EBU6018
Advanced Transform Methods

Sampling and the Discrete Fourier Transform (DFT)_1

Andy Watson

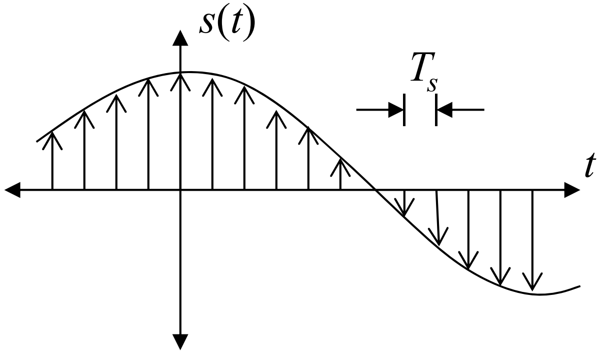
Sampling: Time Domain

- Many signals (all real-world signals) originate as continuous-time signals, e.g. conventional music or voice
- By sampling a continuous-time signal at isolated, equally-spaced points in time, we obtain a sequence of numbers (a discrete signal):

$$s[k] = s(k\,T_s)$$

$k \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

T_s is the sampling period.

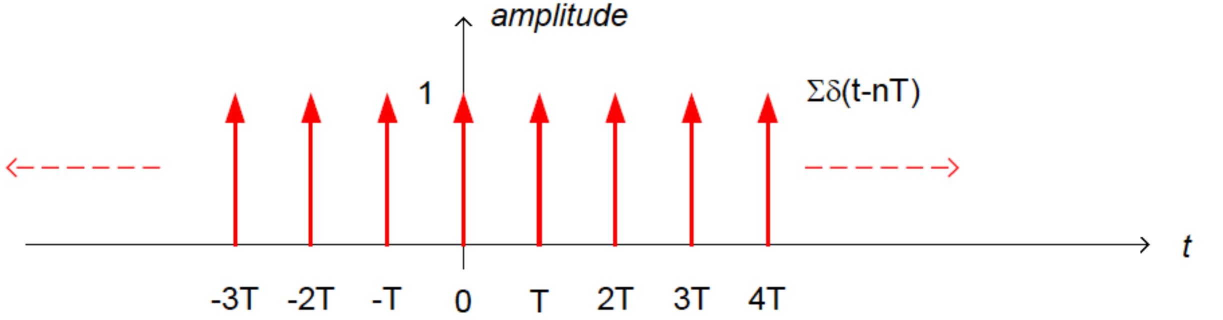


Sampled analog waveform

$$s_{sampled}(t) = s(t) \underbrace{\sum_{k=-\infty}^{\infty} \delta(t - k\,T_s)}_{\text{impulse train } \delta_{T_s}(t)} = \sum_{k=-\infty}^{\infty} \underbrace{s(k\,T_s)}_{s[k]} \delta(t - k\,T_s)$$

"Comb" of Delta Functions

$$f(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$



Sampling s(t) = multiplication of s(t) by a comb of delta functions

In frequency domain -> convolve by "FT of comb of delta functions".

Property of FT

So - What is "FT of comb of delta functions"?

Repeats with interval T, so can use Fourier Series expansion:

$$f(t) = \sum_{k=-\infty}^{+\infty} a_k e^{j2\pi k f_0 t} \quad \text{where} \quad f_0 = 1/T$$

So – need to solve for a_k

FT of Comb of Delta Functions

$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-j2\pi f_0 t}dt$$ Fourier series term

$$= \frac{1}{T}\int_{-T/2}^{T/2} \sum_{n=-\infty}^{\infty}\delta(t-nT)e^{-j2\pi f_0 t}dt$$ by expanding $f(t)$

$$= \frac{1}{T}\int_{-T/2}^{T/2} \delta(t)e^{-j2\pi f_0 t}dt$$ only $\delta(t-nT)$ in $-T/2 < t < T/2$

$$= \frac{1}{T}e^{-j2\pi f_0 0} = \frac{1}{T}$$ by action of delta function

so $f(t) = \sum_{k=-\infty}^{+\infty} a_k e^{j2\pi k f_0 t} = \frac{1}{T}\sum_{k=-\infty}^{+\infty} e^{j2\pi k f_0 t}$ sum of sinusoids

$\omega = 2\pi f$

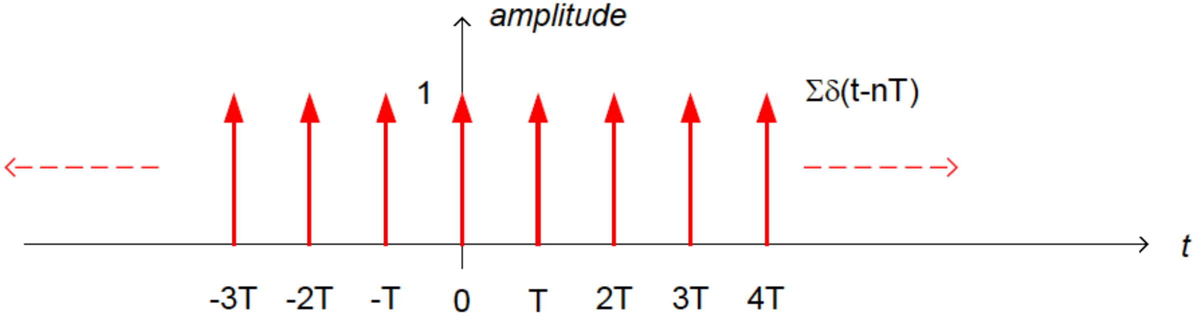
$F(f) = \frac{1}{T}\sum_{k=-\infty}^{+\infty}\delta(f-kf_0)$ or $F(\omega) = \frac{1}{T}\sum_{k=-\infty}^{+\infty}\delta(\omega-k\omega_0)$

