Introduction:

Goal: Transmit a message from one location to another.

When message is...
continuous waveform $\rightarrow$ analog comm (e.g., FM radio),
sequence of numbers $\rightarrow$ digital comm (e.g., mp3 file),
though the sequence of numbers might represent a continuous waveform (as in the case of mp3 audio).

Typical communication media:

twisted pair wire	(e.g., telephone$_A$)
coaxial cable	(e.g., TV$_{A,D}$, data$_D$)
fiber optic cable	(e.g., ethernet$_D$)
EM waves	(e.g., cellular phones$_{A,D}$, WiFi$_D$, TV$_{A,D}$)
water waves	(e.g., underwater network$_{A,D}$)
power lines$_{A,D}$	
compact disc$_D$	
hard drive$_D$	
magnetic tape$_{A,D}$	

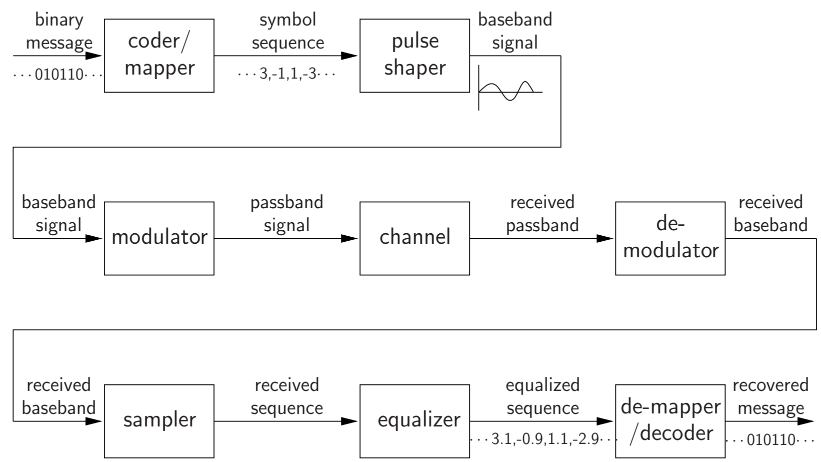
where $_A$ = analog and $_D$ = digital.

Note that, whether the message signal is discrete-time or continuous-time, the transmitted signal is continuous-time!

Analog Communication:

- Perfect recovery is impossible in the presence of noise!

Digital Communication:

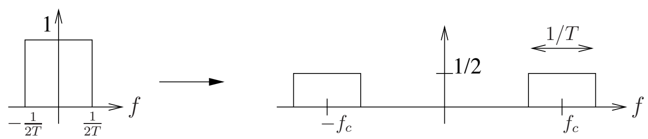


- A digital message is converted to an analog message coding and pulse-shaping, and then transmitted using analog modulation. To recover the message, the received signal is demodulated, sampled, and digitally processed.
- Perfect recovery is possible even in the presence of noise!

Preview of Comm System Components:

Modulator:

- Translates "baseband" analog signal to "passband":



  where f_c is the "carrier frequency."

- There are two principal motivations for doing this:
  1. Often we want to communicate several signals simultaneously (e.g., TV, radio, voice). It's difficult or impossible to do this if they overlap in frequency!
  2. Wireless EM transmission/reception is much easier at higher frequencies, since need antenna length $> \frac{\lambda}{10}$. ($\lambda = \frac{c}{f_c}$ is wavelength and c=3e8 m/s speed of light.)

system	transmission band	$\lambda/10$
VHF (TV)	30–300 MHz	1–0.1 m
UHF (TV)	0.3–3 GHz	10–1 cm
cellular	824–960 MHz	3 cm
WiFi	2.4 GHz	1 cm

  Notice that practical antenna length determines where different signal types can be transmitted.

Coder/Mapper:

- Coder transforms sequence of message bits into an error-resiliant sequence of coded bits.
- Mapper transforms coded bits into discrete "symbols." Ex: If the "symbol alphabet" is $\{-3, -1, 1, 3\}$ and the symbol mapping is

bits	symbol
00	3
01	-1
10	1
11	-3

, then ASCII text would be transmitted via

letter	ASCII code				symbol sequence			
a	01	10	00	01	-1	1	-3	-1
b	01	10	00	10	-1	1	-3	1
c	01	10	00	11	-1	1	-3	3
d	01	10	01	00	-1	1	-1	-3
⋮		⋮				⋮		

