$

PSD of a Linear Modulated Signal

- Suppose: Output of ideal DAC is

$$u_\delta(t) = \sum_{n=-\infty}^{\infty} s[n]\delta(t-nT)$$

- After pulse shaping:

$$u(t) = p(t) * u_\delta(t) = \sum_{n=-\infty}^{\infty} s[n]p(t-nT)$$

- Suppose $s[n]$ is a discrete-time power signal with digital PSD $S_s(\Omega)$
- Theorem: PSD of $u_\delta(t)$ and $u(t)$

$$S_{u_\delta}(f) = \frac{1}{T} S_s(2\pi f T), \qquad S_u(f) = \frac{1}{T} S_s(2\pi f T)|P(f)|^2$$

  - Note that $S_s(2\pi f T)$ is periodic with period $\frac{1}{T}$.

Units of PSD Formula

- From previous slide: $S_u(f) = \frac{1}{T} S_s(2\pi f T)|P(f)|^2$
  - $|p(t)|^2$ has units samples/second or frequency
  - $|P(f)|^2$ has units samples/Hz or samples x time
  - Why? Since $\int |P(f)|^2 df = \int |p(t)|^2 dt$
  - $S(2\pi f T)$ has units energy per sample
- Hence units of

$$S_u(f) = \frac{1}{\text{time}} \times \frac{\text{energy}}{\text{sample}} \times (\text{sample} \times \text{time}) = \text{energy}$$

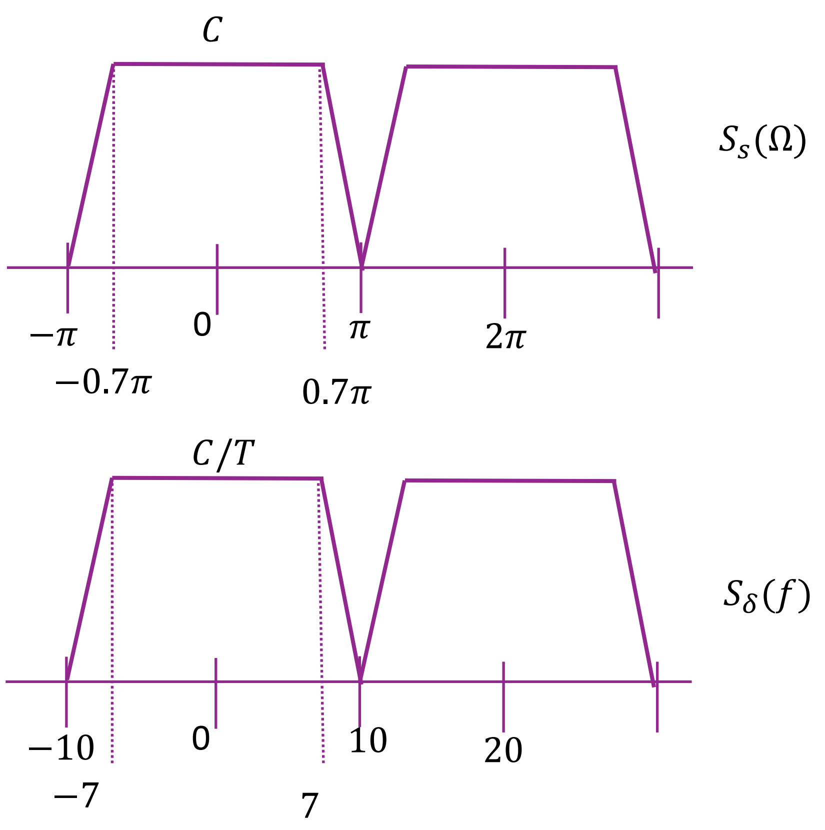
  - This is consistent with our earlier units:
  - Units of $S_u(f)$ is power / Hz = energy

Special Case: IID Symbols

- Suppose: Output of ideal DAC is $u_\delta(t) = \sum_{n=-\infty}^{\infty} s[n]\delta(t-nT)$
- After pulse shaping: $u(t) = p(t) * u_\delta(t) = \sum_{n=-\infty}^{\infty} s[n]p(t-nT)$
- Suppose that:
  - Assume $s[n]$ are uncorrelated and zero mean
  - Average symbol energy: $E_s = E|s[n]|^2$
- Then $S_s(\Omega) = E_s$
- $S_{u_\delta}(f) = \frac{1}{T}E_s,$
- $S_u(f) = \frac{1}{T}E_s|P(f)|^2$
- Power $P_u = \frac{1}{T}E_s\|p\|^2$

