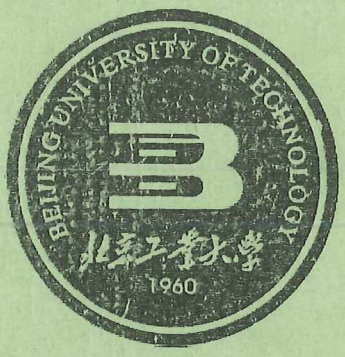
EEEN3004J:
Digital Signal Processing

Lecture Notes



Dr. John Healy

Prof. Anthony Fagan
Dr. Barry Cardiff

School of Electrical and Electronic Engineering,
University College Dublin

Contents

Acronyms

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DFT	Discrete Fourier Transform
DTFT	Discrete Time Fourier Transform
dB	Decibel
LTI	Linear Time Invariant
PDF	Probability Density Function
PSD	Power Spectral Density
RMS	Root Mean Squared
SNR	Signal to Noise Ratio
V	Short for Volts, the measurement unit for electric Voltage

Notation

Notation

Unless otherwise started, or obvious from the context, the following mathematical notation shall be used throughout.

f_s The sampling frequency in Hertz. $f_s = \frac{1}{T}$.

$f(t)$ Continuous time signal

f_n The n^{th} sample of time domain signal $f(f)$, i.e. $f_n = f(nT)$, and T is the sampling interval.

$\{f_n\}$ The (possibly infinite) collection of samples that comprise the entire signal, as opposed to f_n which is just a the n^{th} sample.

$\bar{f}(t)$ Discrete time version (impulse sampled) of $f(t)$, i.e.:

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

$\mathring{f}_n$ The n^{th} sample of the (infinitely long) periodically extended version of the finite length sequence $\{f_0 \ldots, f_n, \ldots, f_{N-1}\}$, i.e.:

$$\mathring{f}_{n+kN} = f_n \text{ for } \begin{matrix} 0 \leq n \leq N-1 \\ k = 0, \pm 1, \pm 2, \cdots \end{matrix}$$

$\bar{F}(j\omega)$ The DTFT of $f(t)$, i.e. the Fourier transform of $\bar{f}(t)$.

$\bar{F}_m$ The m^{th} sample of the DFT of the finite length N sequence $\{f_0 \ldots, f_n, \ldots, f_{N-1}\}$.

$$\bar{F}_m \triangleq \sum_{n=0}^{N-1} f_n e^{-j\frac{2\pi \cdot m \cdot n}{N}} \quad 0 \leq m \leq M-1$$

$\tilde{F}(z)$ The z-transform of $\{f_n\}$ $\tilde{F}(z) = \sum_{n=-\infty}^{+\infty} f_n z^{-n}$

m Often used as discrete frequency index

n Often used as discrete time index

T Often used as the sampling interval, $T = \frac{1}{f_s}$.

ω Angular frequency in units of radians per second. $\omega = 2\pi f$

ω_s Sampling angular frequency. $\omega_s = 2\pi f_s = \frac{2\pi}{T}$

$\star$ Convolution operator

$\circledast$ Circular convolution operator

Chapter 1

Course outline

1.1 Part I

1. Sampling Theorem
2. Aliasing
3. Discrete Time Fourier Transform (DTFT)
4. Discrete Fourier Transform (DFT)
5. Fast Fourier Transform (FFT)
6. Discrete Convolution (direct and periodic)
7. Parseval's Theorem
8. Spectrum Estimation (Welch's method)

This will be followed by a mid-term exam.

1.2 Part II

1. The z transform
2. Discrete Linear systems
   (Transfer functions, impulse response, pole/zero plots)
3. Digital Filters - Intro
4. Finite Impulse Response (FIR) digital filters
   (Phase linearity condition, design Methods)
5. Window design method of FIR Filters
6. Stability of Digital Filters
7. Infinite Impulse Response Digital Filters (design methods)

This will be followed by an end-of-term exam.

1.3 Grading

	Component	%
1.	Mid-term exam	15%
2.	Matlab assignments / Labs	15%
3.	End-of-term exam	70%

Table 1.1: Overall grade breakdown.

1.4 Recommended texts

There two main books I recommend:

1. "Digital Signal Processing", by John G. Proakis & Dimitris G. Manolakis, and
2. (advanced text) "Digital signal Analysis" by Samuel D. Stearns & Don R. Hush

The first book is by Proakis who is very well known in DSP and communications, I highly recommend this text for this module and as an every day DSP handbook. The other text is more technical, harder to follow, but arguably better in some detailed derivations. Material from both books was used in the development of this module.

Chapter 2

Impulse sampling and Discrete Signals

2.1 Continuous signals

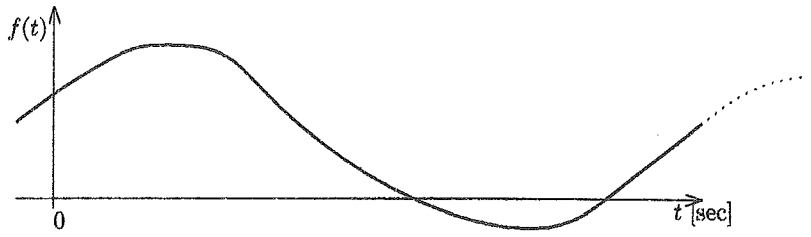


Figure 2.1.1: Continuous time analog signal

A *continuous signal* otherwise know as an *analog signal*, $f(t)$, like the one shown in Figure 2.1.1, has the following properties:

- Continuous in t
- It is defined for all t, i.e. $-\infty \rightarrow +\infty$
- Can obtain any amplitude value
- $f(t)$ could be a voltage, a current, an intensity etc.

The independent variable t could be time, distance or some other quantity.

$f(t)$ is usually real but can also be complex.

2.2 The Dirac delta function $\delta(t)$

2.2.1 Definition

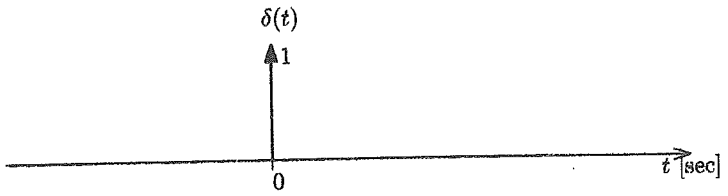


Figure 2.2.1: The Dirac delta function $\delta(t)$, note the weight is "1", i.e. $1.\delta(t)$

A *Dirac delta function*, $\delta(t)$, otherwise know as an *impulse* like the one shown in Figure 2.2.1, is defined by how it behaves within an integral:

$$\int_{\mathcal{D}} \delta(t)\,dt \triangleq \begin{cases} 1 & \text{if } \mathcal{D} \text{ contains orgin } (t=0) \\ 0 & \text{otherwise} \end{cases}$$

The first part says the area under the function in any arbitrarily small non-zero region about $t = 0$ is always $= 1$ irrespective of the size of the integration range;
i.e. we say $\delta(t)$ has *weight* 1 (Note it has infinite amplitude at $t = 0$).
The second part says that the delta function is zero everywhere except at $t = 0$ as shown in Figure 2.2.1.