Comb of Delta Functions: Summary

In Time Domain

$$f(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$

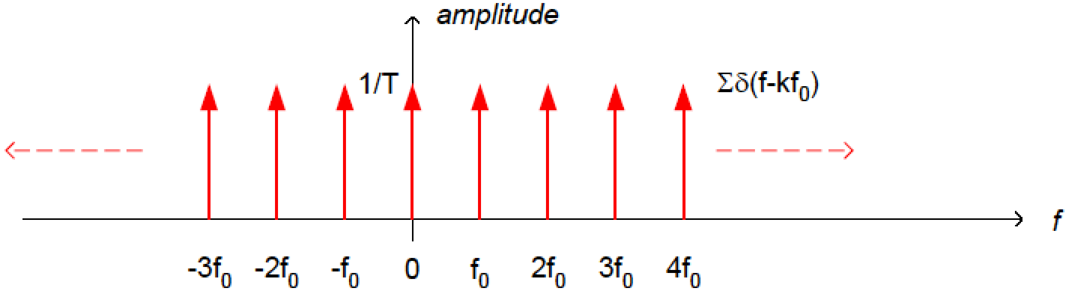
$$= \frac{1}{T} \sum_{k=-\infty}^{+\infty} e^{j2\pi k f_0 t}$$



In Frequency Domain

$$F(f) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta(f - kf_0)$$

$$F(\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta(\omega - k\omega_0)$$



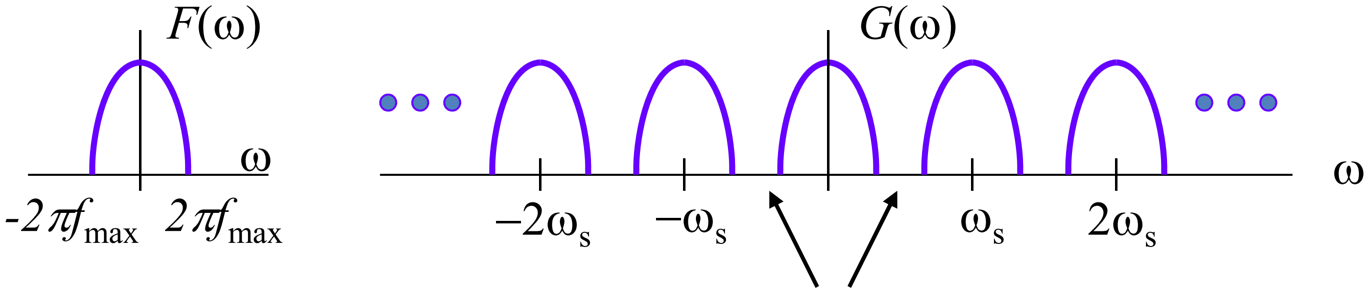
So: sampling in time domain = convolution with comb of deltas in freq domain

Sampling: Frequency Domain

- Sampling replicates the spectrum of continuous-time signal at integer multiples of sampling frequency
- Fourier series of impulse train where $\boldsymbol{w_s = 2\pi f_s}$

$$\delta_{T_s}(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\,T_s) = \frac{1}{T_s}\left(1 + 2\cos(\omega_s\, t) + 2\cos(2\,\omega_s\, t) + \cdots\right)$$

$$g(t) = f(t)\,\delta_{T_s}(t) = \frac{1}{T_s}\left(f(t) + \underbrace{2\,f(t)\cos(\omega_s\, t)}_{\text{Modulation by } \cos(\omega_s\, t)} + \underbrace{2\,f(t)\cos(2\,\omega_s\, t)}_{\text{Modulation by } \cos(2\,\omega_s\, t)} + \cdots \right)$$



$$\boxed{\text{gap if and only if } 2\pi f_{\max} < 2\pi f_s - 2\pi f_{\max} \Leftrightarrow f_s > 2\,f_{\max}}$$

If $f_s < 2f_{max}$ then the baseband and bandpass spectra overlap and cannot be separated

Amplitude Modulation by Cosine (produces multiple sidebands)

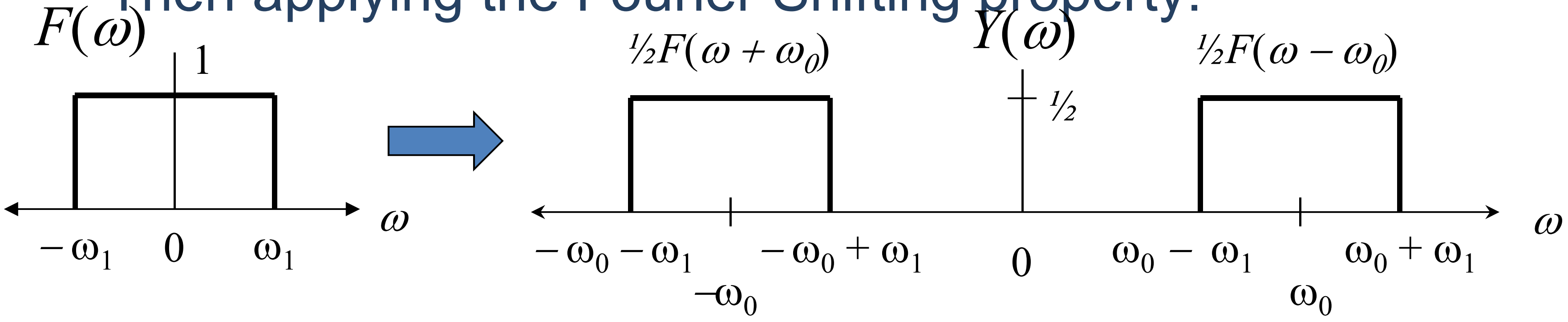
- Example: ***y*(*t*) = *f*(*t*) cos($\mathbf{w_0}$ *t*),** that is, modulation

  Assume *f*(*t*) is a signal that has bandwidth w_1

  Assume $w_1 << w_0$

  Y(w) is real-valued if *F*(w) is real-valued

  Then applying the Fourier Shifting property:



- Similar derivation for modulation with **sin($\mathbf{w_0}$ t)**
- Demodulation is lowpass filtering of the modulated signal

Shannon/Nyquist Sampling Theorem

- Continuous-time signal ***x*(*t*)** with frequencies no higher than $\boldsymbol{f_{max}}$ can be reconstructed from its samples $\boldsymbol{x(k\ T_s)}$ if samples taken at rate $\boldsymbol{f_s > 2\ f_{max}}$