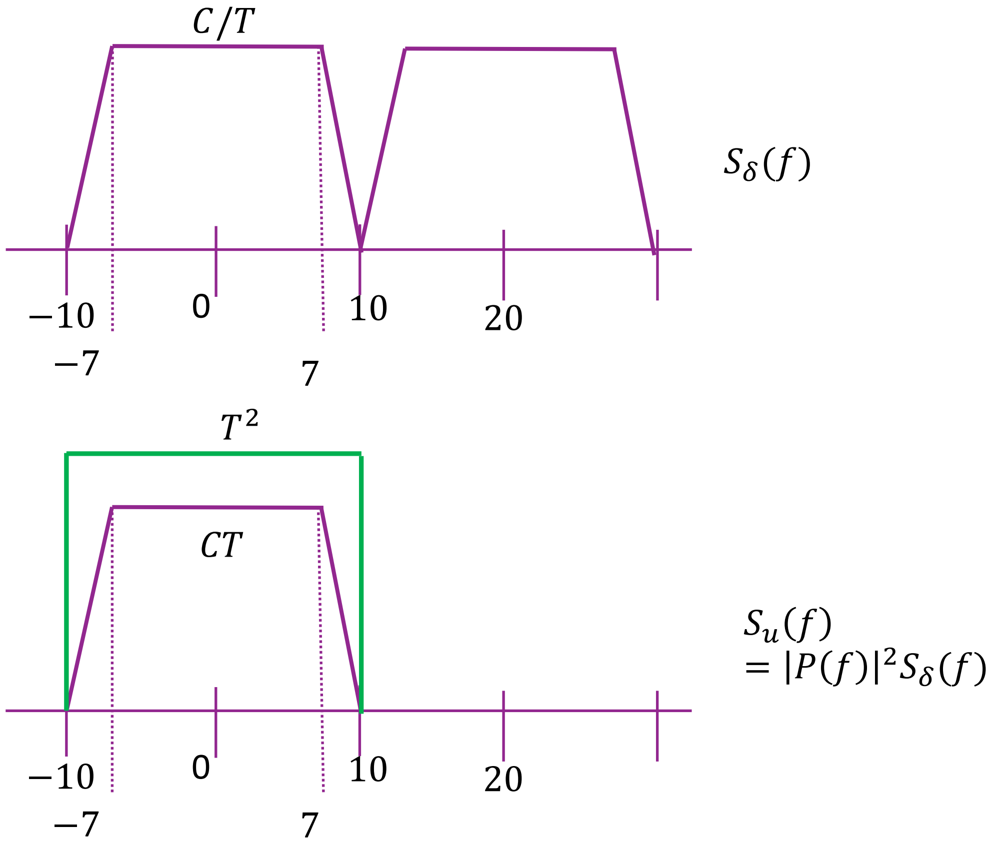
Example Problem: Part 1

- Given PSD of $s[n]$ $S_s(\Omega)$ as shown with $C = 0.1$
- Suppose $f_s = \frac{1}{T} = 20$ MHz
- Draw PSD of $U_\delta(f)$
- Solution: $S_\delta(f) = \frac{1}{T} S(2\pi f T)$
  - $S_\delta(f)$ has period $f_s = 20$ MHz
  - Vertical scaled by $\frac{1}{T}$
  - $\Omega = 0.7\pi$ maps to $f = \frac{0.7\pi}{2\pi}(20) = 7$ MHz
  - $\Omega = \pi$ maps to $f = \frac{\pi}{2\pi}(20) = 10$ MHz



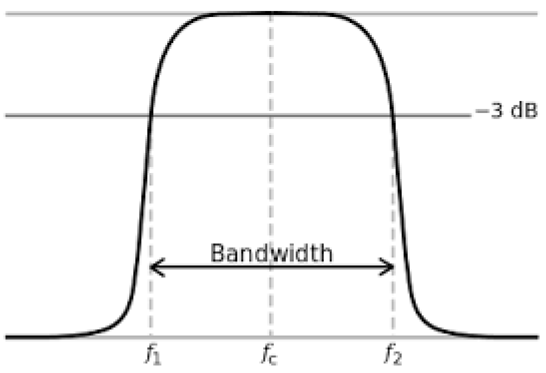
Example Problem: Part 2

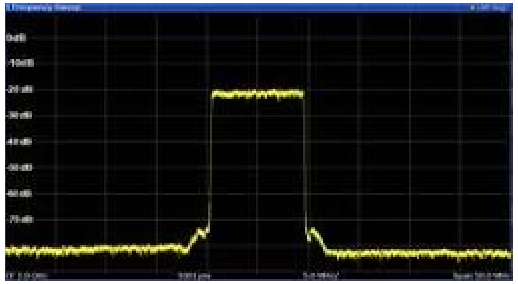
- Now suppose $p(t) = sinc\left(\frac{t}{T}\right)$
- Draw $S_u(f)$
- Solution:
  - $P(f) = TRect(fT)$
  - $|P(f)|^2 = T^2 Rect(fT)$
  - Scales low-pass signal by T^2
  - Removes all sideloble
- Total power in the signal:
  - Area of a trapezoid
  - $P_u = \int S_u(f)df = \frac{CT}{2T}[0.7 + 1] = 0.85C$



Measuring Bandwidth

- PSD of modulated bits is $S_u(f) = \frac{1}{T} E_s |P(f)|^2$
  - Complex baseband signal
  - After upconversion will be shifted to $\pm f_c$
- Definition: Signal is exactly band-limited to $|f| \leq W$
  - if $S_u(f) = 0$ for $|f| \geq W$
- Exact bandwidth = $2W$
- Approximate BW: Typically require $S_u(f) \approx 0$ for $|f| \geq W$
- Different measures of approximate bandwidth
  - 3 dB bandwidth
  - 98% bandwidth, …





Examples

- Recangular pulse:

$$p(t) = \frac{1}{T}\mathrm{I}_{[-\frac{T}{2},\frac{T}{2}]} \Rightarrow |P(f)|^2 = sinc^2(fT)$$