A Dirac delta function is often simply referred to as an *impulse*.

We can think of an impulse as being an impossibility narrow rectangular pulse with unit area centered on $t = 0$:

$$\delta(t) = \lim_{\varepsilon \to 0} \{\text{rect}(\varepsilon; t)\}$$

$$\text{Where } \text{rect}(\varepsilon; t) \triangleq \begin{cases} \frac{1}{\varepsilon} & \text{if } |t| < \frac{\varepsilon}{2} \\ 0 & \text{otherwise} \end{cases}$$

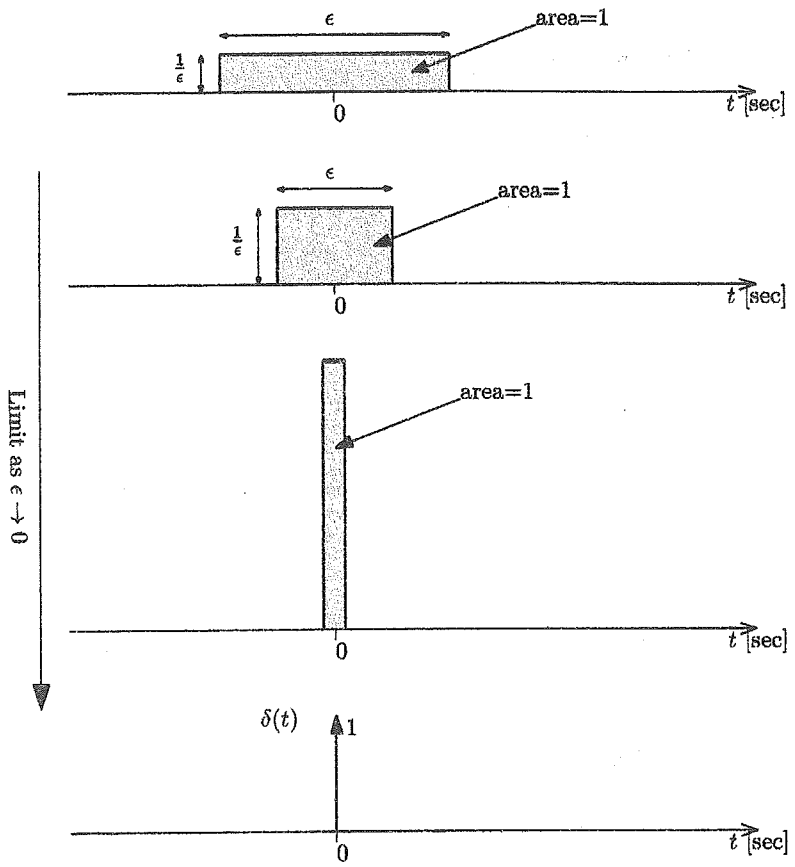


Figure 2.2.2: The Dirac delta function is the limit as $\epsilon \to 0$.

2.2.2 The delayed impulse

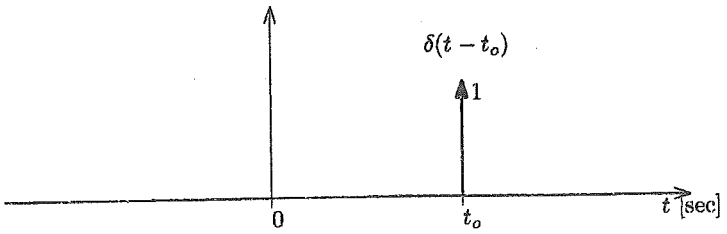


Figure 2.2.3: The delayed Dirac delta function delayed by t_o

An impulse, delayed in time by t_o is written $\delta(t-t_o)$ and is illustrated in Figure 2.2.3.

2.2.3 The weighted impulse

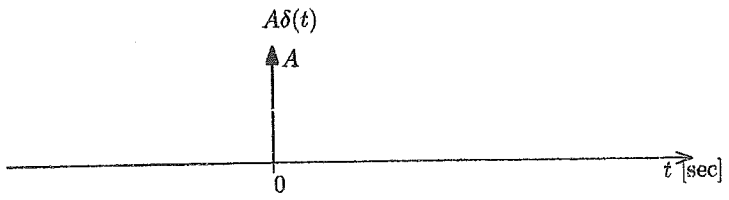


Figure 2.2.4: The delayed Dirac delta function weighted by A

An impulse with weight A is simply written $A\delta(t)$ and is illustrated in Figure 2.2.4.

2.2.4 Sifting property

Perhaps the most important property of the delta function is that of the sifting (or sampling) property. Simply put it states that integral of any function $f(t)$ times a delta function $\delta(t-t_o)$ is just just the original function evaluated at $t=t_o$, i.e.:

$$\int_D f(t)\,\delta(t-t_o)\,dt \;=\; f(t_o)$$

provided the integration range D includes t_o.

Note: Some texts actually take this as the definition of the delta function, and prove everything else from that starting point, i.e. it's the delta function's behavior in an integral that makes it so important.

2.2.5 Scaling property

Care must be taking when changing variables involving the delta function, we have the following property:

$$\delta(\lambda x) = \frac{1}{|\lambda|}\delta(x)$$

for any real valued scalar λ.

This is particularly an issue when we have a delta function in the frequency domain f and we do a change of variable to $\omega = 2\pi f$, in this case we have:

$$\delta(f) = 2\pi\delta(\omega)$$

As a consequence, we note the following Fourier transform pairs:

$$\begin{aligned} e^{j2\pi f_o t} &\leftarrow \mathcal{F} \rightarrow \delta(f - f_o) \\ &\text{and} \\ e^{j\omega_o t} &\leftarrow \mathcal{F} \rightarrow 2\pi\delta(\omega - \omega_o) \end{aligned} \tag{2.2.1}$$

Exercise:

Prove the scaling property. (Hint: start by computing $\int_{-\infty}^{+\infty} f(x)\,\delta(\lambda x)\,dx$).