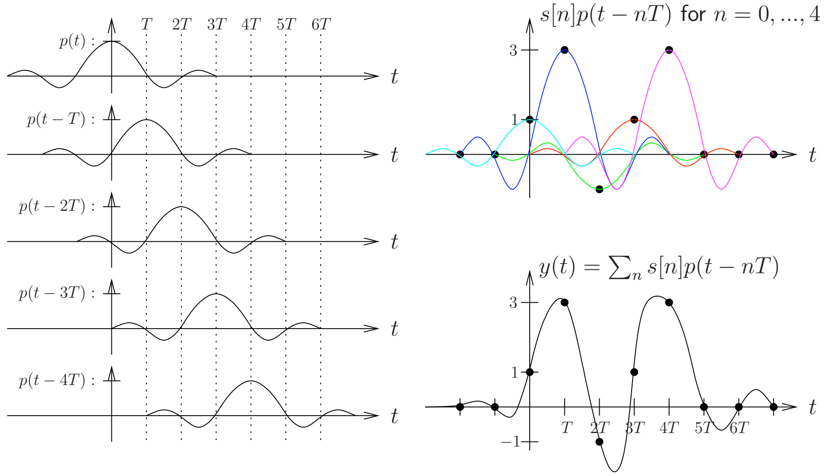
.

Pulse Shaper:

- Converts symbol sequence into a continuous waveform.
- In linear modulation schemes, the time-n symbol $s[n]$ scales a nT-delayed version of pulse $p(t)$:

$$\begin{aligned} y(t) &= \sum_n s[n]p(t-nT) \quad \text{"baseband signal"} \\ T &= \text{"symbol period"} \end{aligned}$$

Ex: Say symbol sequence is $[1, 3, -1, 1, 3]$. Then



Preliminaries (Ch.2):

Fourier Transform (FT):

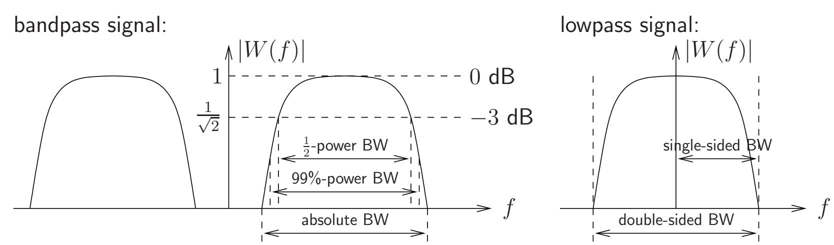
Definition:

$$\begin{aligned} W(f) &= \int_{-\infty}^{\infty} w(t)e^{-j2\pi ft}dt = \mathcal{F}\{w(t)\} \\ w(t) &= \int_{-\infty}^{\infty} W(f)e^{j2\pi ft}df = \mathcal{F}^{-1}\{W(f)\}. \end{aligned}$$

Properties:

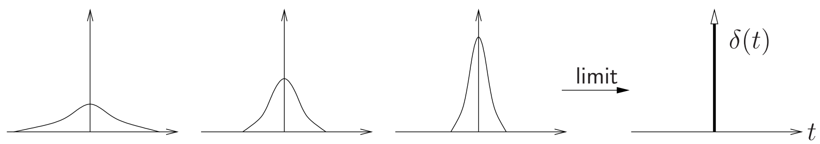
- Linearity: $\mathcal{F}\{c_1w_1(t) + c_2w_2(t)\} = c_1W_1(f) + c_2W_2(f)$.
- Real-valued $w(t) \Rightarrow \begin{cases} \text{conjugate symmetric } W(f) \\ |W(f)| \text{ symmetric around } f = 0. \end{cases}$

"Bandwidth":



Dirac Delta (or "continuous impulse") $\delta(\cdot)$:

An infinitely tall and thin waveform *with unit area*:



that's often used to "kick" a system and see how it responds.

Key properties:

1. Sifting: $\int_{-\infty}^{\infty} w(t)\delta(t-q)dt = w(q)$.
2. Time-domain impulse $\delta(t)$ has a flat spectrum:

$$\mathcal{F}\{\delta(t)\} = \int_{-\infty}^{\infty} \delta(t)e^{-j2\pi ft}dt \ = \ 1 \text{ (for all } f).$$

3. Freq-domain impulse $\delta(f)$ corresponds to a DC waveform:

$$\mathcal{F}^{-1}\{\delta(f)\} = \int_{-\infty}^{\infty} \delta(f)e^{j2\pi ft}df \ = \ 1 \text{ (for all } t).$$

ECE-501 Phil Schniter January 16, 2008

Frequency-Domain Representation of Sinusoids:

