BINGHAMTON UNIVERSITY

State University of New York

EECE 301
Signals & Systems
Prof. Mark Fowler

Note Set #28

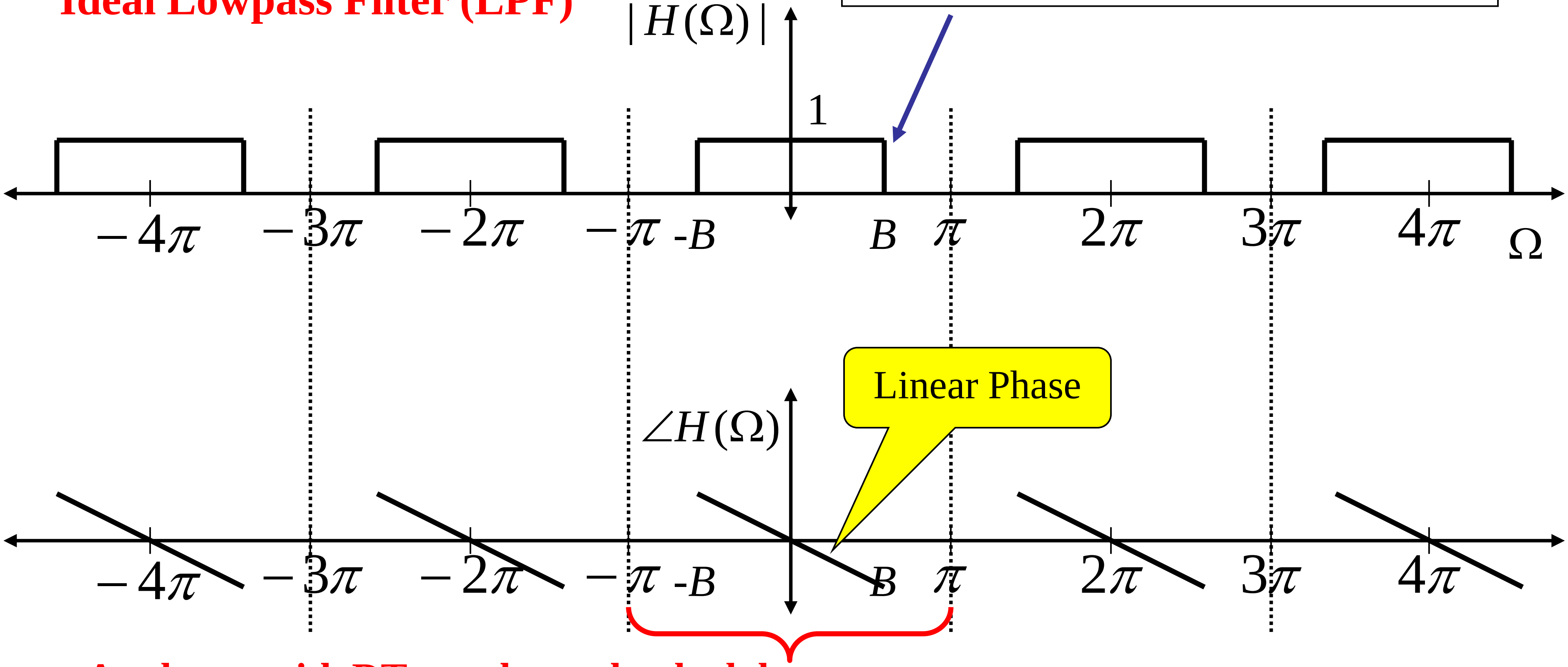
- D-T Systems: DT Filters – Ideal & Practical

Ideal D-T Filters

Just as in the CT case… we can specify filters. We looked at the ideal filter for the CT case… here we look at it for the DT case.

Ideal Lowpass Filter (LPF)

Cut-off frequency = *B* rad/sample



Linear Phase

As always with DT… only need to look here

As for CT… there are lowpass, highpass, bandpass, and bandstop filters.

Why can't an ideal filter exist in practice?? Same as for CT…. It is non-causal

For the ideal LPF $H(\Omega) = p_{2B}(\Omega)e^{-j\Omega n_d}$ $\quad \Omega \in [-\pi, \pi)$ repeats elsewhere

Now consider applying a delta function as its input: $x[n] = \delta[n] \leftrightarrow X(\Omega) = 1$

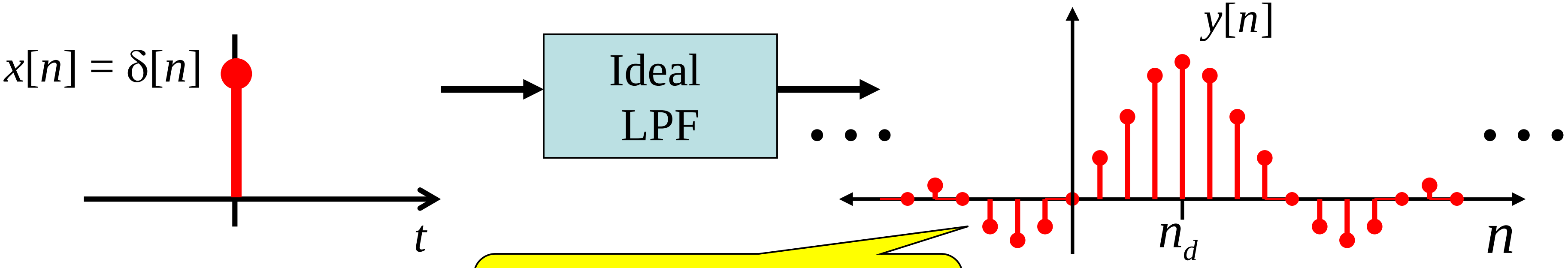
Then the output has DTFT:

$$Y(\Omega) = X(\Omega)H(\Omega) = p_{2B}(\Omega)e^{-j\Omega n_d} \quad \Omega \in [-\pi, \pi) \quad \text{repeats elsewhere}$$

Linear Phase Imparts Delay

From the DTFT Table: $\boxed{\frac{B}{\pi}\text{sinc}\left[\frac{B}{\pi}n\right]} \Longleftrightarrow \boxed{\sum_{k=-\infty}^{\infty} p_{2B}(\Omega + 2\pi k)}$

So the response to a delta (applied at $t = 0$) is: $y[n] = (B/\pi)\text{sinc}\left[(B/\pi)(n - n_d)\right]$



Starts before input starts…
Thus, system is non-causal!