  Nyquist rate = 2 f_{max}

  Nyquist frequency = $f_s / 2$

Critical Sampling if

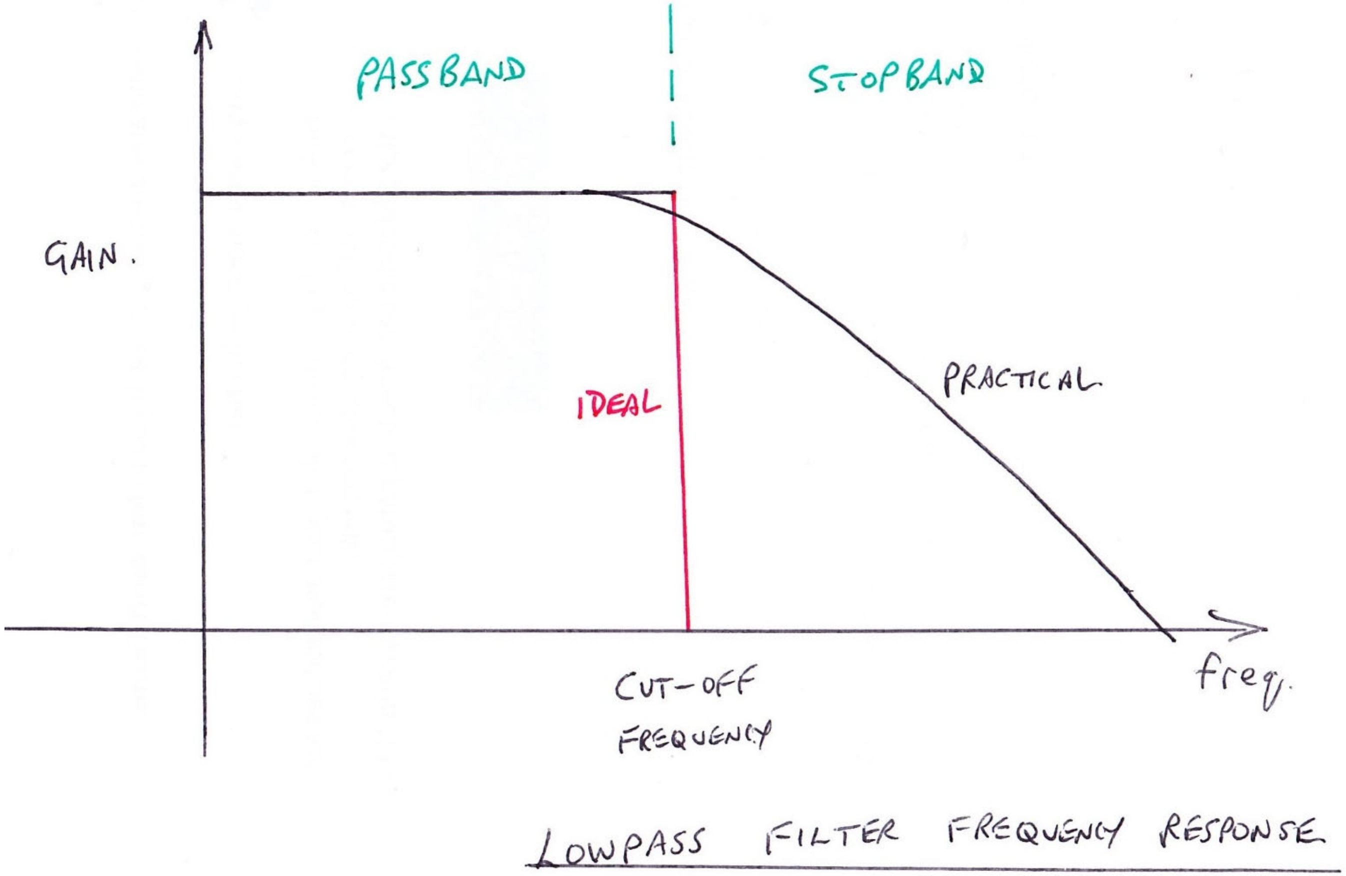
$f_s = 2\, f_{max}$

- Example: Sampling audio signals
  - Human hearing is from about 20 Hz to 20 kHz
  - So minimum sampling rate to allow reconstruction is 40 kHz

Bandlimiting

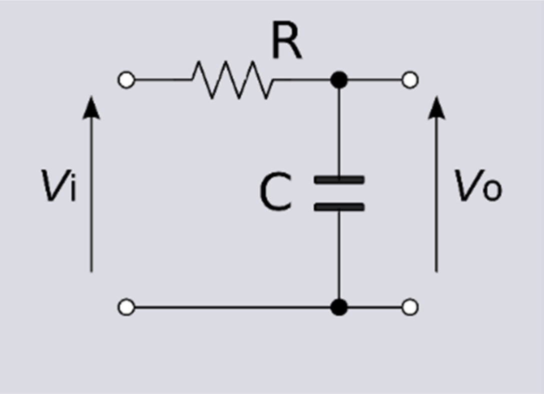
- Bandlimiting filters are low pass filters used to limit the sampling rate to the minimum required (i.e. to stop higher frequencies that we do not want)
- For audio signals apply lowpass filter before sampling to pass frequencies up to 20 kHz and reject high frequencies
- The filter roll-off needs to be steep enough to separate the wanted and unwanted frequencies
- Lowpass filter needs 10% of maximum passband frequency to roll off to a sufficiently small value (2 kHz rolloff in this case), hence high order filter.