Solution:

2.2.6 The impulse train $\delta_T(t)$

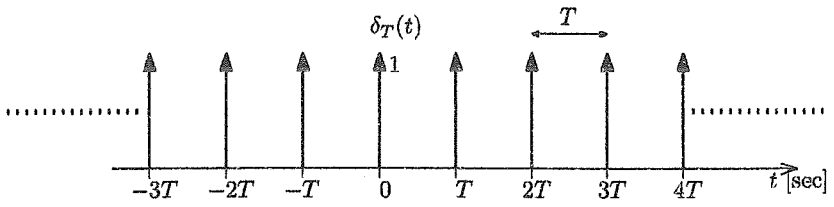


Figure 2.2.5: A train (sequence) of uniformly spaced impulses.

A train (sequence) of uniformly spaced impulses, $\delta_T(t)$, is illustrated in Figure 2.2.5 and defined as:

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

2.2.7 Fourier of series $\delta_T(t)$

We will use the delta train to sample an analog signal, so we will call T the sampling interval (in time), and $f_s \triangleq \frac{1}{T}$ the sampling frequency (units = Hertz).

As $\delta_T(t)$ is a periodic function with frequency f_s, it can be written as a Fourier Series, i.e. an infinite weighted sum of complex sinusoids having frequencies equal to integer multiples of f_s:

$$\delta_T(t) = \sum_{n=-\infty}^{+\infty} C_n e^{jn2\pi f_s t} = \sum_{n=-\infty}^{+\infty} C_n e^{jn\omega_s t} \tag{2.2.2}$$

Where we define (for convenience) $\omega_s \triangleq 2\pi f_s$, meaning the angular sampling frequency (units = rads/sec). We will see later that DSP engineers, for good reason, don't like Hz as a measure of frequency, but rather they often use this angular frequency instead.
The C_n are the Fourier coefficients which is given by:

$$C_n = \frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} \delta_T(t)\, e^{-jn\omega_s t} dt \quad n = -\infty, \cdots, 0, \cdots, +\infty$$

But as the range of integration only contains one delta function (see Figure 2.2.6), we can replace the train $\delta_T(t)$ by just $\delta(t)$, yielding:

$$C_n = \frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} \delta(t)\, e^{-jn\omega_s t} dt \quad n = -\infty, \cdots, 0, \cdots, +\infty$$

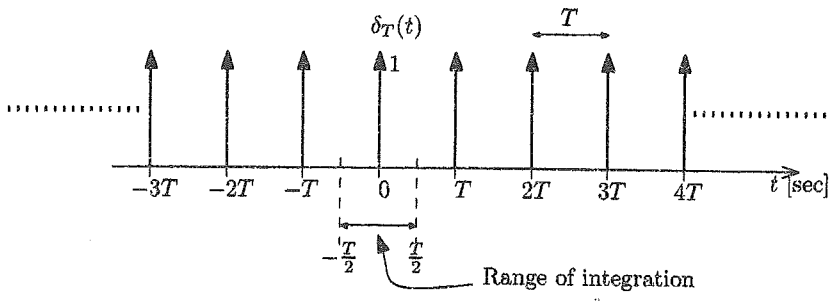


Figure 2.2.6: Range of integration in the Fourier coefficients computation.

but by the sifting property of the delta function (see Section 2.2.4) we have:

$$C_n = \frac{1}{T}e^{-jn\omega_s 0} = \frac{1}{T} \quad \forall n$$

i.e. all the Fourier coefficients have the same value $\frac{1}{T}$.

Putting this into equation 2.2.2 gives us the Fourier series of the impulse train $\delta_T(t)$:

$$\boxed{\delta_T(t) = \frac{1}{T}\sum_{n=-\infty}^{+\infty} e^{jn\omega_s t}} \tag{2.2.3}$$

But we know from equation (2.2.1) that a complex exponential in the time domain is a delta function in the frequency domain, so we have the following Fourier transform:

$$\begin{aligned} \delta_T(t) &\leftarrow \mathcal{F} \rightarrow \frac{1}{T}\delta_{f_s}(f) = f_s\delta_{f_s}(f) \\ &\text{or} \\ \delta_T(t) &\leftarrow \mathcal{F} \rightarrow \frac{2\pi}{T}\delta_{\omega_s}(\omega) = \omega_s\delta_{\omega_s}(\omega) \end{aligned} \tag{2.2.4}$$

This is illustrated in Figure (2.2.7)

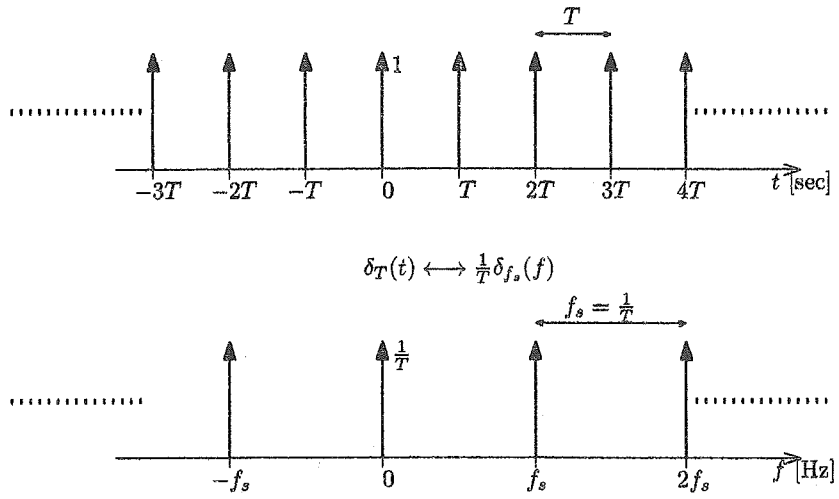


Figure 2.2.7: Fourier Transform of a train of delta function is another train of delta functions with inverse spacing and weight $\frac{1}{T} = f_s$. (Note that if plotted against ω instead the weights would be $\frac{2\pi}{T} = \omega_s$).

2.3 Discrete Signals

A *discrete* (not discreet!) signal can be obtained by multiplying an analog signal by train of impulses (impulse sampling) as shown in Figure 2.3.1.