  - 99% bandwidth = 10.1/T, 90% BW = 0.85/T

- Sinusoidal pulse (for $T = 1$):

$$p(t) = \sqrt{2}\sin(\pi t) I_{[0,1]}(t)$$

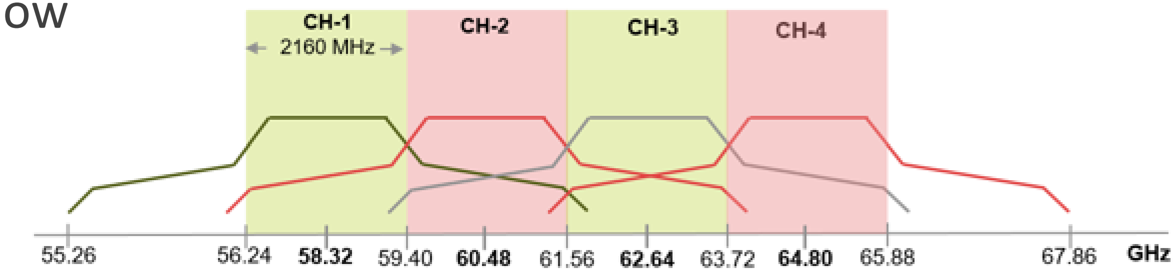
$$|P(f)|^2 = \frac{8}{\pi^2}\frac{\cos^2 \pi f}{(1-4f^2)^2}$$

  - No discontinuities. Less very high frequency components
  - 99% bandwidth = 1.2/T

Spectral Masks

- ❑ Bandwidths for wireless devices are regulated
  - Must transmit most energy in some specified band
  - Ensures no interference between channels
- ❑ Constraints are specified by a spectral mask
  - Represents maximum power level in each band
- ❑ Emissions outside the main band typically very low
  - At least 20 to 40 dB below main lobe

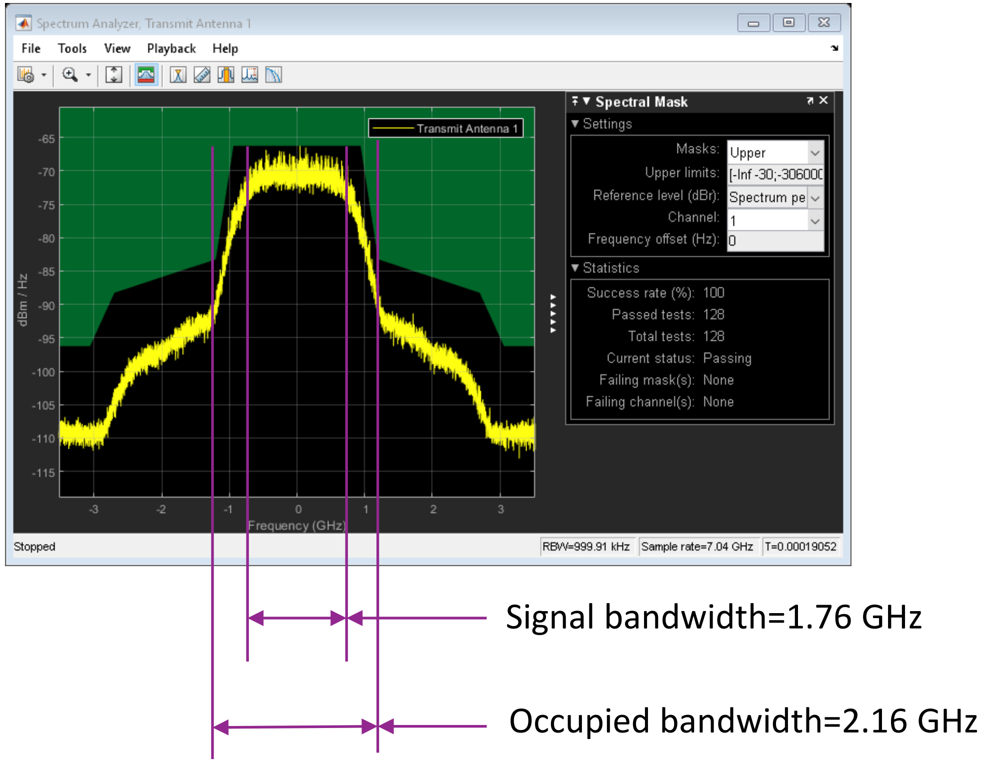
Channels for 802.11ad
Each channel is 2.16 GHz



Signal Bandwidth and Excess Bandwidth

- Usually, signal of interest is contained in smaller band
  - Signal bandwidth < occupied bandwidth
- Excess bandwidth = Occupied – Signal bandwidth
  - Allows a transition region
  - Filters cannot roll off infinitely fast
- 802.11ad example:
  - Sample rate typically 1.76 Gsamp/s
- Lower frequencies, excess bandwidth is even smaller
  - Ex. LTE 20 MHz channel
  - Signal bandwidth = 18 MHz
  - Excess bandwidth $\approx 10\%$

Spectral mask for 802.11ad



Excess bandwidth=22%

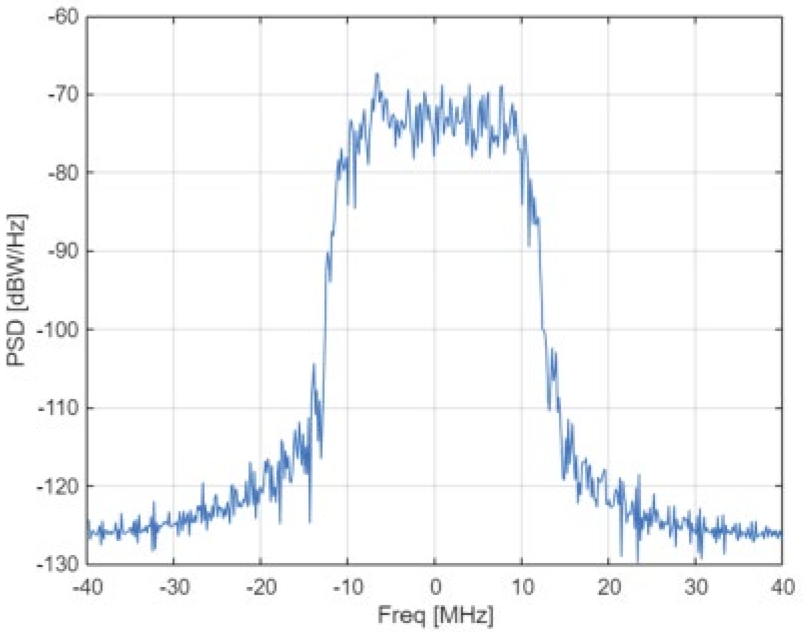
In Class Exercise

Measuring the Power Spectral Density

MATLAB has an excellent tool for measuring the PSD, `pwelch`, based on the Welch algorithm. It properly normalizes the PSD estmate based on the sampling rate. It can be called with the following syntax.

```
window = hamming(512);  % Averaging window
[Px,fx] = pwelch(x,window,[],[],fsamp,'centered');
```

