Lecture 24 (Multirate DSP)

Last class:

- Main types of IIR filters (B, C1, C2, E)

FIR vs IIR ?

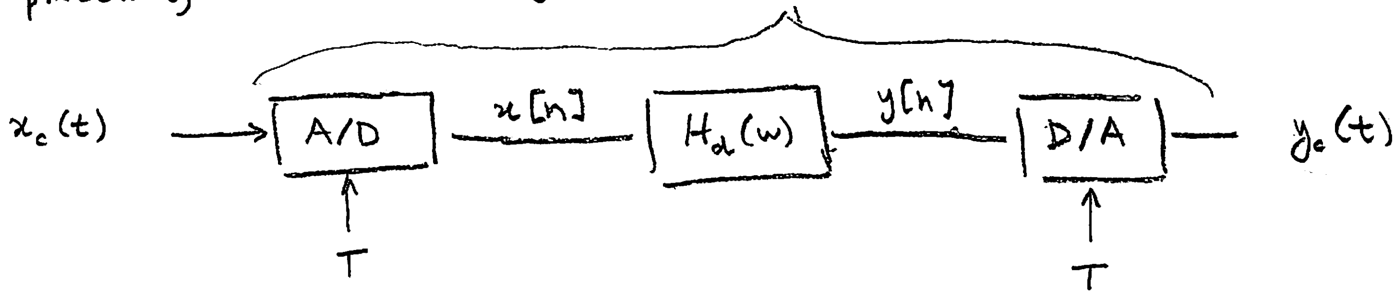
Advantages of FIR :

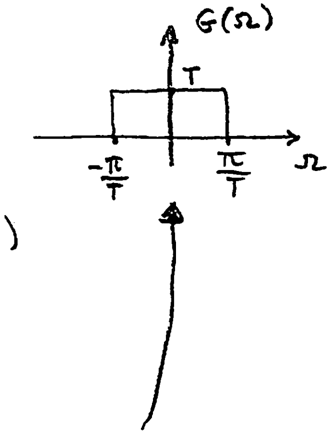
- Right choice if GLP is required. Only FIR has exact GLP
- Easy to design by hand with window-based approach

Disadvantages of FIR (via windowing)

- Same errors (ripples) in passband and stopband
- longer filter lengths than IIR

- DT processing of CT signals: $H_c^{(eff)}(\Omega)$





Main result: If $x_c(t)$ is bandlimited, $\frac{1}{T}$ is above Nyquist rate, ideal D/A:

then the effective analog system is $H_c^{(eff)}(\Omega) = H_d(\Omega T)$

Notice that the whole system operates in the same sampling rate

Some applications may need to convert sampling rate

from T_1 to T_2

Example of multirate DSP:

$$\begin{tikzcd}
 & \text{smartphone} \ 320 \times 240 \\
\boxed{\text{picture on the web}} \arrow[ur] \arrow[dr, "\text{resize}"] & \\
 & \text{desktop} \ 1920 \times 1080
\end{tikzcd}$$

640×480 pixels

smartphone (downsampling, decimation)

desktop (upsampling, interpolation)

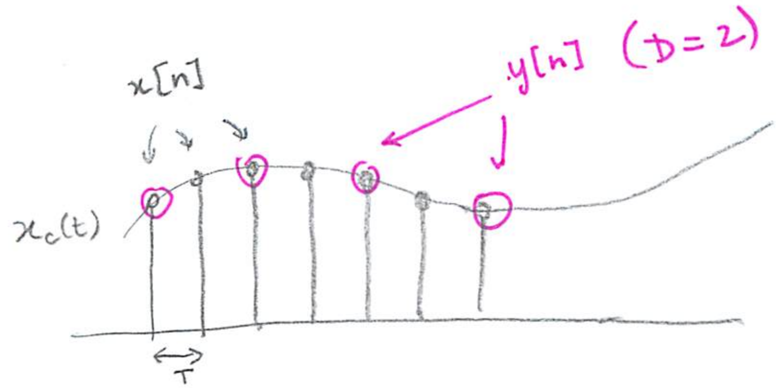
Using tools we've learned, <u>sampling rate conversion</u> could be implemented as follows:

$$\begin{tikzcd}
x[n] \arrow[r] & \boxed{D/A} \arrow[r, "x_c(t)"] & \boxed{A/D} \arrow[r] & y[n] \\
 & T_1 \arrow[u] & T_2 \arrow[u] &
\end{tikzcd}$$

originally sampled at T_1

We will look at sample rate conversion while staying in the digital domain

Downsampling (by an integer)