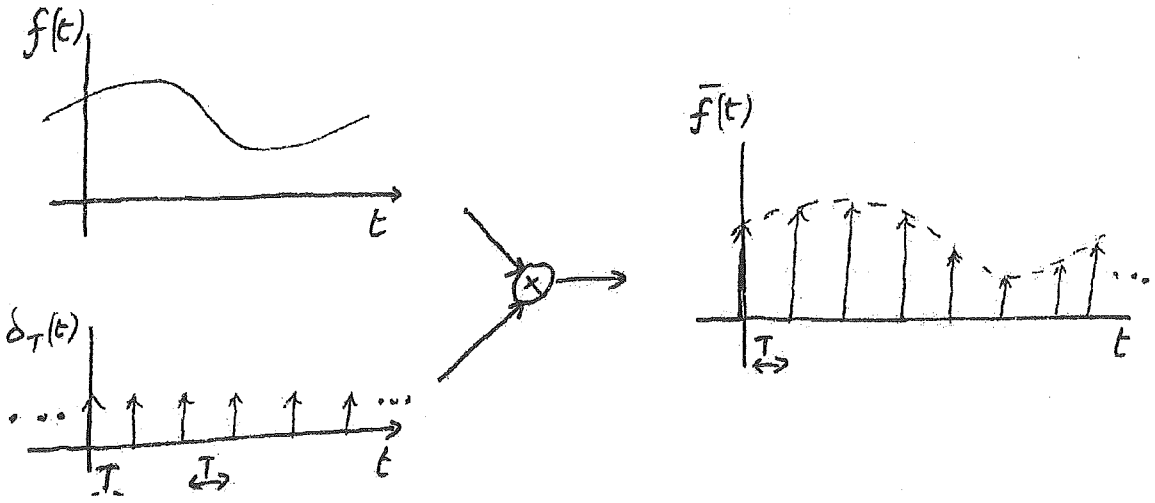


Figure 2.3.1: Impulse sampling.

We have:

$$\begin{aligned}\bar{f}(t) &\triangleq f(t)\,\delta_T(t) \\ &= f(t)\sum_{n=-\infty}^{+\infty}\delta(t-nT) \\ &= \sum_{n=-\infty}^{+\infty} f(nT)\,\delta(t-nT)\end{aligned}$$

Where T is sampling Interval (seconds).

The Sampling frequency is $f_s = \frac{1}{T}$Hz.

Notation:

It is usual to use the following notation

$$f(t)|_{t=nT} = f(nT) \triangleq f_n$$

Here f_n represent the sample at ANY time instant $t = nT$. We use the notation $\{f_n\}$ to represent the complete set of samples, i.e. ALL samples.

Using this notation we have:

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

2.3.1 Properties of $\bar{f}(t)$

$\bar{f}(t)$ has the following basic properties:

- $\bar{f}(t)$ is a train of impulses, defined for all t.
- the weights f_n are the samples of the signal taken at time $t = nT$
- the amplitudes $\bar{f}(nT)$ are all infinite (except for the case where weight is zero).

Sometimes we are only interested in the values at the sampling instances, then we plot the signal like this *stem diagram*:

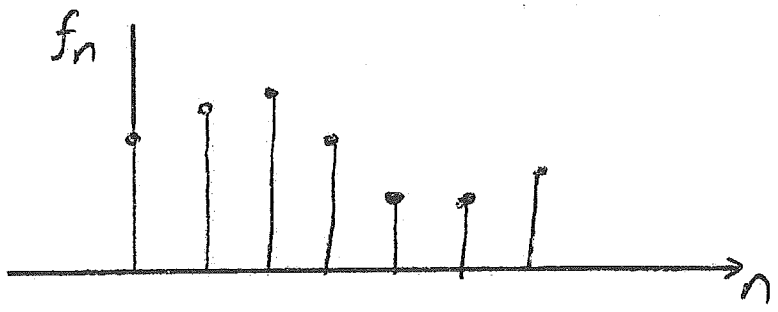


Figure 2.3.2: Illustration of a sampled signal using a stem diagram.

Discrete signals can be real or complex. Discrete signals can be multidimensional, e.g. a still image is 2-D (x and y), video is 3-D (time dimension added).

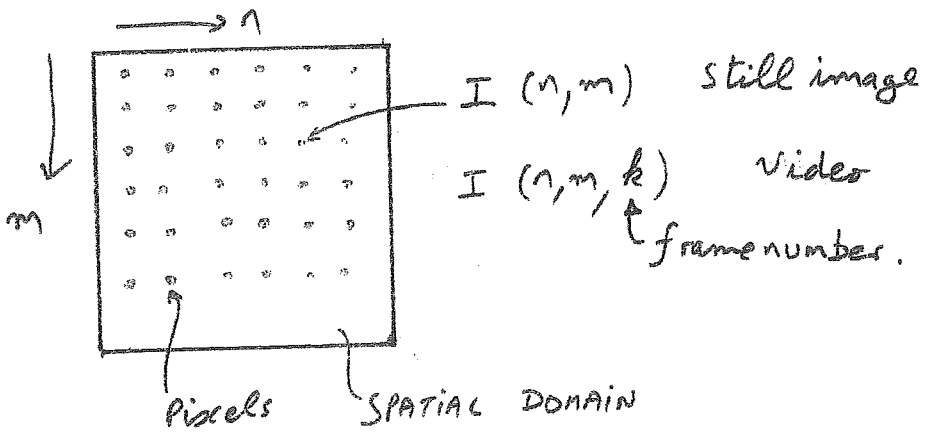


Figure 2.3.3: Illustration of 2D sampling (in the spatial domain)

If a sequence (discrete signal), f_n, is represented by numbers stored in a machine then we have a *digital signal*.

(Discrete signals can be represented in other ways, e.g. by charges on a collection of capacitors).

Chapter 3

Nyquist's Sampling Theorem

3.1 Introduction

An important question in digital systems is 'What is the correct sampling rate?'