Causal LPF Design – Truncated sinc Method

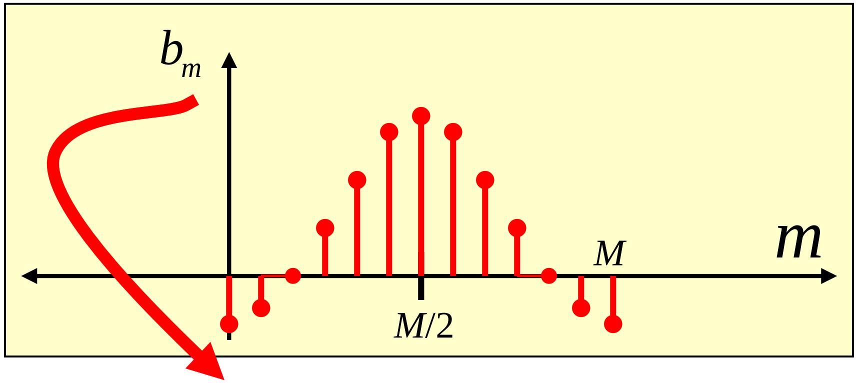
In practice, the best we can do is try to approximate the ideal LPF.

We already tried this: $y[n] = \frac{1}{N}x[n] + \frac{1}{N}x[n-1] + \ldots + \frac{1}{N}x[n-(N-1)]$

A "really bad" LPF!

But… now we've seen that a shifted sinc function seems to be involved…

…so a simple approach is to define the "b" coefficients in terms of a ***truncated*** shifted sinc function:



B = "Cutoff Frequency"

$$\boxed{b_m = (B/\pi)\,\mathrm{sinc}\big[(B/\pi)(m - M/2)\big], \quad m = 0, 1, 2, \ldots, M}$$

$$\boxed{y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]}$$

Causal, Non-Recursive Filter

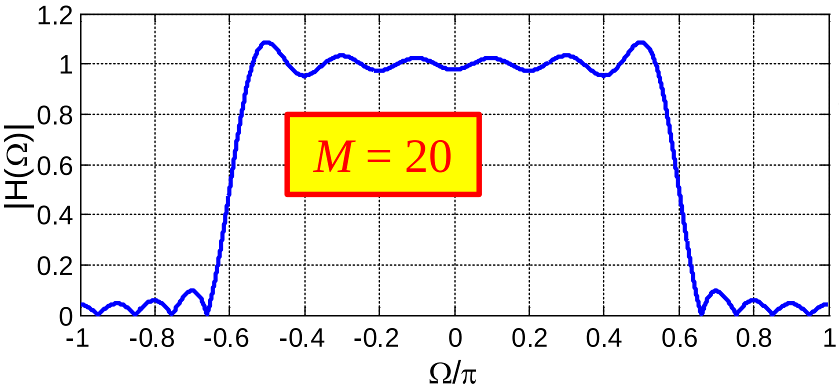
Error in Video

$$\boxed{H(z) = b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}$$

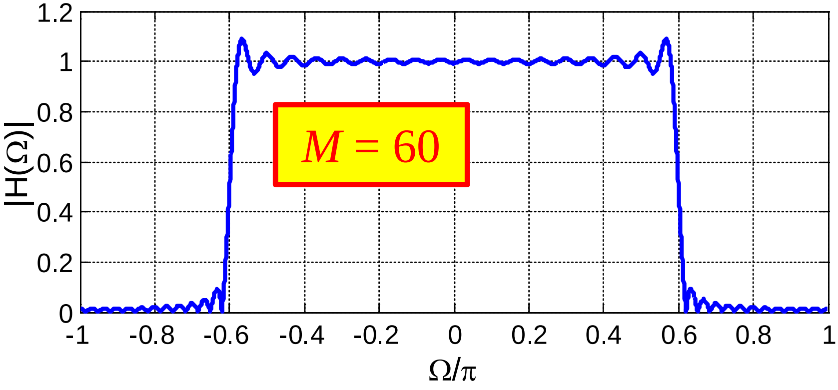
$$\boxed{H(\Omega) = b_0 + b_1 e^{-j\Omega} + \ldots + b_M e^{-j\Omega M}}$$

Let’s see how well this method works… These all have Linear Phase!