Low-pass Filter



Low-pass Filter Roll-off Rate

The circuit diagram of a passive low-pass filter is:



The operation of the circuit is simply that the higher the frequency of the input signal, the lower the reactance of the capacitor so higher frequency signals are shunted to ground. Treating the circuit as a voltage divider:

$$Z_c = \frac{-j}{\omega C}$$

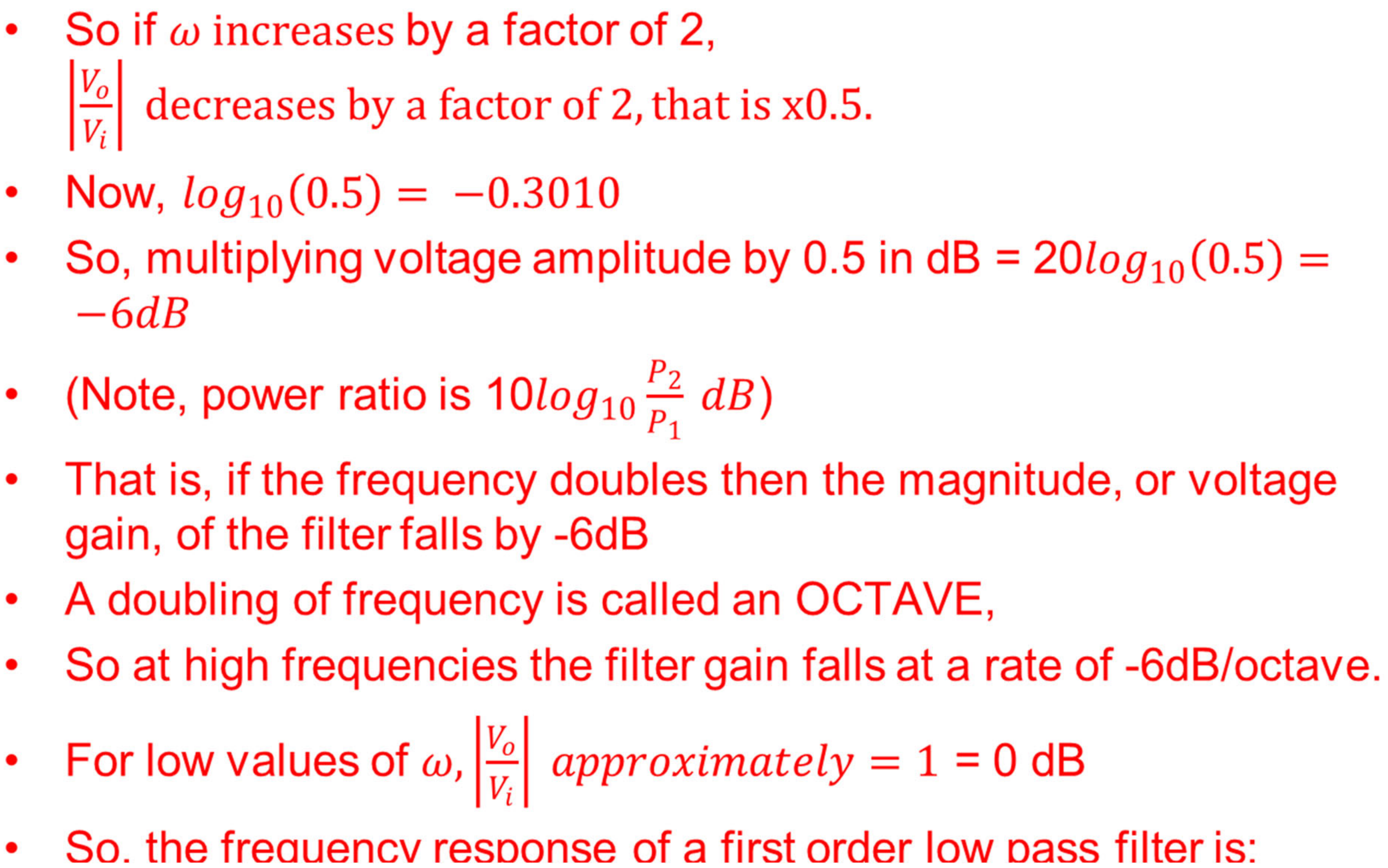
$$\frac{V_o}{V_i} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j\omega RC}$$

The time-constant of the circuit is RC

Magnitude = $\left|\frac{V_o}{V_i}\right| = \frac{1}{\sqrt{1+(\omega RC)^2}}$. This decreases as ω increases.

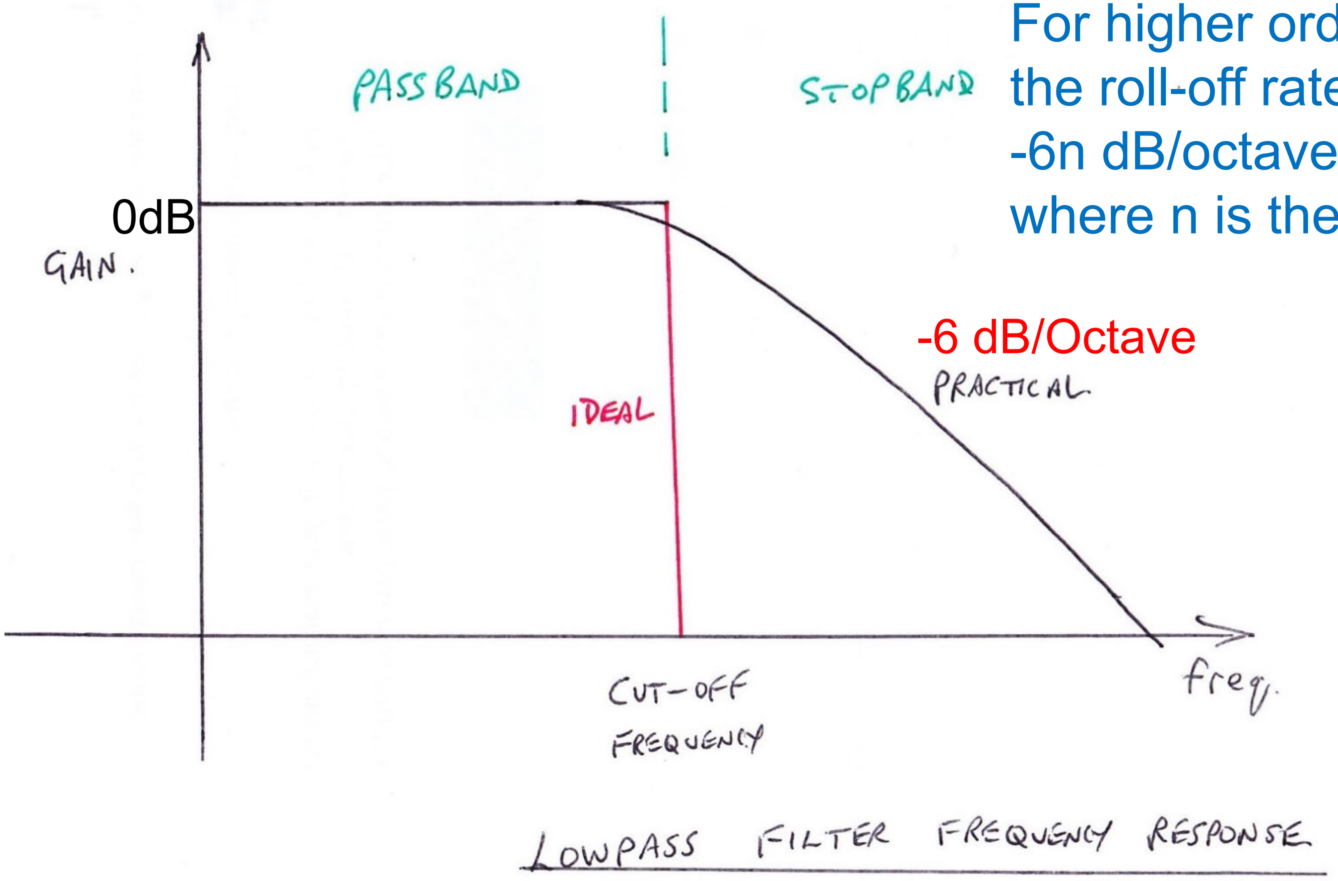
For large ω, $\left|\frac{V_o}{V_i}\right|$ $approx = \frac{1}{\omega RC}$