- How well represented is an analog signal by its samples?
- In other words, how accurately could we reconstruct a signal from its samples?
- Intuitively we might feel that the sampling interval, $T \to 0$, in order to have perfect reconstruction.
- However, to aid storage / transmission of digital data we would like to use the smallest number of samples possible, i.e. a large T

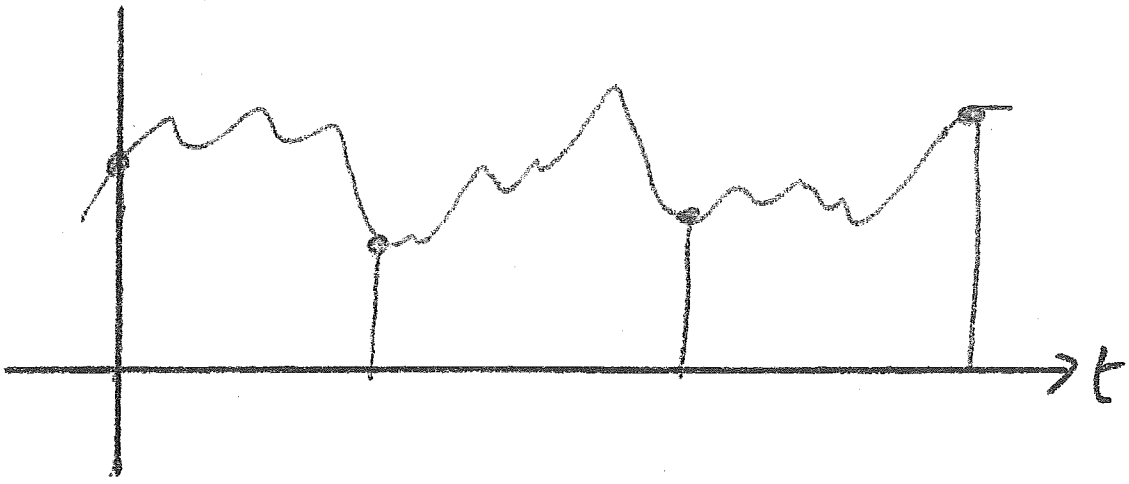


Figure 3.1.1: Signal changes quickly $\Rightarrow$ is the sample rate too low?

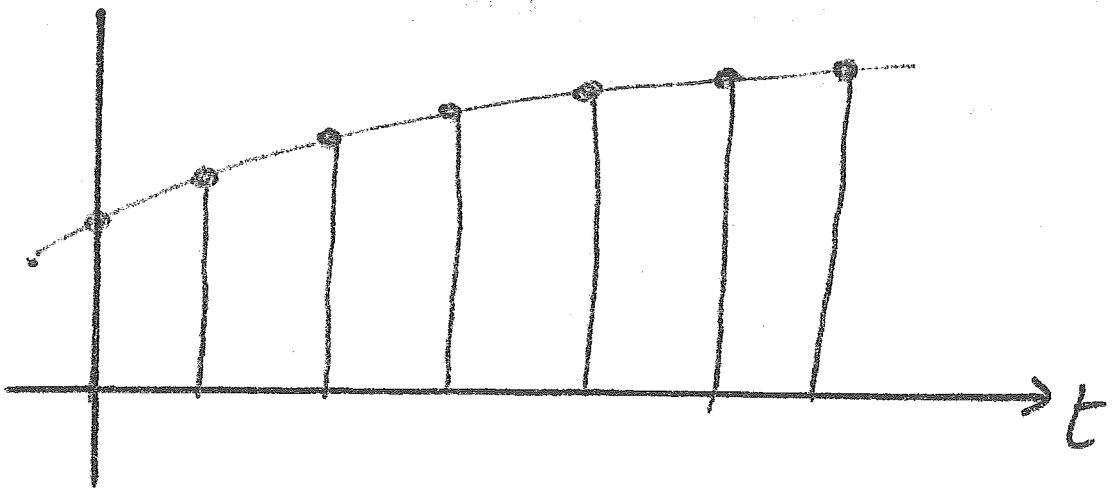


Figure 3.1.2: Signal changes slowly ⇒is the sample rate too high?

We have a problem... Nyquist's sampling theorem comes to rescue!

3.2 Nyquist's sampling theorem

We start by deriving the spectrum of a sampled signal and then graphically derive the sampling theorem.

3.2.1 Spectrum of sampled signal

Consider a signal, $f(t)$, that is sampled by a train of impulses:

$$\bar{f}(t) = f(t)\,\delta_T(t) \tag{3.2.1}$$

But we know, from equation (2.2.3), the Fourier series of an impulse train, which by substitution yields:

$$\bar{f}(t) = \frac{1}{T} f(t) \sum_{n=-\infty}^{+\infty} e^{jn\omega_s t}$$

Taking the Fourier transform of both sides:

$$\begin{aligned}
\mathcal{F}\left[\bar{f}(t)\right] &= \frac{1}{T}\mathcal{F}\left[f(t)\sum_{n=-\infty}^{+\infty} e^{jn\omega_s t}\right] \\
\Rightarrow \bar{F}(j\omega) &= \frac{1}{T}\sum_{n=-\infty}^{+\infty}\mathcal{F}\left[f(t)\,e^{jn\omega_s t}\right] \\
&= \frac{1}{T}\sum_{n=-\infty}^{+\infty} F(j(\omega - n\omega_s))
\end{aligned}$$

(where ω is the angular frequency in units of rads/sec).

In the last step we used the shift theorem[1].

In summary, the spectrum of the sampled signal is:

$$\bar{F}(j\omega) = \frac{1}{T}\sum_{n=-\infty}^{+\infty} F(j(\omega - n\omega_s))$$

Where:

$F(j\omega)$ = complex spectrum of the original analog signal.

$\bar{F}(j\omega)$ = complex spectrum of the sampled signal (sampled by impulses).

We see that when a signal is sampled by impulses, two things happen the spectrum:

1. The spectrum is scaled by a factor $\frac{1}{T}$
2. An infinite number of copies of the (scaled) original spectrum, each shifted by a unique integer multiple of the sampling frequency, are added together.

[1]Shift Theorem states that if $G(f)$ is the Fourier transform of $g(t)$, then multiplying (in the time domain) by a complex exponential, i.e. $g(t)\,e^{-j2\pi f_o t}$ results in a shifted Fourier transform $G(f - f_o)$.

3.2.2 Graphical development of the sampling theorem

We will consider two cases:

- band limited signals, and
- non-band limited signals

3.2.2.1 Band-limited signal case

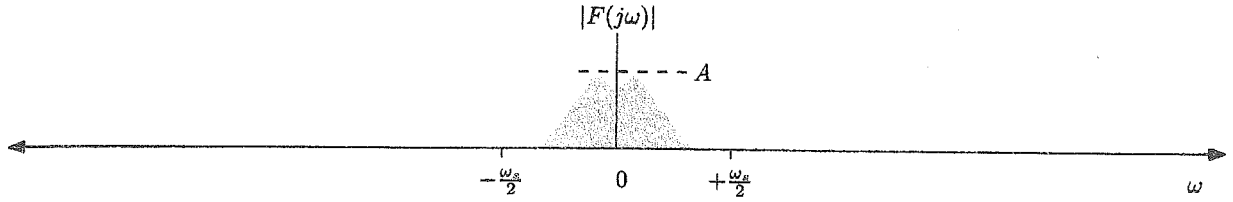


Figure 3.2.1: Spectrum of a band-limited analog signal.