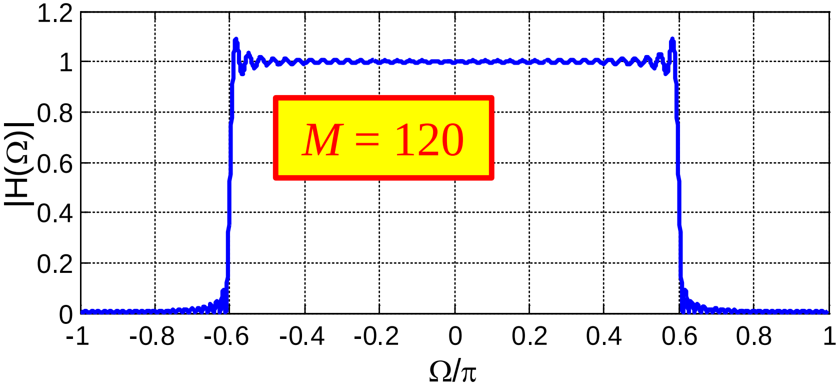
All cases
for
$B = 0.6\pi$



Frequency Response of a “sinc filter” truncated to **21 samples**



Frequency Response of a “sinc filter” truncated to **61 samples**



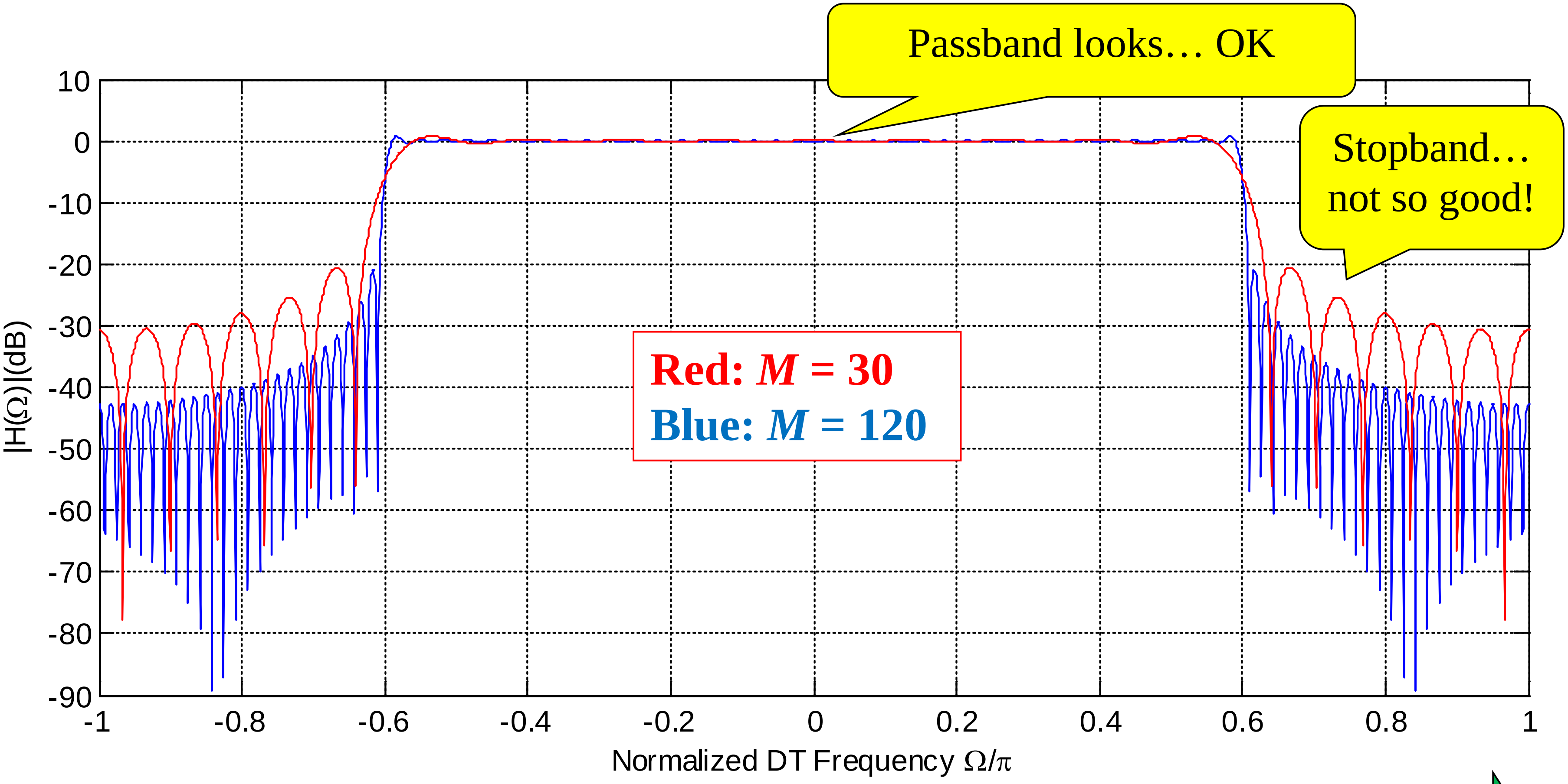
Frequency Response of a “sinc filter” truncated to **121 samples**

Some general insight: Longer lengths for the truncated impulse response Gives better approximation to the ideal filter response!!

For DT filters… "always" plot in dB but "never" use a log frequency axis!

This "truncated sinc" approach is a very simplistic approach and does not yield the best possible filters… as we can see even better in the dB plot below!

There are very powerful methods for designing REALLY good DT filters… we'll look at those in the next set of notes.