Notation (downsampler)

$$x[n] \longrightarrow \boxed{\downarrow D} \longrightarrow y[n]$$

D is an integer

$x[n] = x_c(nT)$

convert to $T_2 = DT$

$y[n] = x_c(nDT) = x[Dn]$

In frequency domain?

$$X_d(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega + 2\pi k}{T}\right)$$

$$Y_d(\omega) = \frac{1}{DT}\sum_{m=-\infty}^{\infty} X_c\left(\frac{\omega + 2\pi m}{DT}\right) = \frac{1}{DT}\sum_{m} X_c\left(\frac{\frac{\omega}{D} + \frac{2\pi m}{D}}{T}\right)$$

$m = D\cdot k + l$

where $-\infty \le k \le \infty$

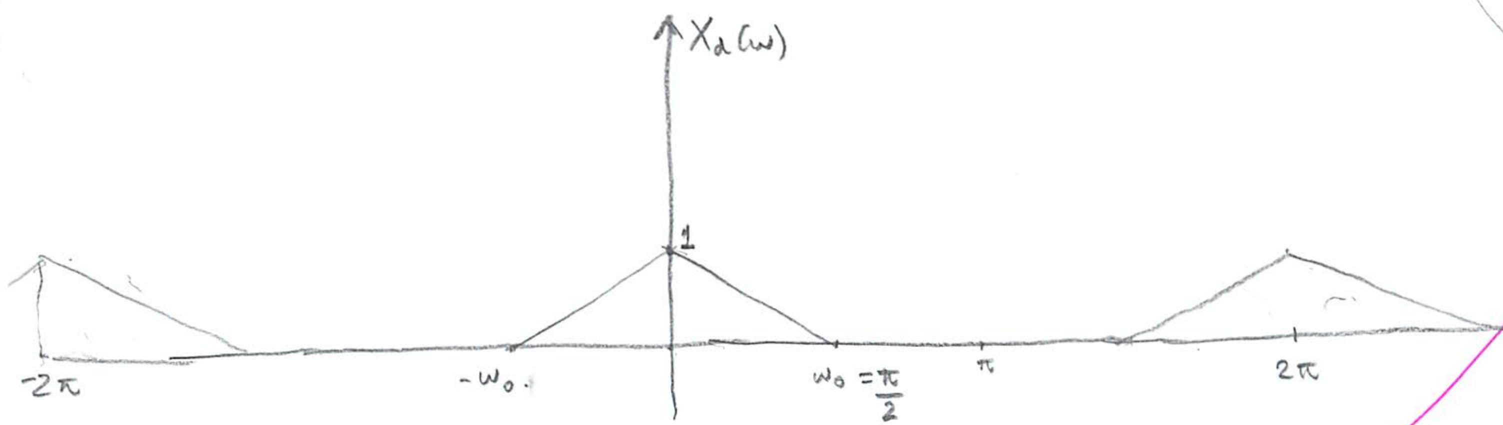
and $0 \le l \le D-1$

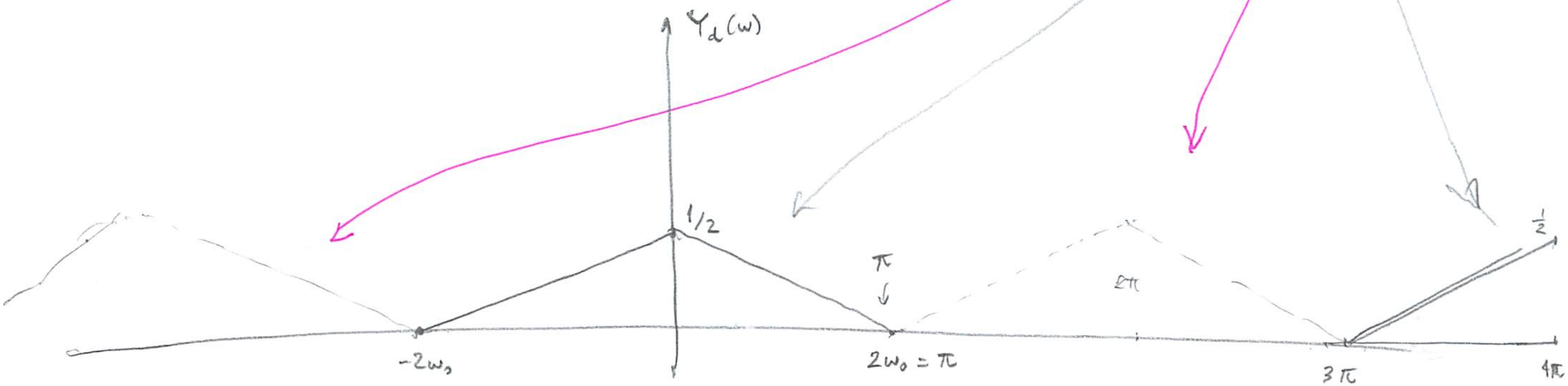
$$= \frac{1}{DT}\sum_{k=-\infty}^{\infty}\sum_{l=0}^{D-1} X_c\left(\frac{\frac{\omega}{D} + 2\pi k + \frac{2\pi l}{D}}{T}\right) = \frac{1}{D}\sum_{l=0}^{D-1} \underbrace{\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{\frac{\omega + 2\pi l}{D} + 2\pi k}{T}\right)}_{X_d\left(\frac{\omega + 2\pi l}{D}\right)}$$