The above function returns the PSD `Px` in linear scale and frequency `fx`. Plot the PSD in dBW/Hz vs. frequency in MHz. It should match the pulse shape frequency response. Label the axes.



Outline

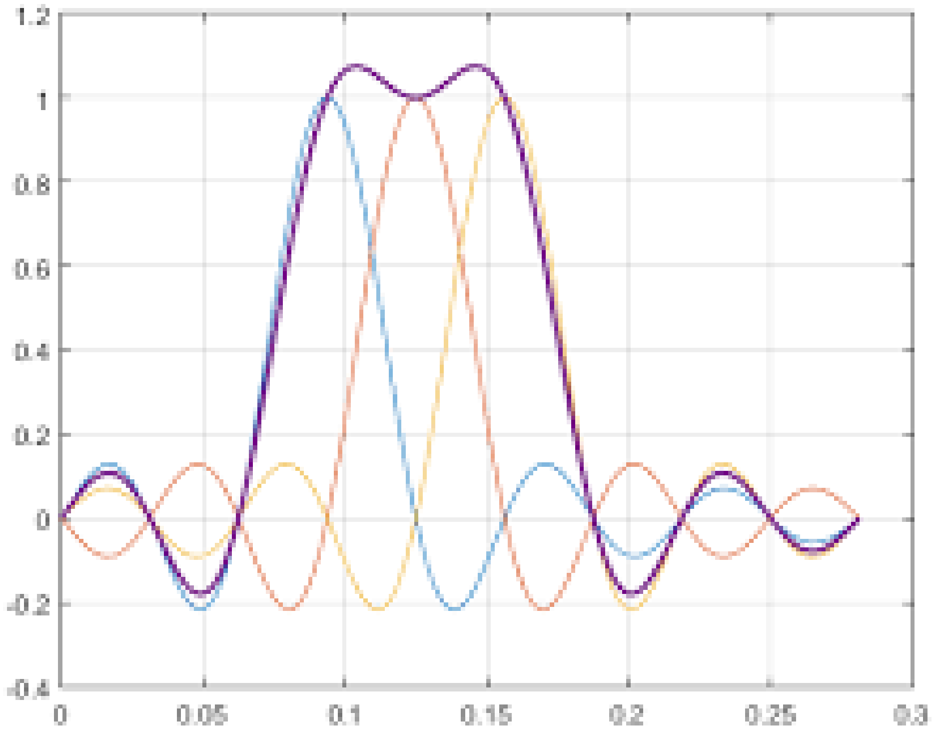
- Symbol mapping
- DAC and pulse shaping
- Fourier analysis and bandwidth of TX filtering
- Power spectral density analysis
- Sinc pulse and Ideal low pass filtering
- Digitally implementing pulse shaping

Design Goals

- Want to design pulse with two goals
- Goal 1. Bandwidth limits:
  - Most systems (esp. RF) impose bandwidth limits on transmissions.
  - PSD of modulated bits is $S_u(f) = \frac{1}{T} E_s |P(f)|^2$
  - Want $|P(f)|^2 \approx 0$ for $|f| \geq W$ where W is (single-sided) bandwidth limit
- Goal 2: Recover symbols $s[n]$ from $u(t)$
  - Sufficient condition: Use zero ISI pulse
  - Then recover with correct sampling
- Can we find a pulse shape satisfying both goals?