Passband looks… OK

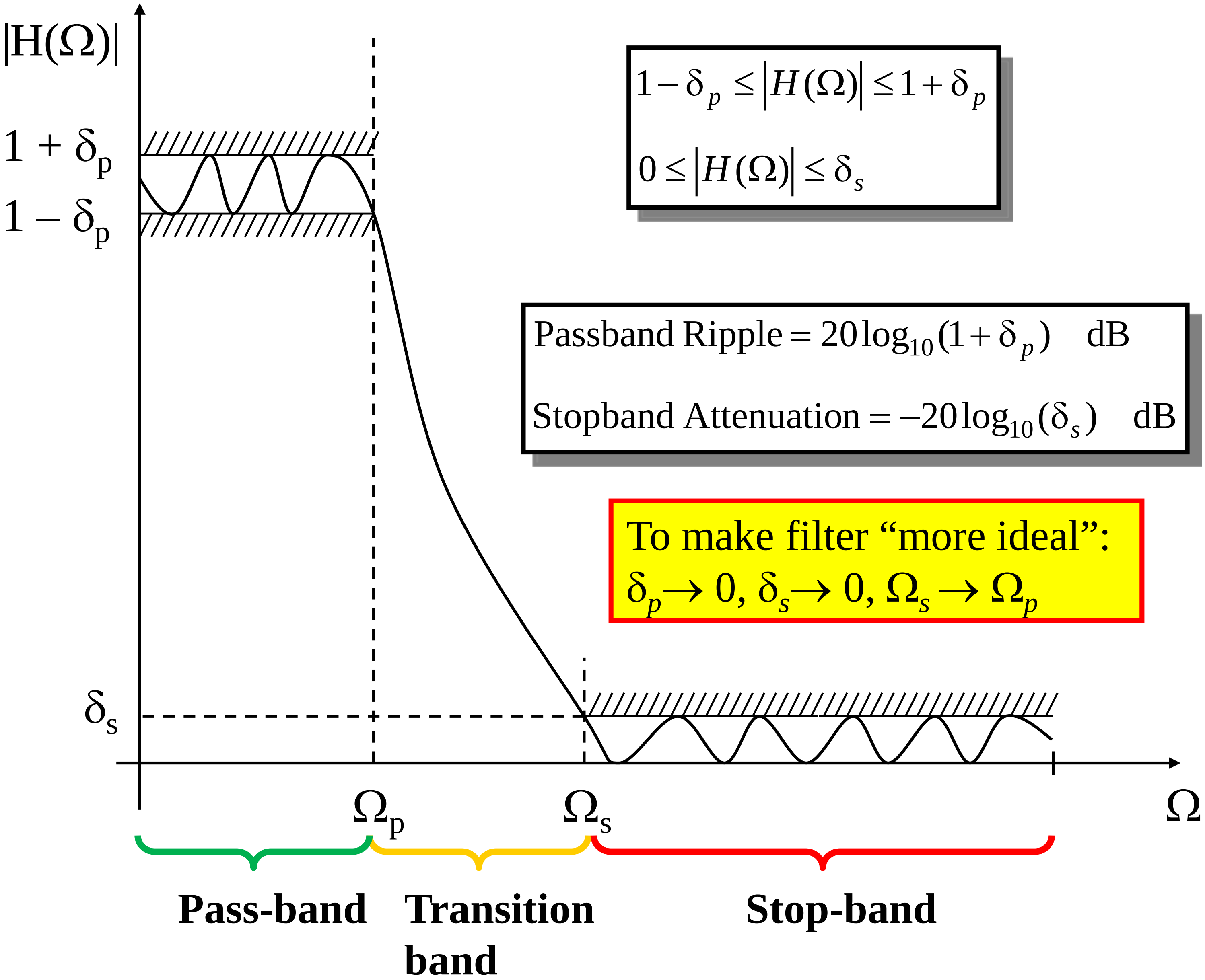
Stopband… not so good!