Low-pass Filter Roll-off Rate cont.



- So if ω increases by a factor of 2, $\left|\frac{V_o}{V_i}\right|$ decreases by a factor of 2, that is x0.5.
- Now, $log_{10}(0.5) = -0.3010$
- So, multiplying voltage amplitude by 0.5 in dB = $20log_{10}(0.5) = -6dB$
- (Note, power ratio is $10log_{10}\frac{P_2}{P_1}\, dB$)
- That is, if the frequency doubles then the magnitude, or voltage gain, of the filter falls by -6dB
- A doubling of frequency is called an OCTAVE,
- So at high frequencies the filter gain falls at a rate of -6dB/octave.
- For low values of ω, $\left|\frac{V_o}{V_i}\right|$ $approximately = 1$ = 0 dB
- So, the frequency response of a first order low pass filter is:

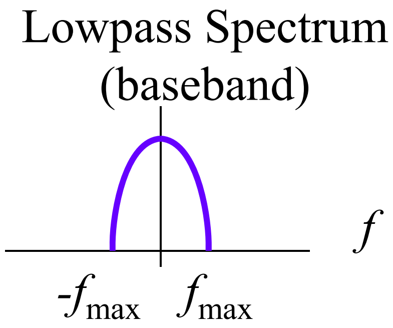
First order Low-pass Filter Roll-off

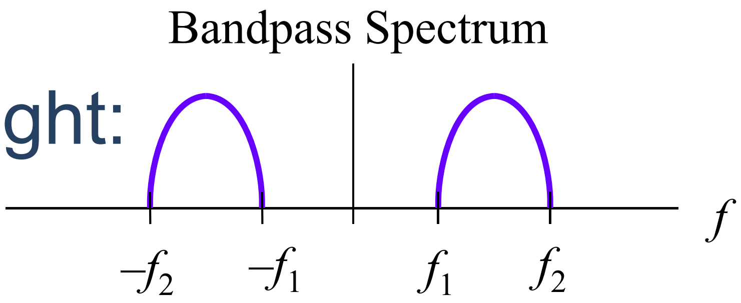


For higher order filters, the roll-off rate is -6n dB/octave, where n is the order of the filter.

Generalized Sampling Theorem

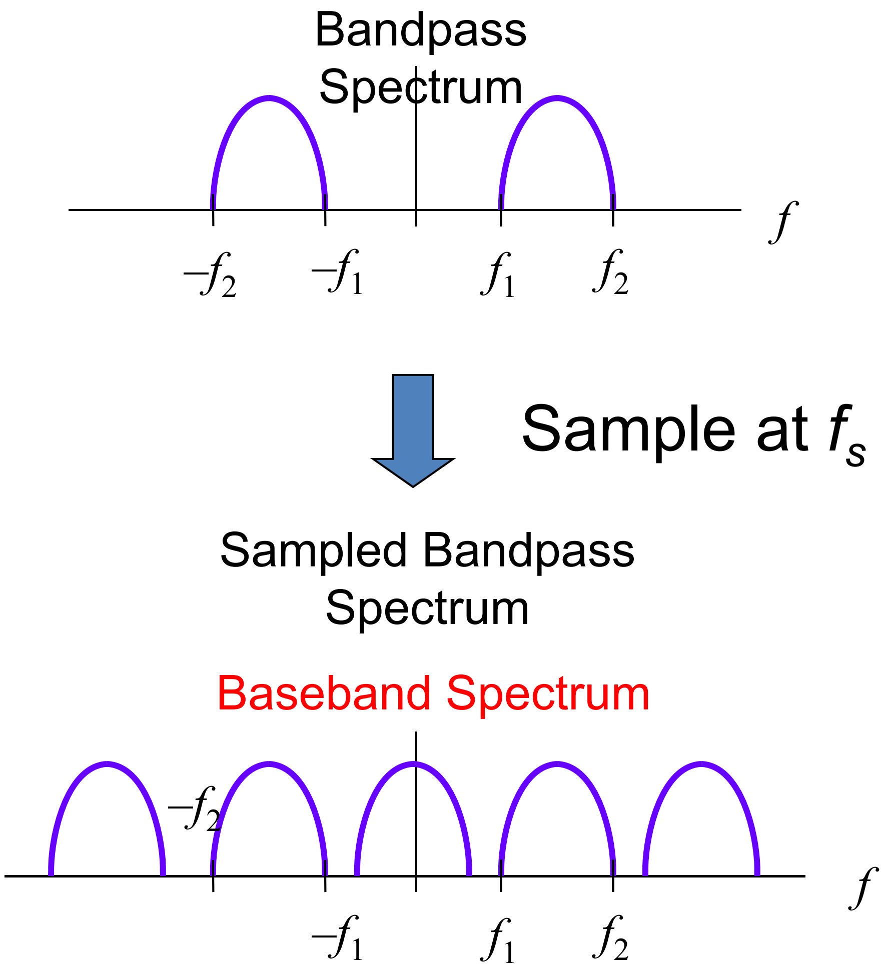
- Sampling rate must be greater than analog signal's bandwidth
  - Bandwidth is defined as non-zero extent of spectrum of the continuous-time signal in positive frequencies
  - Lowpass spectrum on right: bandwidth is f_{max}
  - Bandpass spectrum on right: bandwidth is $f_2 – f_1$