Here the spectrum is strictly band limited to within the range

$$-\frac{\omega_s}{2} < \omega < +\frac{\omega_s}{2}$$

($\frac{\omega_s}{2}$ is the half angular sampling frequency)

So $\left|\bar{F}\left(j\omega\right)\right|$ looks like this:

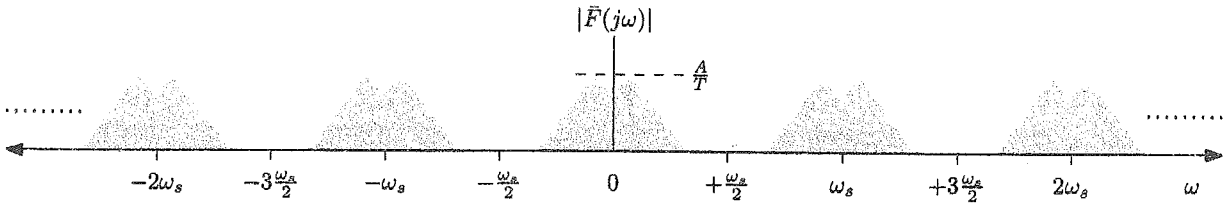


Figure 3.2.2: Spectrum of a band-limited signal after sampling.

We see that the spectrum of the original signal exists unchanged (apart from scaling) centered on $\omega = 0$.

So we *can*, at least in theory, have perfect reconstruction, i.e. it *should* be possible reconstruct the original signal by low pass filtering.

3.2.2.2 Perfect Reconstruction

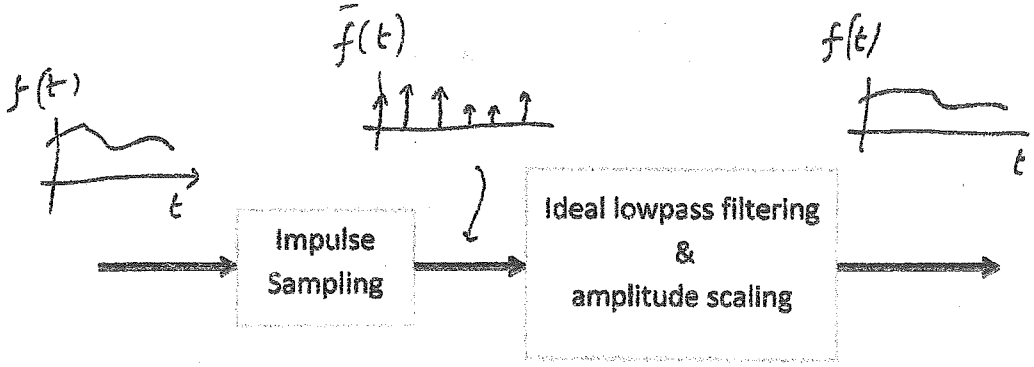


Figure 3.2.3: Perfect Reconstruction of an analog signal from its samples using a low pass filter.

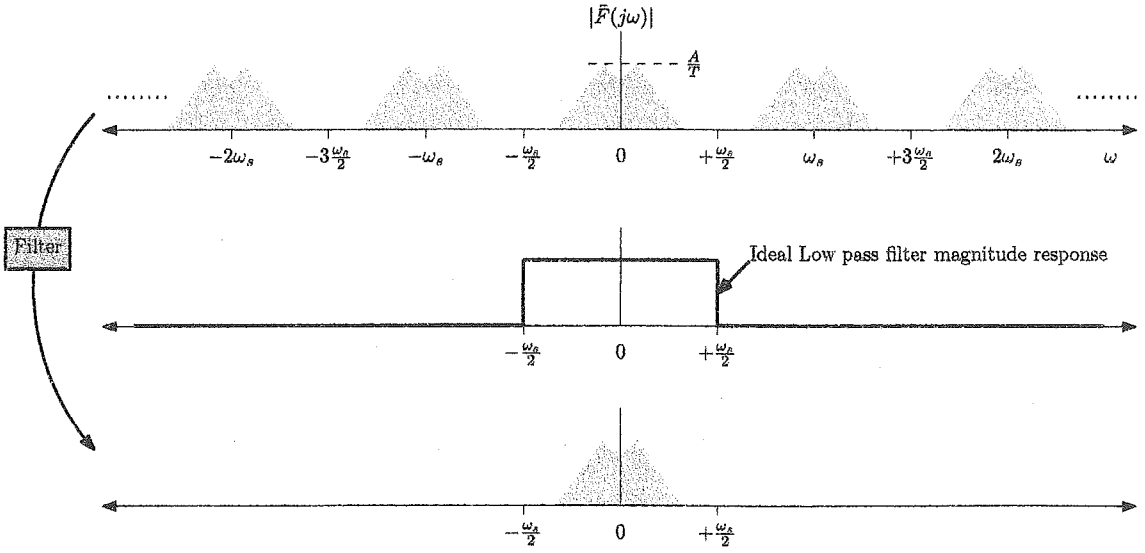


Figure 3.2.4: Perfect Reconstruction of an analog signal from its samples using a low pass filter.

3.2.2.3 non-Band-limited signal case

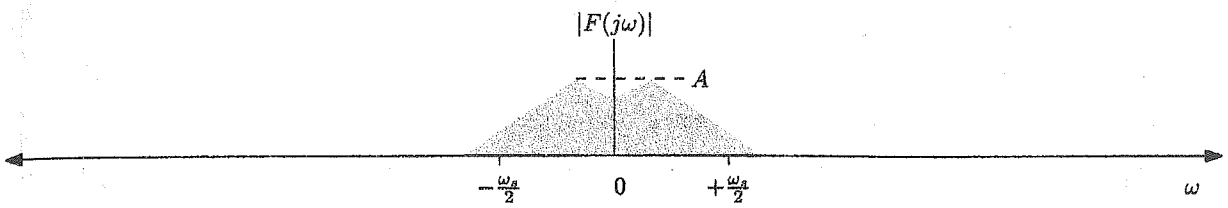


Figure 3.2.5: Spectrum of a non-band-limited analog signal.