Red: *M* = 30

Blue: *M* = 120

Practical Filter Specification

Lowpass Filter Specification



$$1-\delta_p \le |H(\Omega)| \le 1+\delta_p$$

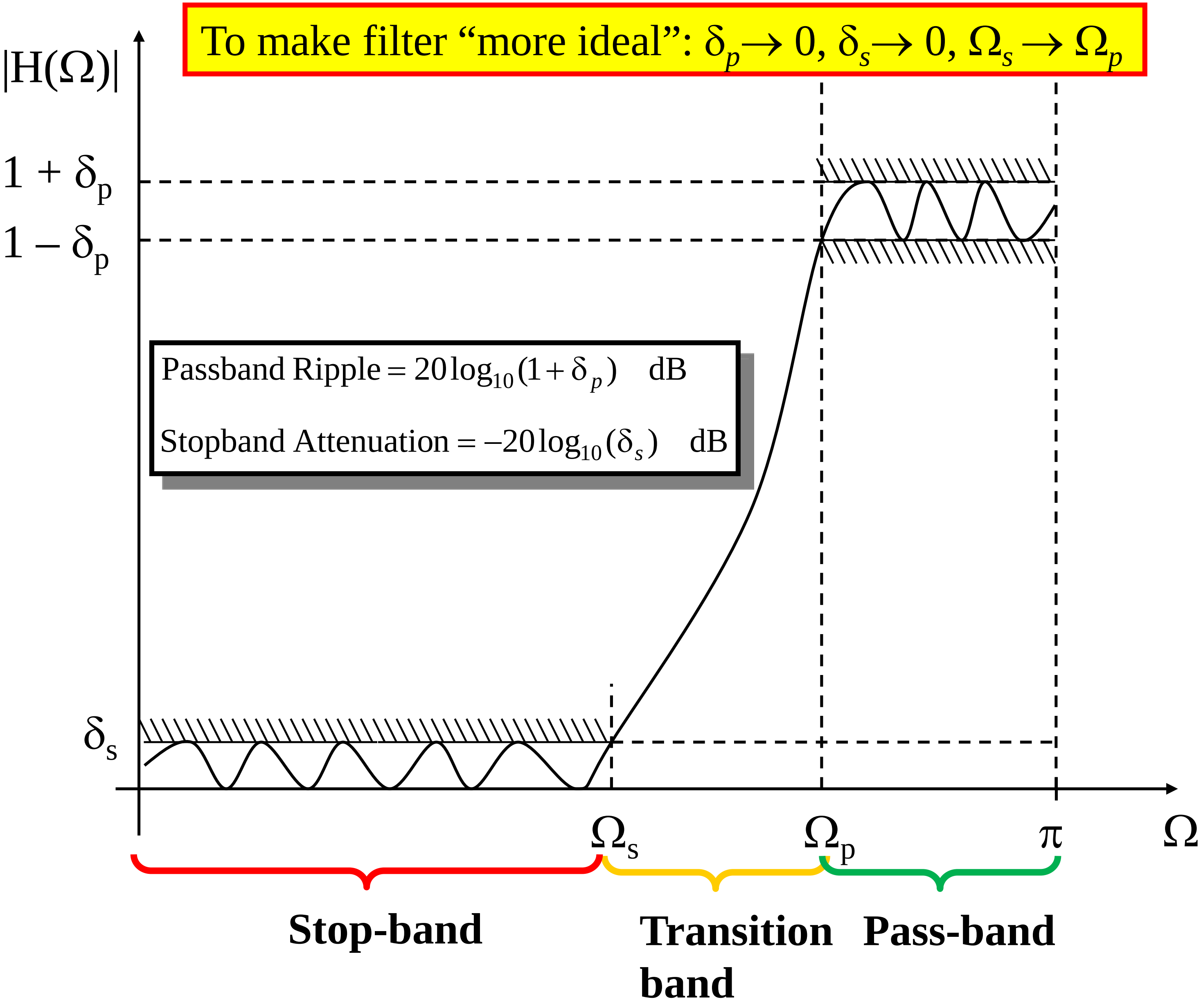
$$0 \le |H(\Omega)| \le \delta_s$$

$$\text{Passband Ripple} = 20\log_{10}(1+\delta_p) \quad \text{dB}$$

$$\text{Stopband Attenuation} = -20\log_{10}(\delta_s) \quad \text{dB}$$

To make filter “more ideal”:
$\delta_p \rightarrow 0,\ \delta_s \rightarrow 0,\ \Omega_s \rightarrow \Omega_p$

Highpass Filter Specification



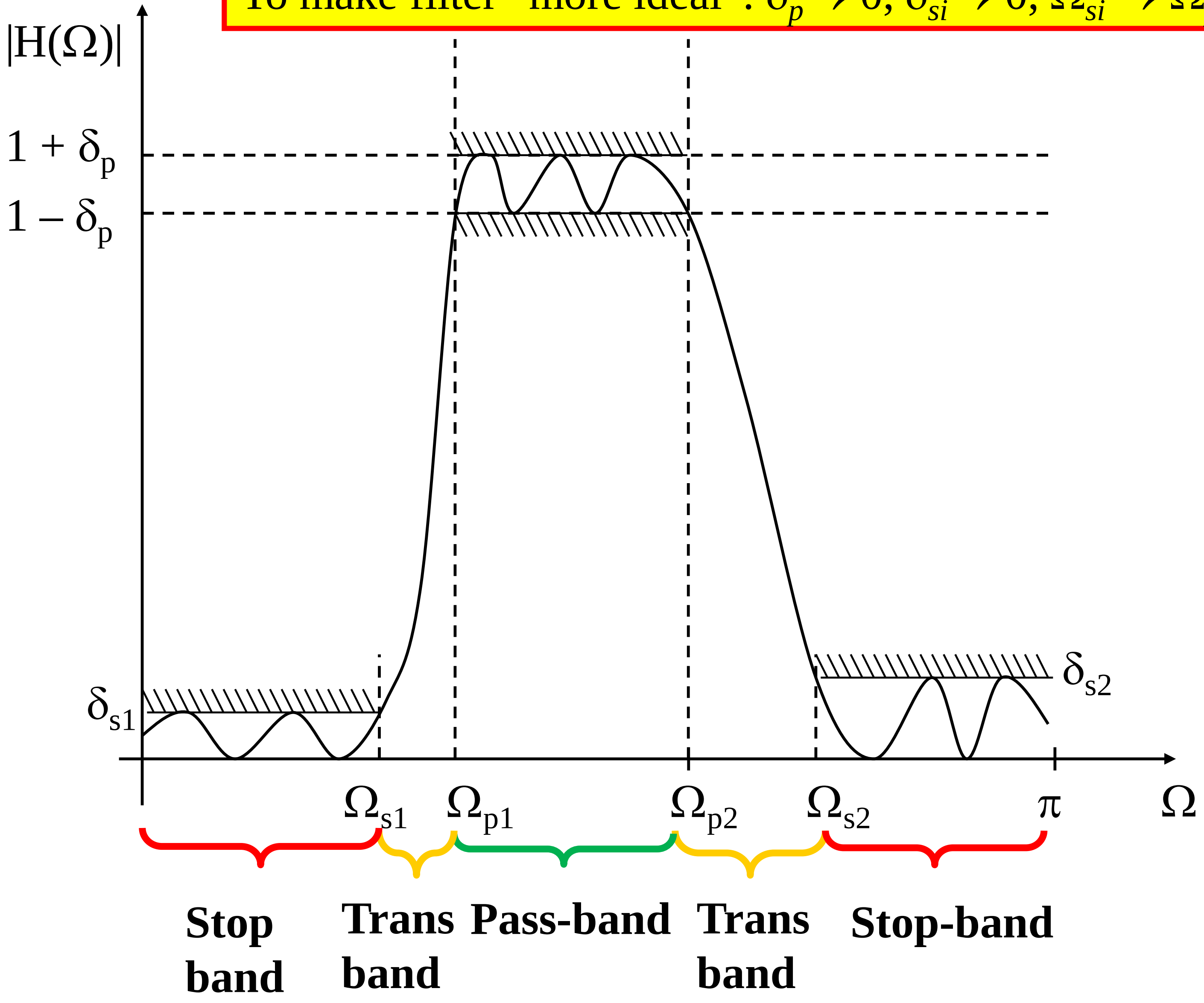
To make filter “more ideal”: $\delta_p \rightarrow 0$, $\delta_s \rightarrow 0$, $\Omega_s \rightarrow \Omega_p$