Bandpass Sampling

- Bandwidth: $\boldsymbol{f_2 - f_1}$
- Sampling rate $\boldsymbol{f_s}$ must greater than analog bandwidth
  $f_s > f_2 - f_1$
- For replicas of bands to be centered at origin after sampling
  $f_c = ½\,(f_1 + f_2) = k\,f_s$
- Lowpass filter to extract baseband



Bandpass Spectrum

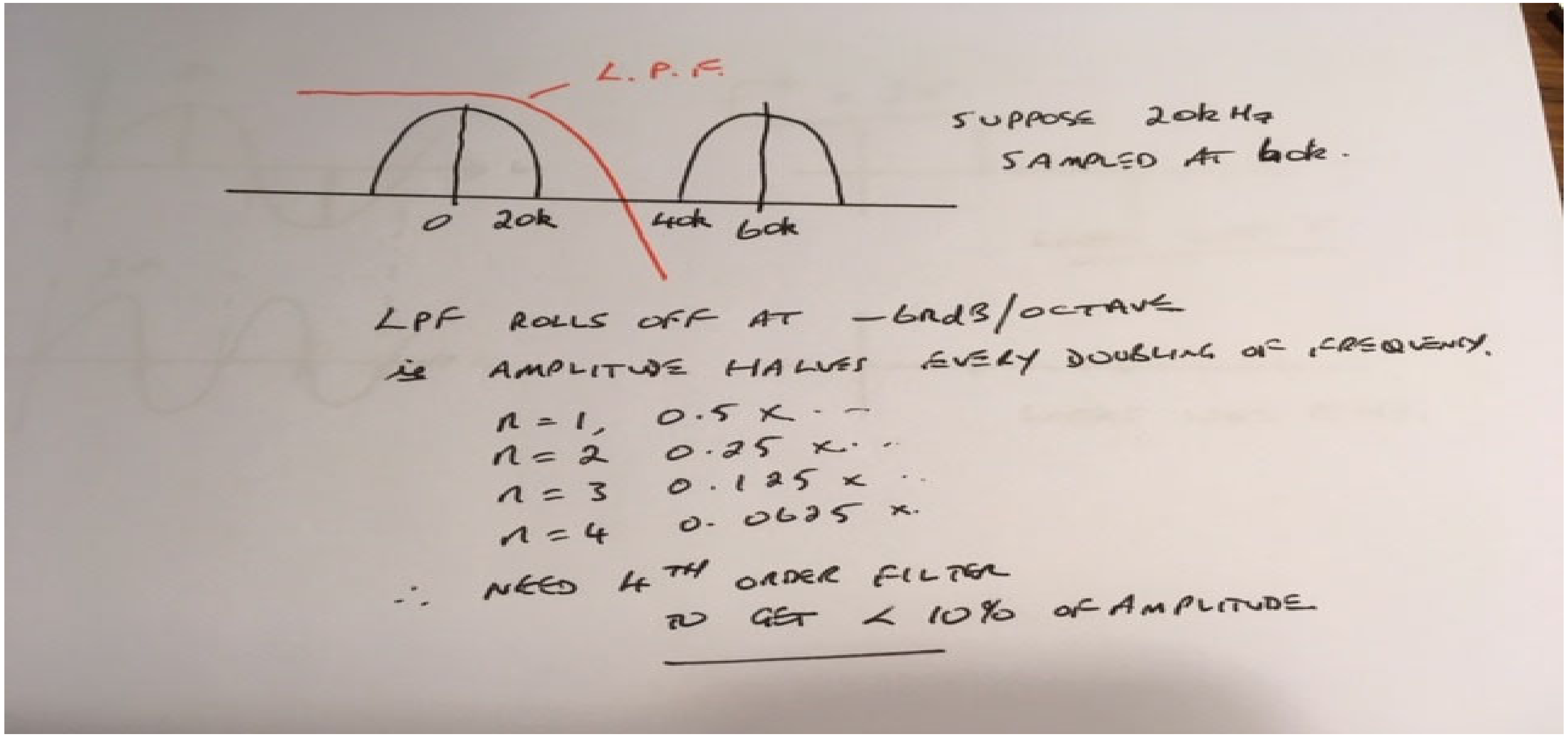
Sample at f_s

Sampled Bandpass Spectrum

Baseband Spectrum

Example of low-pass filter to extract baseband spectrum

Suppose we have an audio signal sampled at 60k/samples/sec and we want to reduce the bandpass amplitude to less than 10% of baseband amplitude



SUPPOSE 20k Hz
SAMPLED AT 60k.

LPF ROLLS OFF AT −6dB/OCTAVE
ie AMPLITUDE HALVES EVERY DOUBLING OF FREQUENCY.

n = 1, 0.5 x ..
n = 2 0.25 x ..
n = 3 0.125 x ..
n = 4 0.0625 x.