The spectrum of the original analog signal is NOT band limited to within:

$$-\frac{\omega_s}{2} < \omega < +\frac{\omega_s}{2}$$

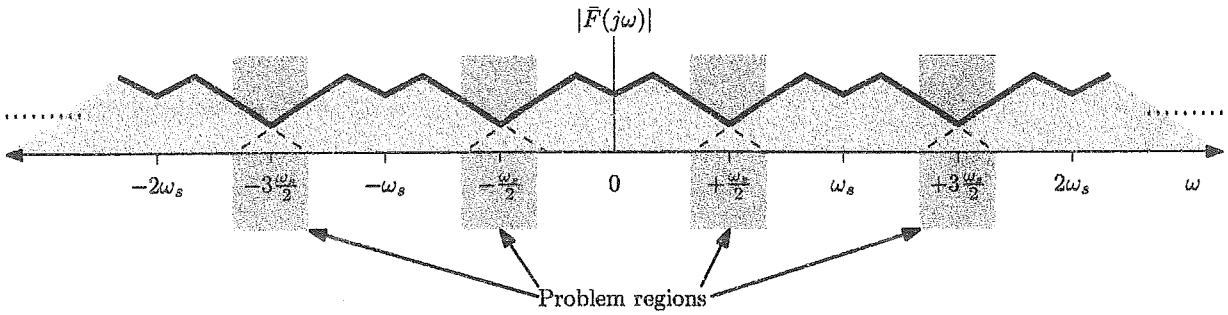


Figure 3.2.6: Spectrum of a non-band-limited signal after sampling.

At frequencies where the shifted spectra overlap, they add and there is a consequent loss of information and so the signal cannot be reconstructed from its samples.

3.3 In summary

The *Nyquist Sampling Theorem* states that a signal can be exactly reconstructed from its uniformly spaced samples if and only if the sampling frequency, $f_s = \frac{1}{T}$, is greater than twice the highest frequency component of the signal.
The reconstruction is done using an ideal low-pass filter with cut-off frequency at $\frac{1}{2}f_s$.

Aside: The above statement of the sampling theorem is for low-pass signals, there is a more general form for band-pass signals. (not done here).

Chapter 4

Aliasing, Sampling & Reconstruction

4.1 Aliasing

Alias:

1. An assumed name
2. (computers) An alternate name or address, especially an e-mail address that forwards incoming e-mail to another address.
3. (signal processing) An spurious signal generated as a technological artifact.

From Nyquist's sampling theorem, we know that when we sample an analog signal $f(t)$ at a sampling rate of f_s samples per second ($= \frac{\omega_s}{2\pi}$), we should ensure that all components of the signal lie within the range $\pm\frac{1}{2}f_s$ Hz.

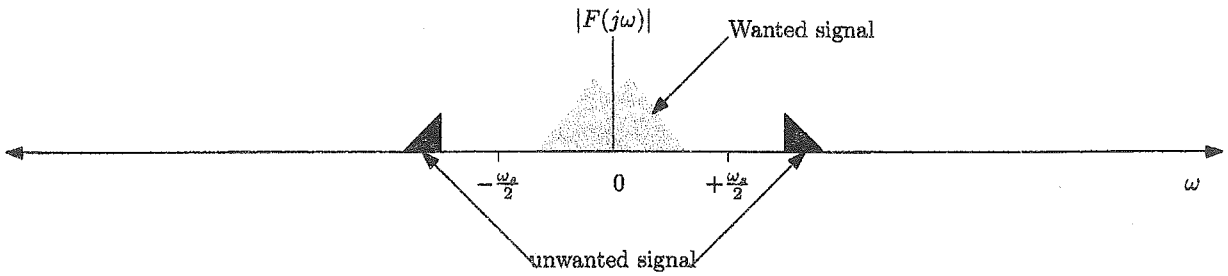


Figure 4.1.1: Analog signal before samping with some unwanted components outside of the allowable frequency range (for sampling at a rate of f_s Hertz).

If there are components that lie outside of that range, e.g. see Figure 4.1.1, they will appear

within that range after sampling - a process known as *aliasing* or *folding*. This process is illustrated in Figure 4.1.2.