$$\text{Passband Ripple} = 20\log_{10}(1+\delta_p) \quad \text{dB}$$

$$\text{Stopband Attenuation} = -20\log_{10}(\delta_s) \quad \text{dB}$$

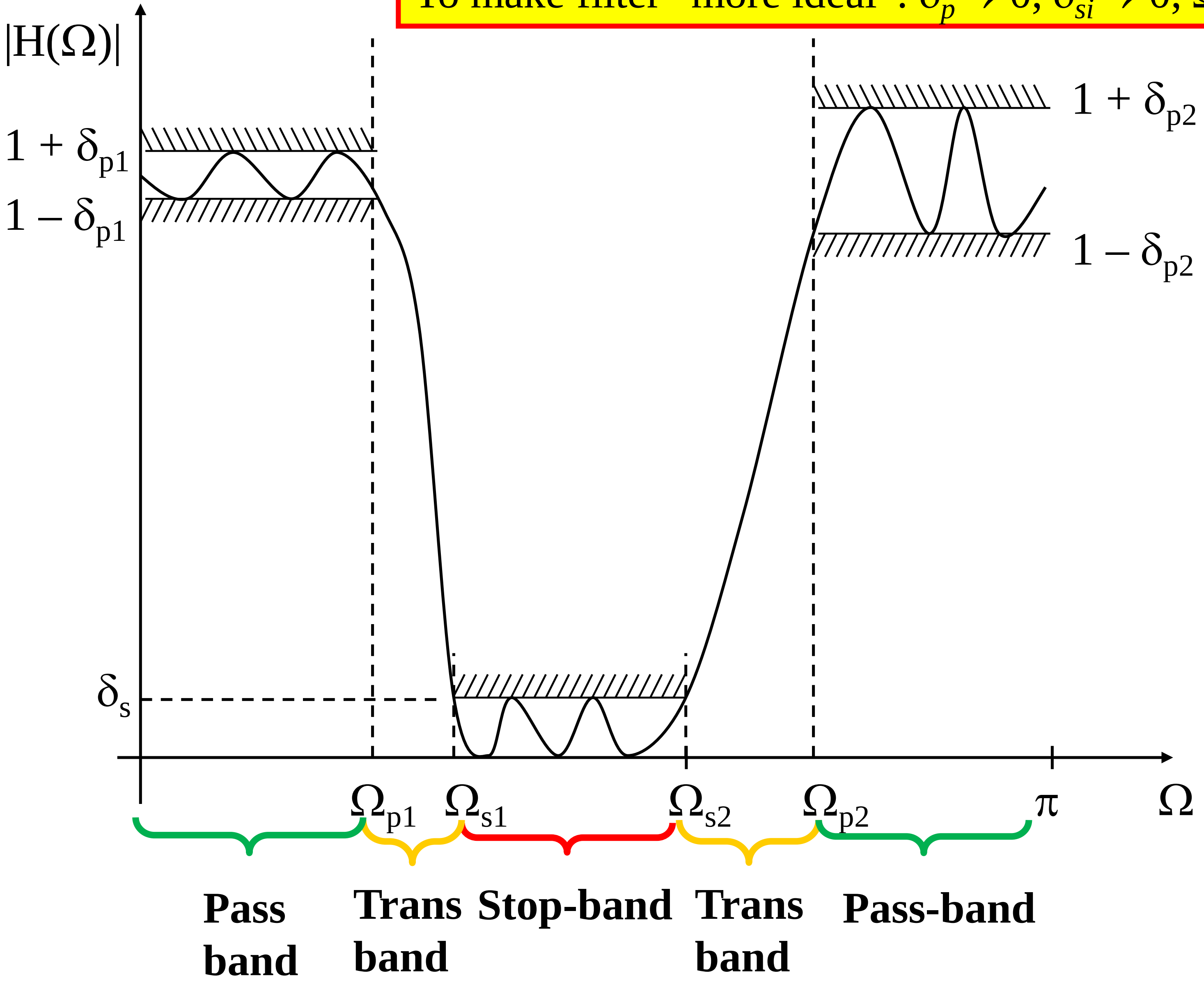
Bandpass Filter Specification

To make filter “more ideal”: $\delta_p \to 0$, $\delta_{si} \to 0$, $\Omega_{si} \to \Omega_{pi}$



Bandstop Filter Specification

To make filter "more ideal": $\delta_p \rightarrow 0$, $\delta_{si} \rightarrow 0$, $\Omega_{si} \rightarrow \Omega_{pi}$



DT Filter Types

There are 2 main types of DT filters, based on the structure of their Diff. Equation:

Recursive Filters… Have **Feedback** in the Difference Equation

$$\boxed{y[n] + \underbrace{a_1 y[n-1] + \ldots + a_N y[n-N]}_{\textbf{Feedback Terms}} = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]}$$