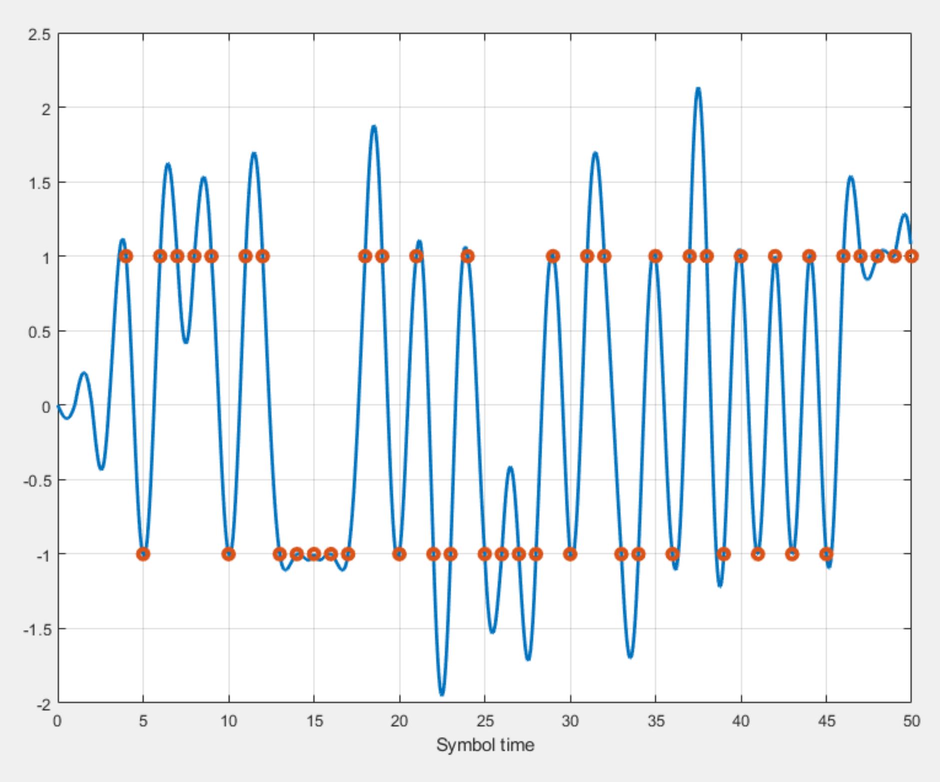
Sinc Pulse

- Use sinc pulse $p(t) = sinc(t/T)$
- Satisfies zero ISI condition:
  - $p(nT) = 0$ for $n \neq 0$
- Pulse shape frequency response:
  $$P(f) = TRect(fT)$$
  - $P(f) = 0$ for $|f| > 1/2T$
- Two-sided bandwidth is $= 1/T$
- Conclusion: sinc pulse satisfies two goals
  - If BW limit > $1/T$



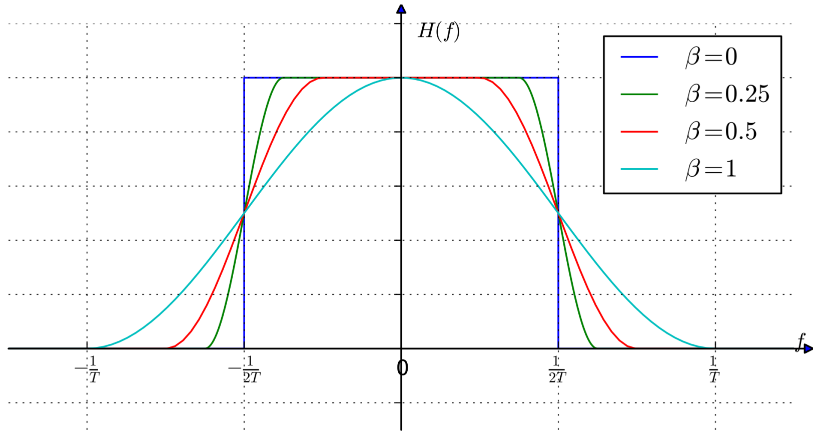
Sinc Pulse Shaping Illustrated

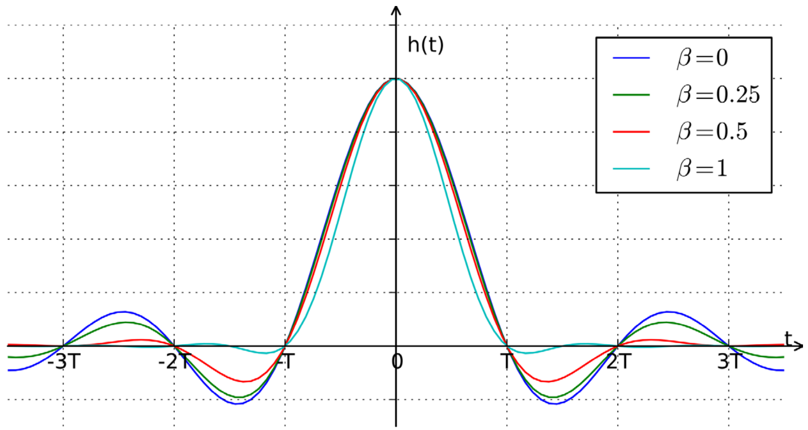
- BPSK symbols
- Sinc pulse interpolates the symbols exactly
- No out of band emissions
- But:
  - Waveform varies rapidly between samples
  - Synchronization offsets will cause errors
  - High peak-to-average ratio
  - Needs an infinite length to implement



Cosine Filtering

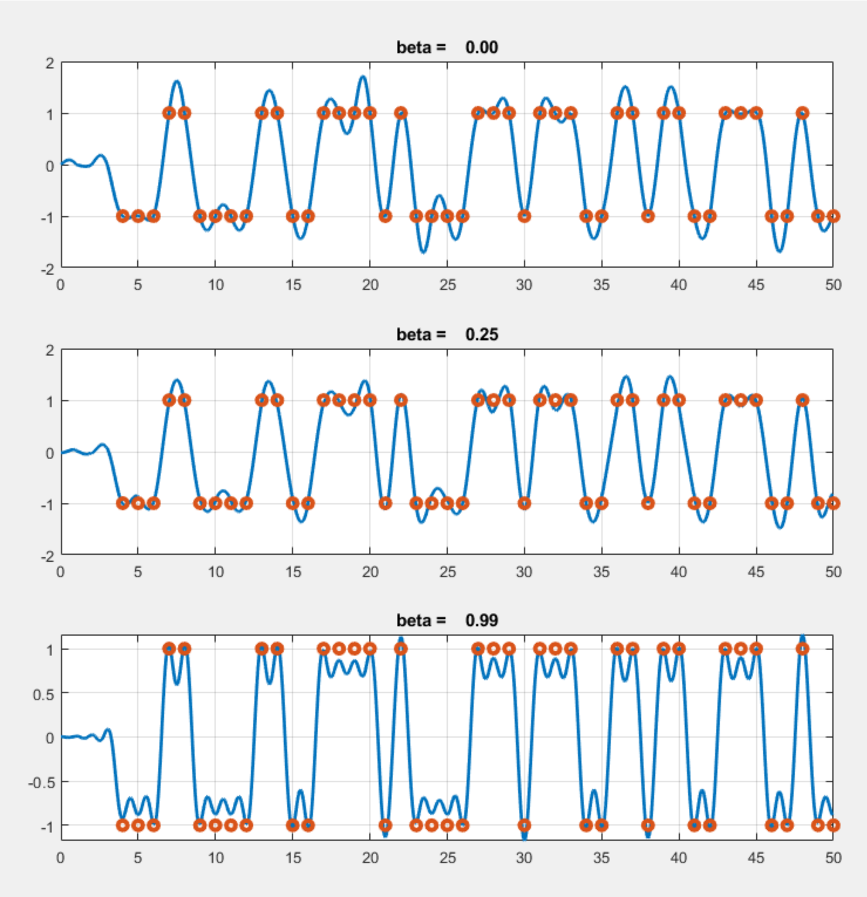
- Set of filters parametrized by β
  - $\beta \in [0,1]$ is called the rolloff
- Excess bandwidth percentage β
- $\beta = 0 \Rightarrow$ Ideal sinc filter
  - No excess bandwidth.
- $\beta > 0$
  - Creates excess bandwidth
  - But, allows shorter filter