$$\Rightarrow Y_d(\omega) = \frac{1}{D}\sum_{l=0}^{D-1} X_d\left(\frac{\omega + 2\pi l}{D}\right)$$

In words: Downsampling by D creates D shifted and scaled copies of $X_d(\omega)$

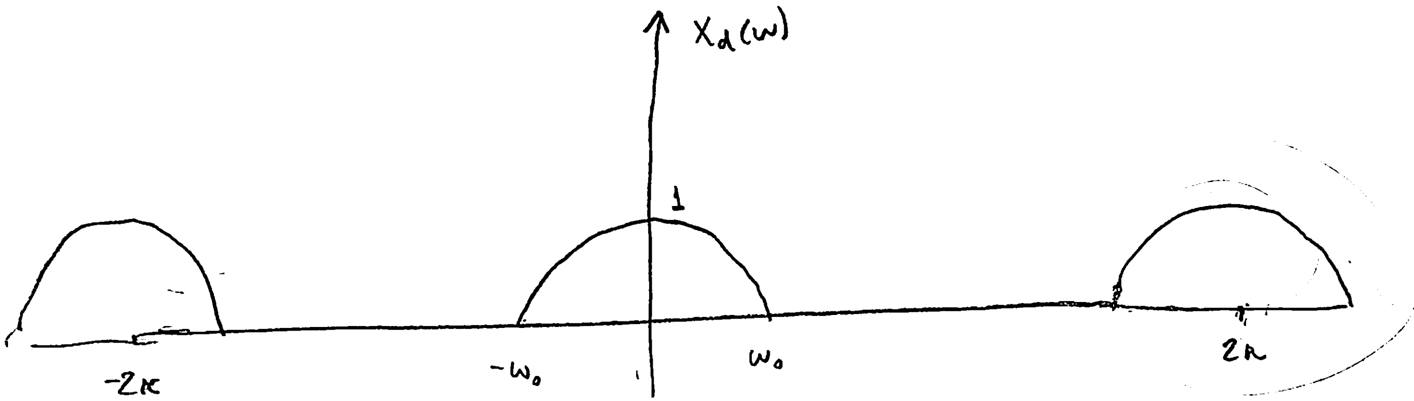
Ex: $D = 2$ $\quad x[n] - \textcircled{\downarrow 2} - y[n]$ $\quad Y_d(\omega) = \frac{1}{2}\left[X_d\left(\frac{\omega}{2}\right) + X_d\left(\frac{\omega + 2\pi}{2}\right) \right]$