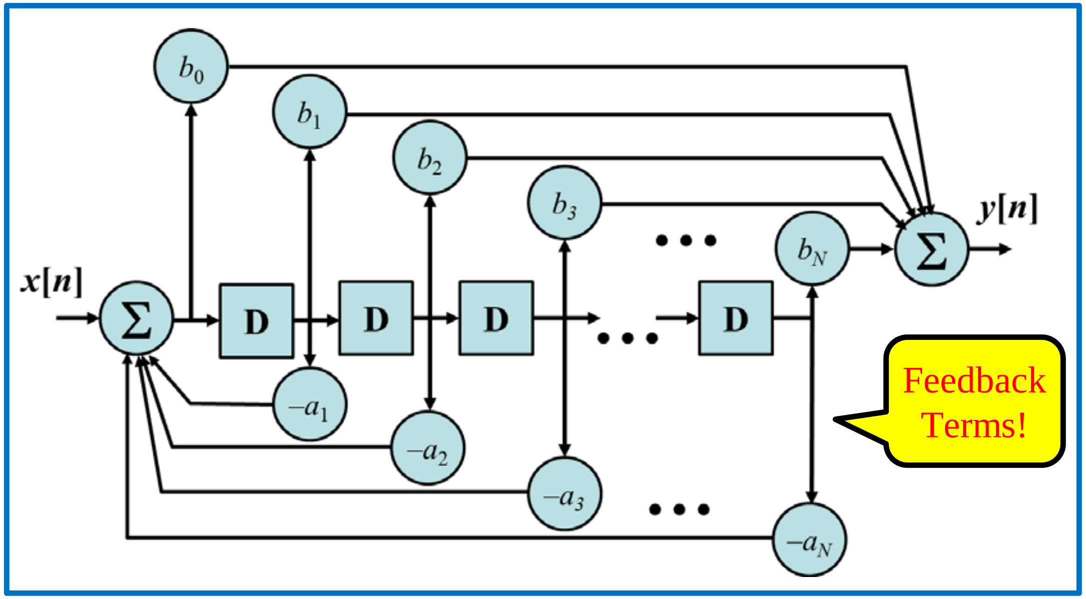
$$\Rightarrow \boxed{y[n] = -\underbrace{\sum_{i=1}^{N} a_i y[n-i]}_{\textbf{Feedback Terms}} + \sum_{i=0}^{M} b_i x[n-i]}$$

Non-Recursive Filters… Have **No Feedback** in the Difference Equation

$$\boxed{y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]}$$

Recursive Filters

$$y[n] = -\underbrace{\sum_{i=1}^{N} a_i y[n-i]}_{\text{Feedback Terms}} + \sum_{i=0}^{M} b_i x[n-i]$$

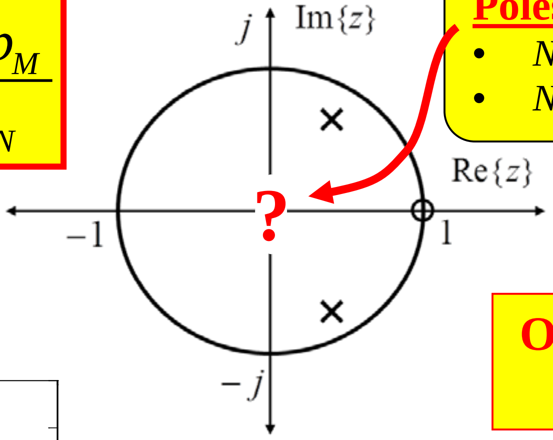


Feedback Terms!

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}$$

$$H(z) = z^{(N-M)} \frac{b_0 z^{M} + b_1 z^{M-1} + \ldots + b_M}{z^{N} + a_1 z^{N-1} + \ldots + a_N}$$

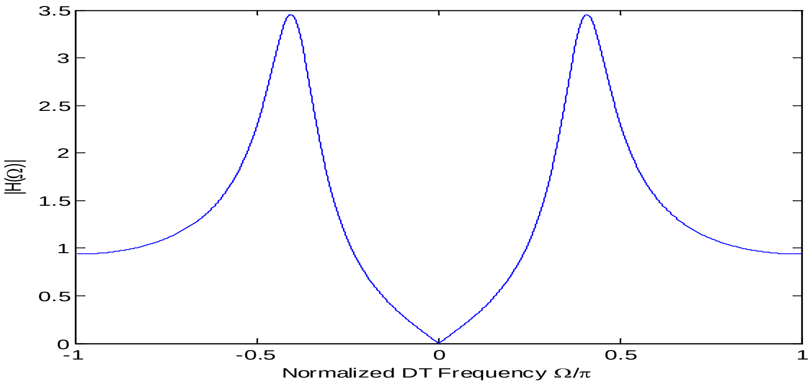
$$H(\Omega) = \frac{b_0 + b_1 e^{-j\Omega} + \ldots + b_M e^{-j\Omega M}}{1 + a_1 e^{-j\Omega} + \ldots + a_N e^{-j\Omega N}}$$



<u>Poles/Zeros @ Origin</u>

- $N - M > 0$: $\boldsymbol{N - M}$ **<u>zeros</u> @ Origin**
- $N - M < 0$: $\boldsymbol{M - N}$ **<u>poles</u> @ Origin**