∴ NEED 4TH ORDER FILTER TO GET < 10% OF AMPLITUDE

Aliasing

1. Analog sinusoid

$$x(t) = A\cos(2p\ f_0\, t + \mathrm{f})$$

2. Sample at $\boldsymbol{T_s = 1/f_s}$

$$x[n] = x(T_s n) =$$

$$A\cos(2p\ f_0\ T_s\ n + \mathrm{f})$$

3. Keeping the sampling period same, sample

$$y(t) = A\cos(2p\ (f_0 + l\ f_s)\ t\ \ + \mathrm{f})$$

where ***l*** is an integer

4. $y[n]= y(T_s n)$

$$= A\cos(2p(f_0 + lf_s)T_s n + \mathrm{f})$$
$$= A\cos(2pf_0 T_s n + 2plf_s T_s n + \mathrm{f})$$
$$= A\cos(2pf_0 T_s n + 2pln + \mathrm{f})$$
$$= A\cos(2pf_0 T_s n + \mathrm{f})$$
$$= x[n]$$

Here, $f_s T_s = 1$

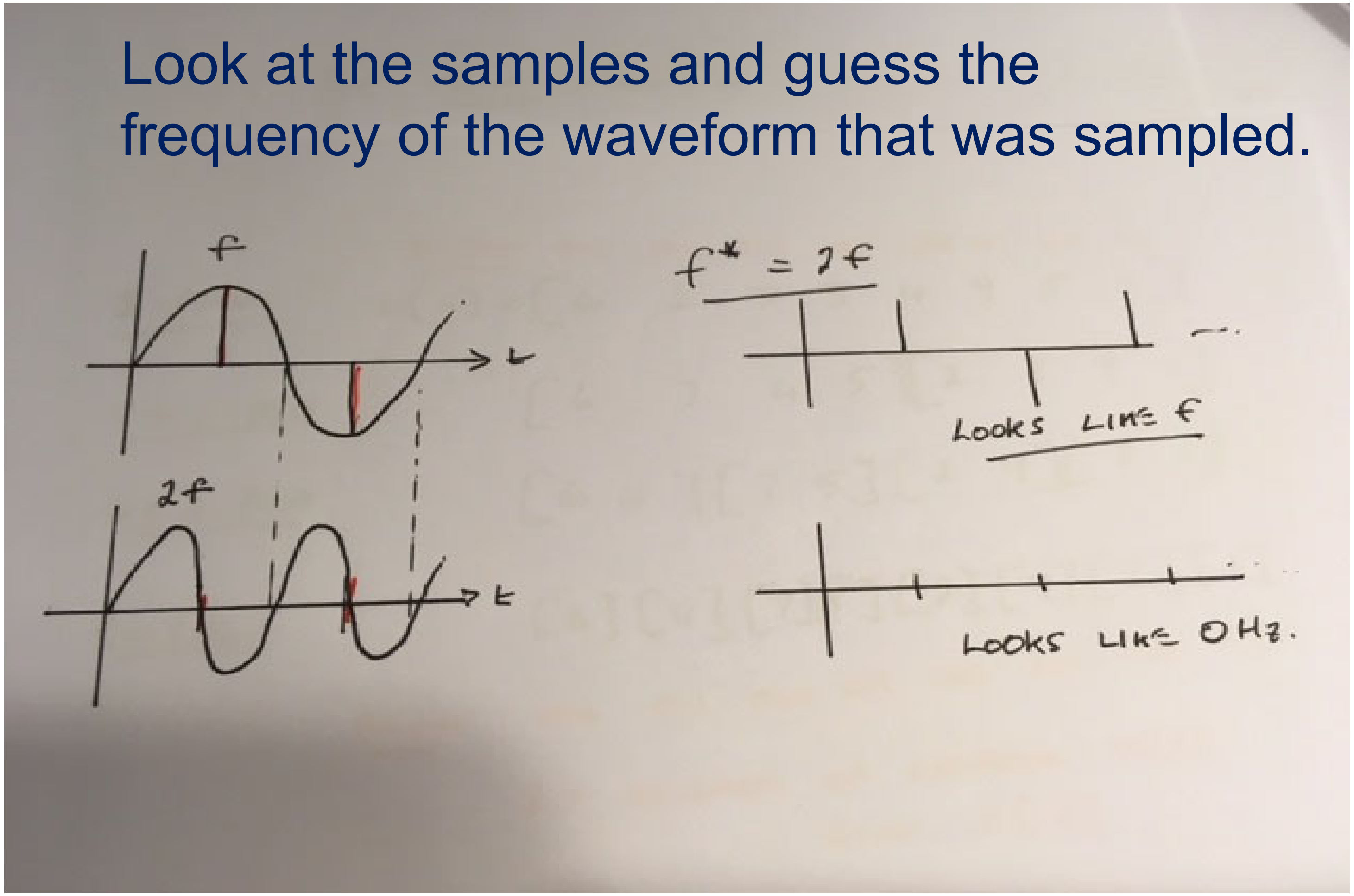
5. Since *l* is an integer,

$$\cos(x + 2\ p\ l) = \cos(x)$$

I is integer multiples of the sampling frequency

6. ***y*[*n*]** indistinguishable from ***x*[*n*]**

Simple example illustrating aliasing



Look at the samples and guess the frequency of the waveform that was sampled.

Aliasing

- Since ***l*** is any integer, a countable but infinite number of sinusoids will give same sequence of samples
- Frequencies $f_0 + l\, f_s$ for $l \neq \mathbf{0}$ are called aliases of frequency f_0 with respect to f_s

All aliased frequencies appear to be the same as f_0 when sampled by f_s

Folding

- Second source of aliasing frequencies
- From negative frequency component of a sinusoid, $-f_0 + l\, f_s$,

$$w(t) = A\cos(2\pi(-f_0 + l\, f_s)\, t - \phi)$$

where l is any integer

f_s is the sampling rate

f_0 is sinusoid frequency

- Sampling **$w(t)$** with a sampling period of **$T_s = 1/f_s$**

$$\begin{aligned}
w[n] &= w(T_s\, n) \\
&= A\cos(2\pi(-f_0 + l\, f_s)\, T_s\, n - \phi) \\
&= A\cos(-2\pi\, f_0\, T_s\, n + 2\pi\, l\, f_s\, T_s\, n - \phi) \\
&= A\cos(-2\pi\, f_0\, T_s\, n + 2\pi\, l\, n - \phi) \\
&= A\cos(-2\pi\, f_0\, T_s\, n - \phi) \\
&= A\cos(2\pi\, f_0\, T_s\, n + \phi)
\end{aligned}$$