Notice from the sifting property that

$$\mathcal{F}^{-1}\{\delta(f-f_o)\} = \int_{-\infty}^{\infty} \delta(f-f_o)e^{j2\pi ft}df = e^{j2\pi f_o t}.$$

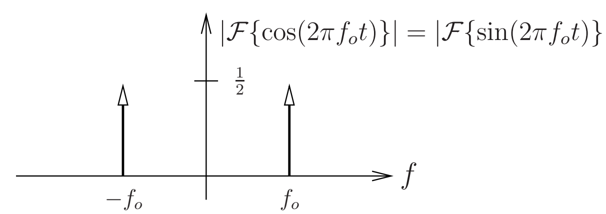
Thus, Euler's equations

$$\begin{aligned}\cos(2\pi f_o t) &= \tfrac{1}{2}e^{j2\pi f_o t} + \tfrac{1}{2}e^{-j2\pi f_o t}\\ \sin(2\pi f_o t) &= \tfrac{1}{2j}e^{j2\pi f_o t} - \tfrac{1}{2j}e^{-j2\pi f_o t}\end{aligned}$$

and the Fourier transform pair $e^{j2\pi f_o t} \leftrightarrow \delta(f-f_o)$ imply that

$$\begin{aligned}\mathcal{F}\{\cos(2\pi f_o t)\} &= \tfrac{1}{2}\delta(f-f_o) + \tfrac{1}{2}\delta(f+f_o)\\ \mathcal{F}\{\sin(2\pi f_o t)\} &= \tfrac{1}{2j}\delta(f-f_o) - \tfrac{1}{2j}\delta(f+f_o).\end{aligned}$$

Often we draw this as



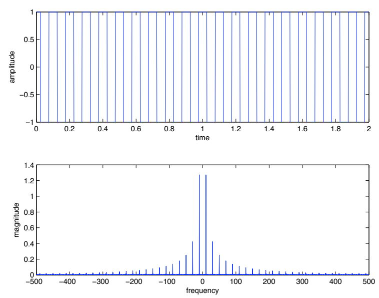
Frequency Domain via MATLAB:

Fourier transform requires evaluation of an integral. What do we do if we can't define/solve the integral?

1. Generate (rate-$\frac{1}{T_s}$) sampled signal in MATLAB.
2. Plot magnitude of Discrete Fourier Transform (DFT) using `plottf.m` (from course webpage).

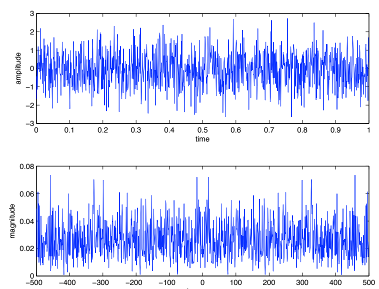
Square-wave example:

```
f = 10;
t_max = 2;
Ts = 1/1000;
t = 0:Ts:t_max;
x = sign(cos(2*pi*f*t));
plottf(x,Ts);
```



Noise-wave example:

```
t_max = 1;
Ts = 1/1000;
x = randn(1,t_max/Ts);
plottf(x,Ts);
```

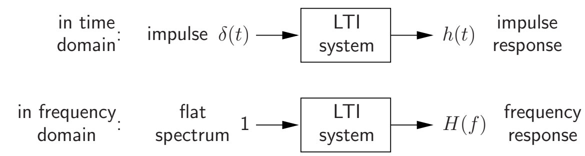


Notice that `plottf.m` only plots frequencies $f \in [-\frac{1}{2T_s}, \frac{1}{2T_s})$.

ECE-501 Phil Schniter January 16, 2008

Linear Time-Invariant (LTI) Systems:

An LTI system can be described by either its "impulse response" $h(t)$ or its "frequency response" $H(f) = \mathcal{F}\{h(t)\}$.