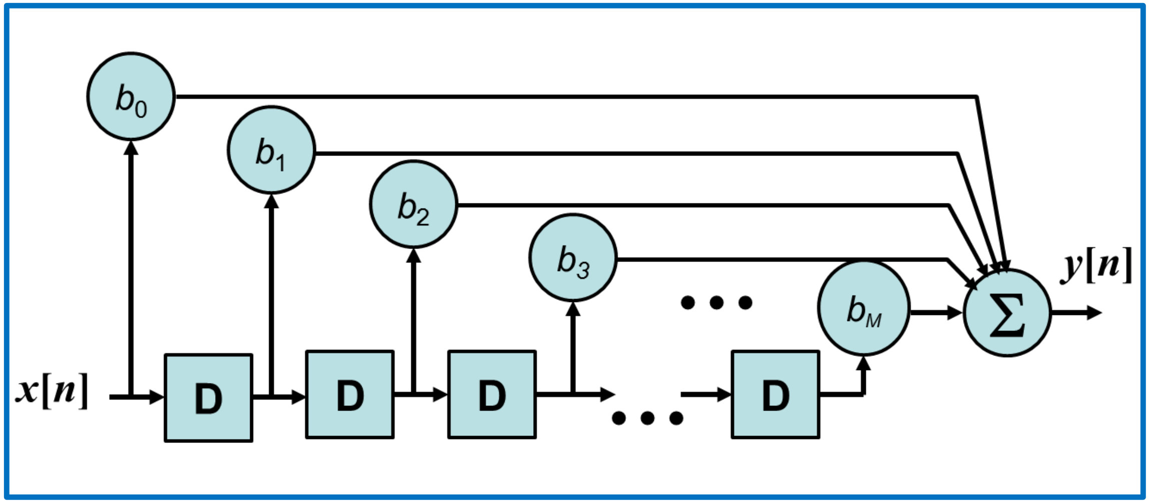
Only Feedback can give poles <u>not</u> at origin!



Must worry about stability!
Make sure all poles are inside UC!

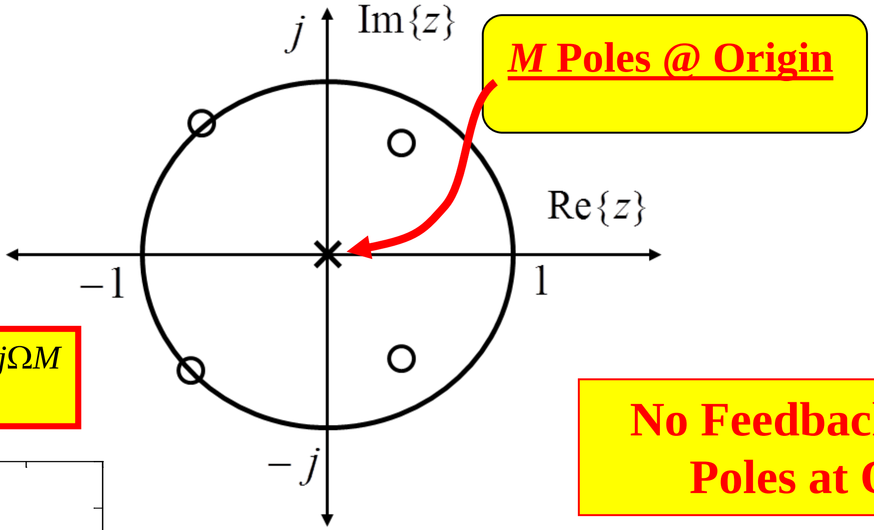
Non-Recursive Filters

$$y[n]=\sum_{i=0}^{M} b_i x[n-i]$$



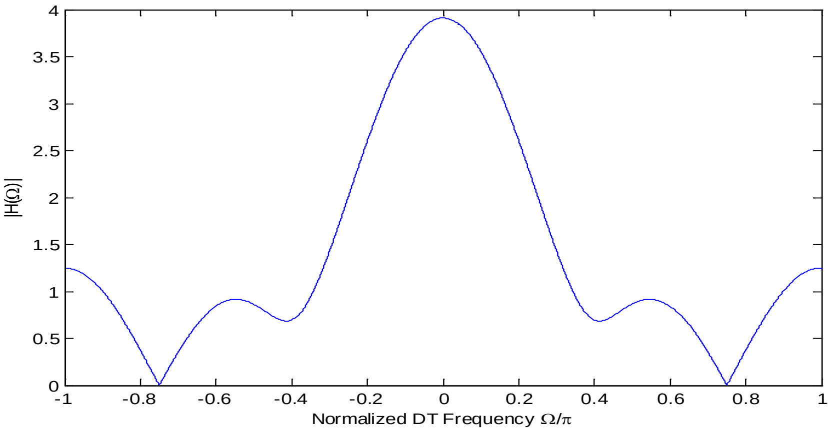
$$H(z)=b_0+b_1z^{-1}+\ldots+b_Mz^{-M}$$

$$H(z)=\frac{b_0z^{M}+b_1z^{M-1}+\ldots+b_M}{z^{M}}$$



***M* Poles @ Origin**

$$H(\Omega)=b_0+b_1e^{-j\Omega}+\ldots+b_Me^{-j\Omega M}$$



No Feedback… Only Poles at Origin!

No worries about stability! Always Stable!!!

IIR & FIR Filters

These are just new names for filters we've already seen!

IIR: "Infinite Impulse Response"… Impulse Response has infinite duration

FIR: "Finite Impulse Response"… Impulse Response has finite duration

Assumes no repeated roots &
No Direct Terms

Recursive Filters are IIR

$$\boxed{H(z) = z^{(N-M)}\frac{B(z)}{A(z)}}$$

$$\boxed{h[n] = k_1 p_1^n u[n] + k_2 p_2^n u[n] + \cdots + k_N p_N^n u[n]}$$

Non-Recursive Filters are FIR

$$\boxed{H(z) = b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}$$

$$\boxed{h[n] = \{\cdots 0, 0, \underset{\substack{\uparrow \\ n=0}}{b_0}, b_1, b_2, \ldots, b_M, 0, 0, 0, 0 \cdots\}}$$