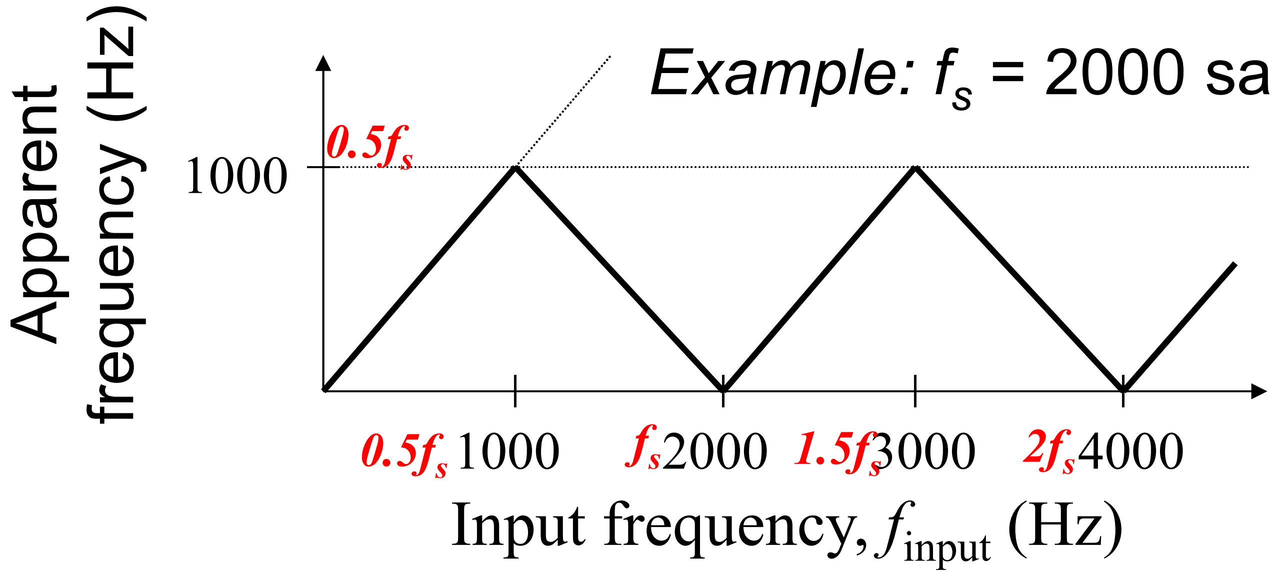
So

$w[n] = x[n] = x(T_s\, n)$

$x(t) = A\cos(2\pi\, f_0\, t + \phi)$

Aliasing and Folding

- Aliasing and folding of a sinusoid **sin(2 π f_{input} t)** sampled at **f_s = 2000** samples/s with **f_{input}** varied



- Mirror image effect about **f_{input} = ½ f_s** gives rise to name of folding

So if sampled at 2000 samples/s, 1000Hz appears as 1000Hz; 2000Hz appears as 0Hz; 3000Hz as 1000Hz; 3500Hz as 500Hz; etc.

Discrete-Time Fourier Transform

- Forward transform of discrete-time signal *x*[*n*]

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\,\omega\, n}$$

  - Assumes that *x*[*n*] is two-sided and infinite in duration
  - Produces *X*(w) that is continuous and periodic in w (in units of rad/sample) with period 2 π due to exponential term So the DTFT is not practical

- Inverse discrete-time Fourier transform

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$$

Discrete Fourier Transform

- Discrete Fourier transform (DFT) of a discrete-time signal *x*[*n*] with finite extent $n \in [0, N-1]$

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}nk} = X(\omega)\Big|_{\omega=\frac{2\pi}{N}k} \quad \text{for } k = 0,\ 1,\ \ldots,\ N\text{-}1$$

  - *X*[*k*] is periodic with period *N* due to exponential
  - DFT assumes *x*[*n*] is also periodic with period *N*

- Inverse discrete Fourier transform

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}nk}$$

- “twiddle factor”

$$W_N = e^{j\frac{2\pi}{N}} \Rightarrow x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{nk}$$

Discrete Fourier Transform (con’t)

- Forward transform

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{-nk}$$

for k = 0, 1, …, N-1

Exponent of W_N has period N

- Memory usage

$x[n]$: N complex words of RAM

$X[k]$: N complex words of RAM

W_N: N complex words of ROM

- Halve memory usage

Allow output array $X[k]$ to write over input array $x[n]$

Exploit twiddle factor symmetry

- Computation

N^2 complex multiplications

N (N –1) complex additions

N^2 integer multiplications

N^2 modulo indexes into lookup table of twiddle factors

- Inverse transform

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, W_N^{nk}$$

for n = 0, 1, …, N-1

Memory? Computation?

Watch out for sign of “twiddle factor” !!
Different texts use + or - !!

DFT in Practice

- The DFT is the only practical way to find the FT of a signal (any other version of the FT requires an infinite amount of calculation).
- The DFT assumes that the sequence to be transformed is periodic.
- Suppose we want to find the DFT of a piece of music.
- If we sample the whole piece of music we will find its accurate DFT.
- If we take a segment of the music, we can find the segment’s DFT.
- But the DFT assumes that the sequence is periodic and this is very unlikely to be the case for the music.
- So if we find the DFTs for each segment of the music, the result will not be an accurate FT of the music.
- Finding the DFT of the whole music is accurate but will take a long time because there are many samples.
- The shorter each segment (fewer samples) the faster the calculation but the less accurate the transform.

Example of trade-off between accuracy and time

Suppose we have a 5 minute piece of music

5 minutes = 300 seconds

If we want high quality, assume bandwidth = 20kHz

So minimum sampling rate is 40k samples/second

Therefore number of samples = 40000x300 = 12000000

If we perform a DFT of the complete piece of music the input sequence is a 12M-point sequence……BIG

DFT would be accurate but take a long time…..

So could segment the sequence, but lose accuracy.

FFT would help.