Input/output relationships:

- Time-domain: Convolution with impulse response $h(t)$

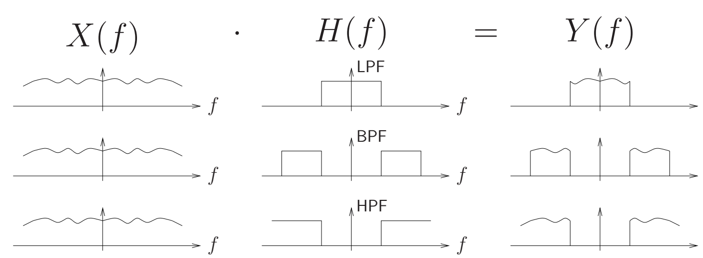
$$x(t) \rightarrow \boxed{h(t)} \rightarrow y(t) \qquad y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)d\tau$$

- Freq-domain: Multiplication with freq response $H(f)$

$$X(f) \rightarrow \boxed{H(f)} \rightarrow Y(f) \qquad Y(f) = H(f)X(f)$$

Linear Filtering:

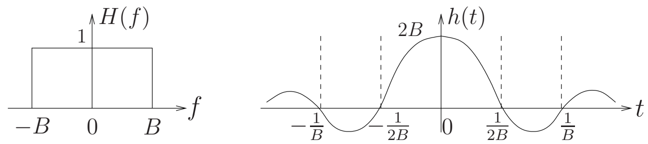
Freq-domain illustration of LPF, BPF, and HPF:



Lowpass Filters:

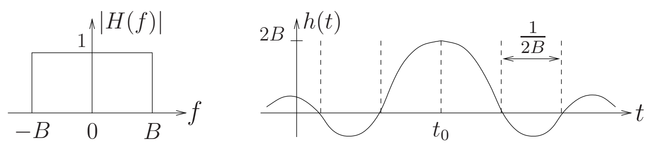
Ideal non-causal LPF (using $\operatorname{sinc}(x) := \frac{\sin(\pi x)}{\pi x}$):

$$H(f) = \begin{cases} 1 & |f| \le B \\ 0 & |f| > B \end{cases} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad h(t) = 2B\,\operatorname{sinc}(2Bt)$$

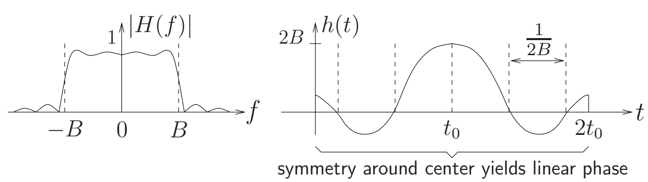


Ideal LPF with group-delay t_o:

$$H(f) = \begin{cases} e^{-j2\pi f t_0} & |f| \le B \\ 0 & |f| > B \end{cases} \quad \stackrel{\mathcal{F}}{\longleftrightarrow} \quad h(t) = 2B\,\operatorname{sinc}\big(2B(t-t_0)\big)$$



A causal linear-phase LPF with group-delay t_o:



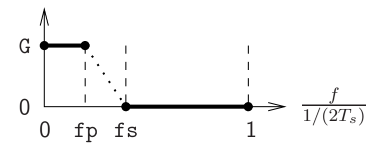
but MATLAB can give better causal linear-phase LPFs. . .

ECE-501 Phil Schniter January 16, 2008

In MATLAB, generate $\frac{1}{T_s}$-sampled LPF impulse response via

```
h = firls(Lf, [0,fp,fs,1], [G,G,0,0])/Ts;
```

where. . .



Lf+1 = impulse response length

{0, fp}, {fs, 1} = *normalized* freq pairs

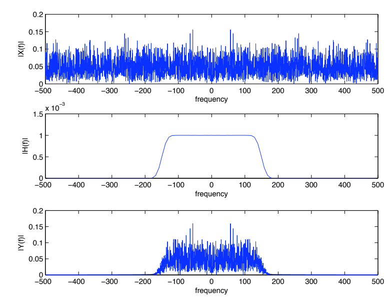
{G, G}, {0, 0} = corresp. magnitude pairs

The commands firpm and fir2 have the same interface, but yield slightly different results (often worse for our apps).

In MATLAB, perform filtering on $\frac{1}{T_s}$-sampled signal x via

```
y = Ts*filter(h,1,x);  or  y = Ts*conv(h,x);
```

```
t_max = 3; Ts = 1/1000;
x = randn(1,t_max/Ts);
h = firls(100,[0,0.2,0.4,1],[1,1,0,0])/Ts;
y = Ts*filter(h,1,x);
subplot(3,1,1);
 plottf(x,Ts,'f');
 ylabel('|X(f)|')
subplot(3,1,2);
 plottf(h,Ts,'f');
 ylabel('|H(f)|')
subplot(3,1,3);
 plottf(y,Ts,'f');
 ylabel('|Y(f)|')
```



Important: The routines firls,firpm,fir2 generate *causal* linear-phase filters with group delay $= \frac{\texttt{Lf}}{2}$ samples. Thus, the filtered output y will be delayed by $\frac{\texttt{Lf}}{2}$ samples relative to x.