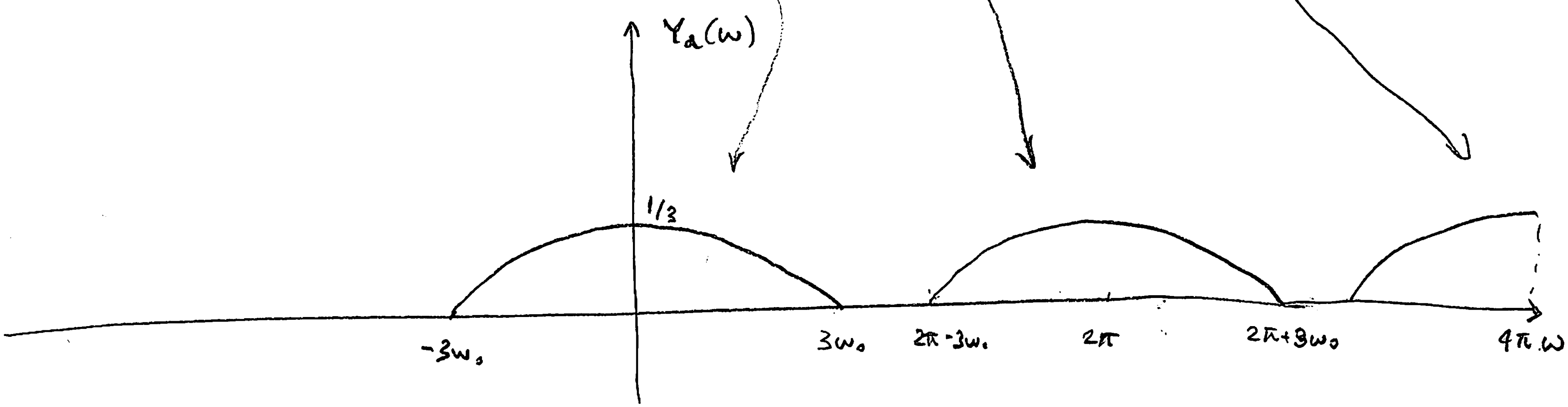


What if $\omega_0 > \frac{\pi}{2}$? Aliasing.

Ex: $D = 3$:

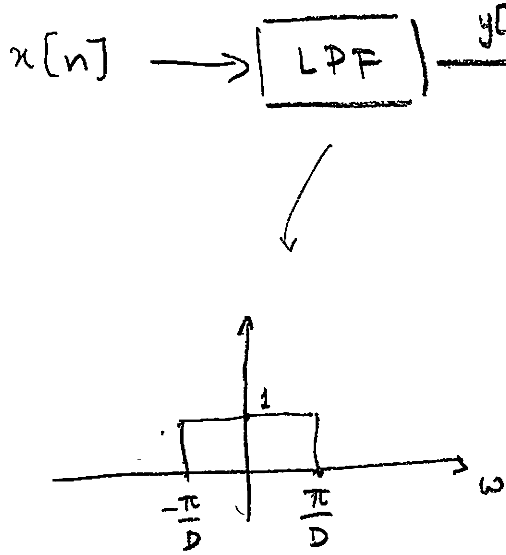


$$Y_d(\omega) = \frac{1}{3} \cdot \sum_{l=0}^{2} X_d\left(\frac{\omega + 2\pi l}{3}\right) = \frac{1}{3}\left[X_d\left(\frac{\omega}{3}\right) + X_d\left(\frac{\omega + 2\pi}{3}\right) + X_d\left(\frac{\omega + 4\pi}{3}\right)\right]$$



Aliasing if $\omega_0 > \frac{\pi}{3}$. $\boxed{\text{In general, aliasing if } \omega_0 > \frac{\pi}{D}}$

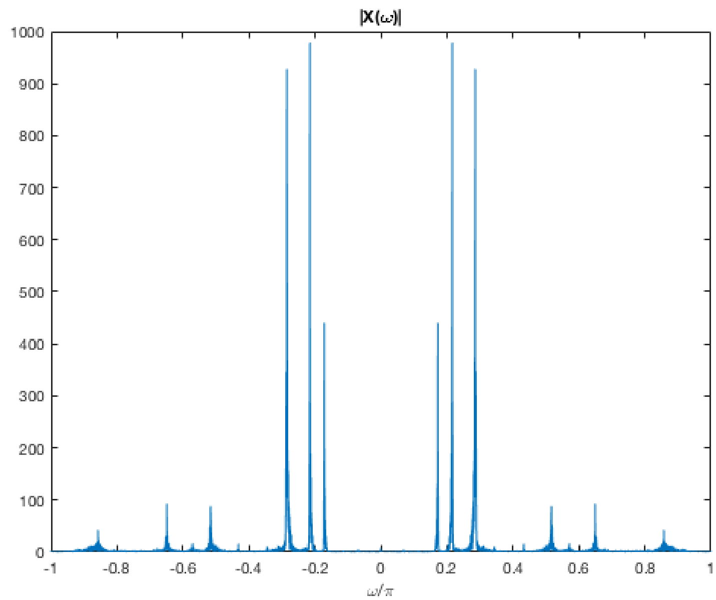
Solution: To avoid aliasing when downsampling by D, we apply a lowpass filter with $w_c = \frac{\pi}{D}$ before downsampling



```
load train; x = y;
```

```
sound(x, Fs); % Listen to the audio signal
```