Cosine Filtering Illustrated

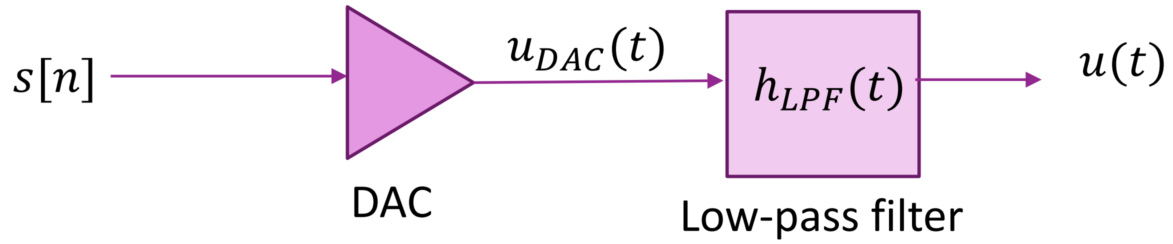
- Plotted to the right:
  - BPSK symbols filtered with raised cosine filters
- Higher values of β
  - Symbol transitions are faster
  - More out-of-band emissions
  - But, less peak-to-average
  - Less variations between symbols



Outline

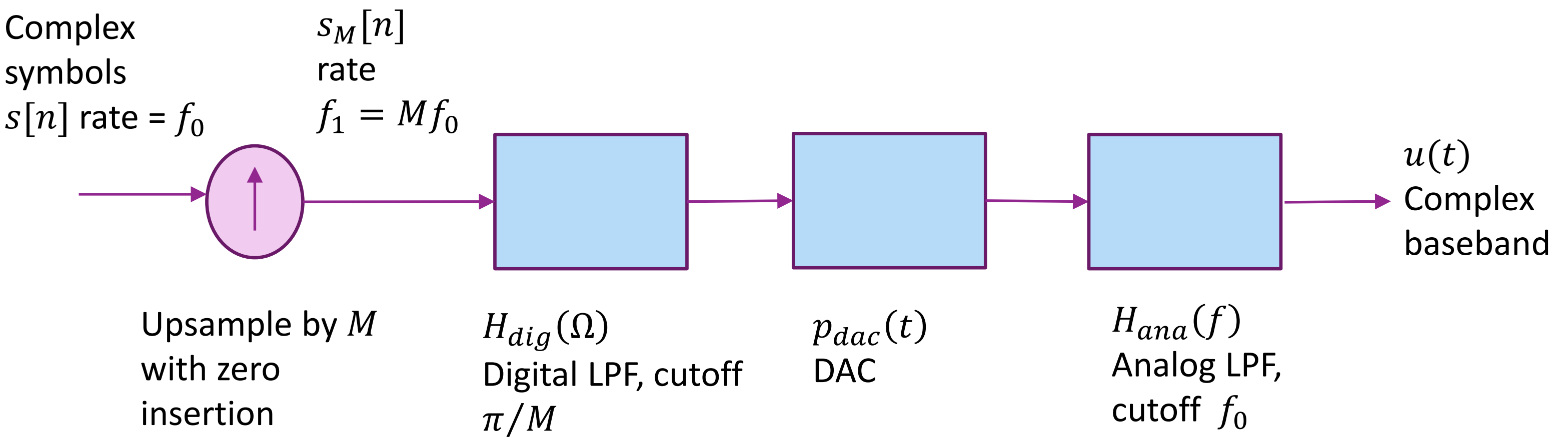
- Symbol mapping
- DAC and pulse shaping
- Fourier analysis and bandwidth of TX filtering
- Power spectral density analysis
- Sinc pulse and Ideal low pass filtering
- Digitally implementing pulse shaping

Problems with Analog LPF solution



- Up to now, we have assumed simple two stage linear modulation
  - DAC followed by LPF
- Challenges: LPF must be implemented in analog.
  - Want LPF filter to approximate ideal Rectangular response
  - Difficult to implement in analog
  - Analog filters typically have limited roll-off

Practical Pulse Shaping Block Diagram



- Practical pulse shaping:
  - Combination of analog and digital filtering

Practical Pulse Shaping

- Start with symbols $s[n]$ at f_0
- Upsample by M with zero insertion
  - $s_M[k] = \begin{cases} s[n] & k = Mn \\ 0 & k \neq Mn \end{cases}$
- Digitally filter with $H_{dig}(\Omega)$
- Pulse shape with DAC $p_{dac}(t)$
- Analog filter $H_{ana}(f)$