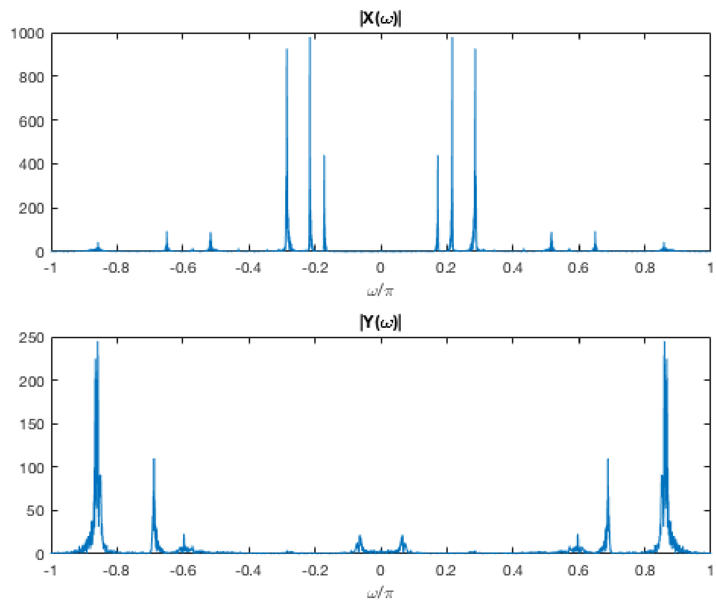
```
% magnitude of DTFT (via fft)
figure(1)
w = linspace(-pi, pi, length(x));
subplot(1,1,1), plot(w/pi, abs((fftshift(fft(x))))); ...
    title('|X(\omega)|'), xlabel('\omega/\pi')
```



```
% Downsampling by D

D = 4;
y = x(1:D:end);

figure(2)
subplot(2,1,1), plot(w/pi, abs((fftshift(fft(x))))); ...
    title('|X(\omega)|'), xlabel('\omega/\pi')
subplot(2,1,2), plot(linspace(-pi,pi,length(y))/pi, ...
    abs((fftshift(fft(y)))); title('|Y(\omega)|'), xlabel('\omega/\pi')
```



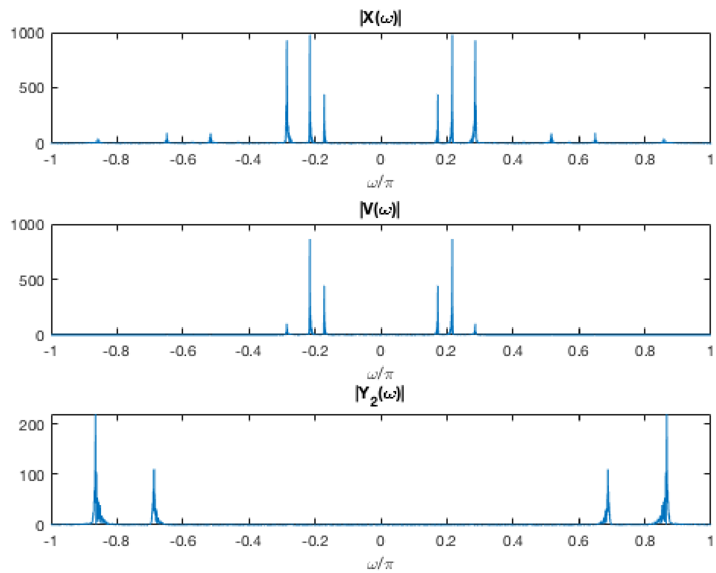

```
sound(y,Fs/D);
```

```
% Filter before downsampling

h = fir1(50, 1/D);
v = filter(h, 1, x);

y2 = v(1:D:end);
```

```
figure(3)
subplot(3,1,1), plot(w/pi, abs((fftshift(fft(x))))); ...
    title('|X(\omega)|'), xlabel('\omega/\pi')
subplot(3,1,2), plot(linspace(-pi,pi,length(v))/pi, ...
    abs((fftshift(fft(v))))); title('|V(\omega)|'), xlabel('\omega/\pi')
subplot(3,1,3), plot(linspace(-pi,pi,length(y2))/pi, ...
    abs((fftshift(fft(y2))))); title('|Y_2(\omega)|'), xlabel('\omega/\pi')
```




```
sound(y, Fs/D);
```

```
sound(y2, Fs/D);
```