Frequency Domain Analysis 1

- $S(\Omega)$ = DTFT of $s[n]$ at symbol rate f_0
- Step 1: Upsample with zero insertion:

$$s_M[k] = \begin{cases} s[n] & k = Mn \\ 0 & k \neq Kn \end{cases} \quad S_M(\Omega) = S(M\Omega)$$

  - Upsampled signal has symbol rate $f_{s1} = Mf_{s0}$
- Step 2: Digital filter with DTFT $H_{dig}(\Omega)$

$$x[k] = h_{dig}[k] * s_M[k] \Rightarrow X(\Omega) = H_{dig}(\Omega)S_M(\Omega)$$

  - Design filter to have cutoff at $\Omega = \pi/M$
  - Theoretically, can use infinite sinc
  - But, in practice use long FIR filter

Frequency Domain Interpretation 2

- Step 3: DAC and analog filtering
  - Create an impulse train

$$x_\delta(t) = \sum_k x[k]\delta(t - n\,T/M) \Rightarrow X_\delta(f) = X\left(\frac{2\pi f T}{M}\right)$$

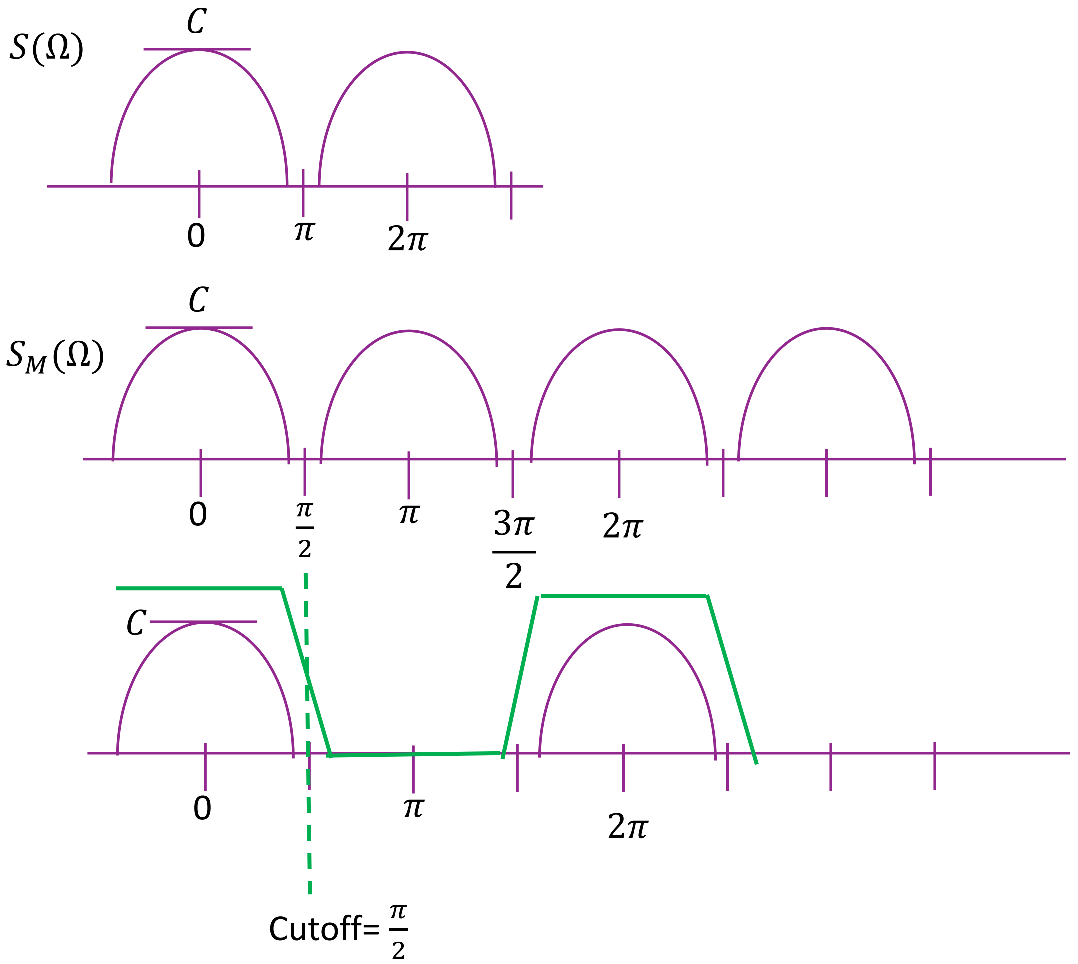
  - Repeated images once every $M/T = f_1 = M f_0$
  - Then,

$$U(f) = X_\delta(f) P_{dac}(f) H_{ana}(f)$$

  - Cut-off frequency of $H_{ana}(f)$ at f_0
  - Removes images $f_1, 2f_1, \ldots$

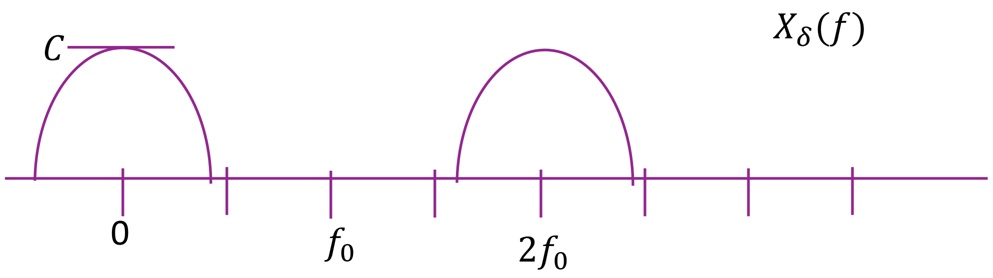
Images 1

- Complex symbols
- Upsampling w/ zero insertion ($M = 2$ shown)
- Digital filtering

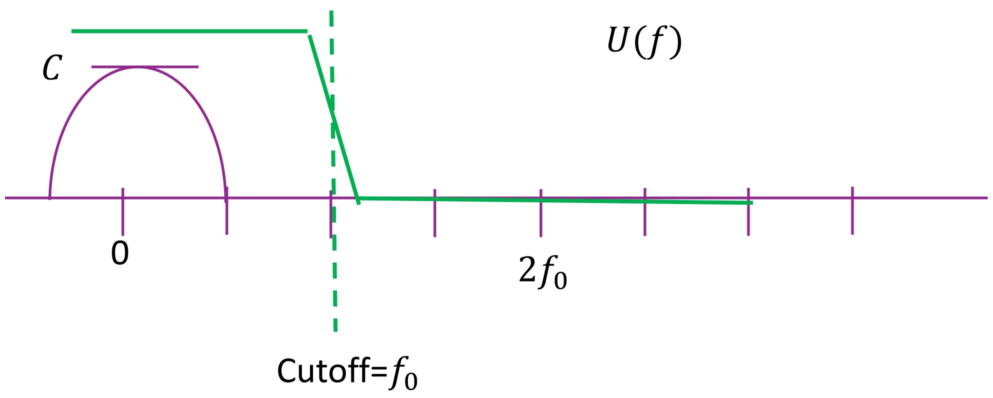


Images 2

- Pulse train



- DAC and analog filtering



Power Spectral Density

- Suppose symbols $s[n]$ are i.i.d. with

$$E(s[n]) = 0, \qquad E|s[n]|^2 = E_s$$

- Can show PSD of $u(t)$ is:

$$S_u(f) = \frac{E_s}{MT_0}|P(f)|^2$$

  - Effective pulse shape: $P(f) = H_{dig}\left(\frac{2\pi f}{Mf_0}\right)P_{dac}(f)H_{ana}(f)$

Effective Pulse Shape

- Can show that the resulting signal is

$$u(t) = \sum s[n]p(t - nT)$$

- Effective pulse shape is:

$$p(t) = \sum_k h_{dig}[k]g\left(t - \frac{k}{M}T\right)$$

  - $g(t) = h_{ana}(t) * p_{dac}(t)$

Example TX Filtering Circuit

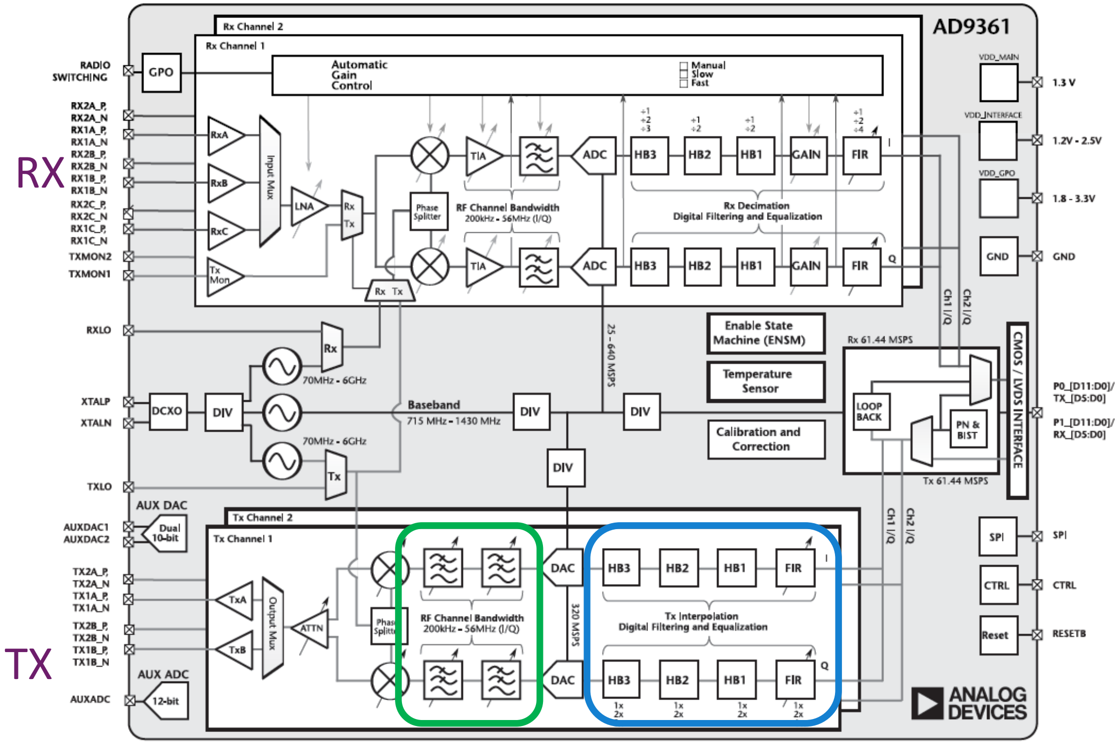


Figure 1.8 Integrated ZIF architecture used in the Pluto SDR.

Analog TX filter

Digital TX filter

- Analog Devices AD9361 Wideband TXCR
- TX filtering performed in two stages:
- Digital filtering:
  - Multiple stages of interpolation
  - Programmable depending on sample rate
- Analog filtering (after DAC)
  - Depends on channel bandwidth