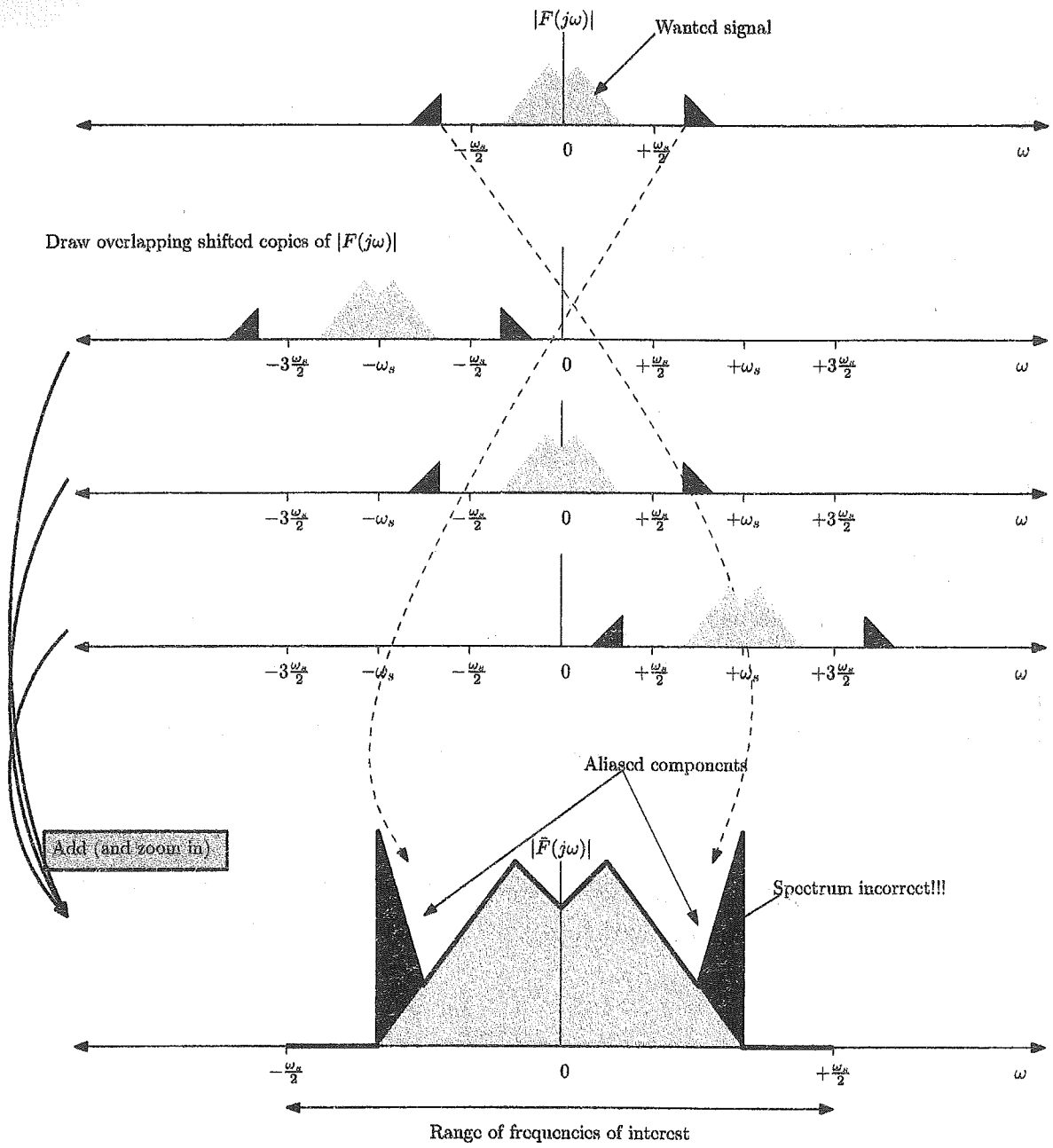


Figure 4.1.2: Spectrum after sampling. The red parts have now appear to have a lower frequency than before - i.e. they are *aliased*.

Exercise:

Try to get the same result using the frequency domains plot as before.

4.1.1 Folding - another name for aliasing

We can think of the process of adding shifted version of the frequency spectrum together as being like folding the spectrum on top of itself as illustrated below:

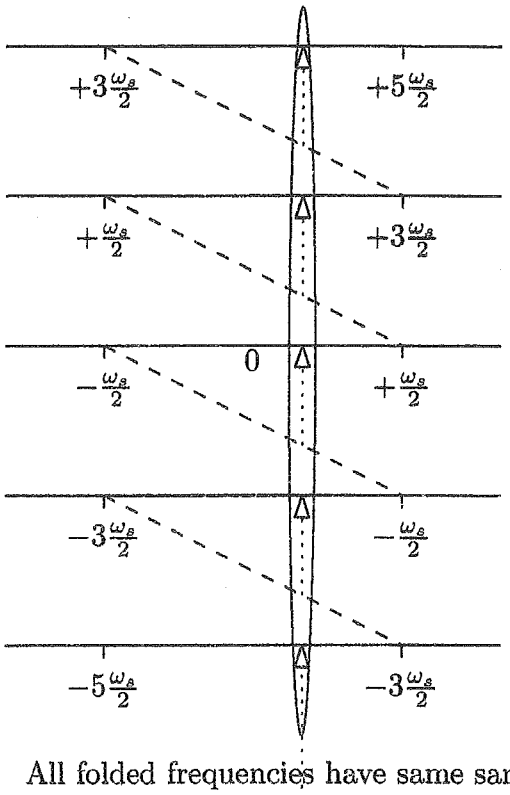


Figure 4.1.3: Spectrum folding after sampling. If we fold the analog spectrum in this way all signal components the "line-up" vertically will appear to have the same frequency after sampling.

4.1.2 Sine wave example

Consider an analog sine wave signal having frequency f_o, i.e. $f(t) = \sin(2\pi f_o t)$. Imagine it is sampled at rate f_s samples per second resulting in the stream of sample values, $f_n \triangleq f(nT)$, where $T = \frac{1}{f_s}$ is the time interval between samples.

Show that:

$$f_n = \sin(2\pi(f_o + mf_s)nT) \quad \forall m \in \mathbb{Z}$$

Solution:

The importance of this is that if the frequency of the sine wave is f_o,or $f_o \pm f_s$, or $f_o \pm 2f_s$, or $f_o \pm 3f_s$ etc..., the values of the samples f_n are going to be the same!

This is a time domain example of how all the frequencies $f_o + mf_s$ for $\forall\, m \in \mathbb{Z}$ alias back to f_o. This is illustrated in Figure 4.1.4.

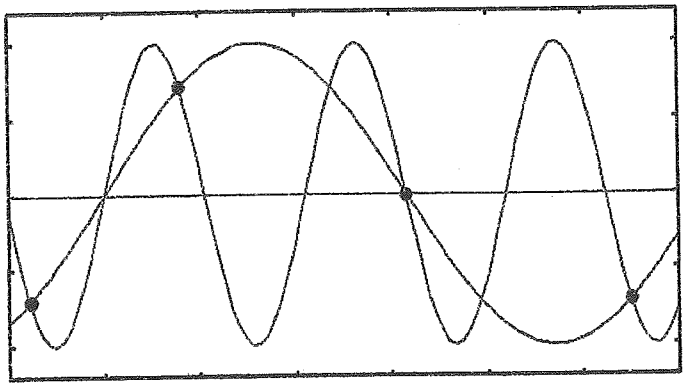


Figure 4.1.4: Two analog sine waves having frequency f_o(blue) and $f_o + 3f_s$ (red) both have the same sampled values (the dots) when sampled at a rate of f_m samples/sec.

4.2 Sampling in practice

4.2.1 Anti-Aliasing filter

As per Nyquist's sampling theorem we know that the signal presented to an Analog-to-Digital (ADC) must not contain frequency components beyond $\pm\frac{1}{2}f_s$. Sometimes (but rarely) this is the case naturally, but usually we need to do some analog signal processing, e.g. filtering, to ensure that this is (at least approximately) true. Specifically there is almost always an analog anti-aliasing filter positioned prior to the ADC.

Another reasons is that in many systems noise is present and it can have a much larger bandwidth than the wanted signals - this noise will also fold during the sampling process and if not removed it can dominate and the signal can be destroyed forever => we almost always have an anti-aliasing filter.

The ideal anti-aliasing filter has the following characteristics:

1. Passes all frequencies between $\pm\frac{1}{2}f_s$ Hertz (gain 0dB)
2. Does not distort the signals passed - just a delay
3. All other frequencies are prevented from passing.

The type of filter is known as a "*brick-wall*" specification and is theoretically impossible!!!

Real-world low-pass filters have the following characteristics:

1. This is some pass-band distortion
2. The stop-band does not attenuate the signals to zero completely
3. There is a transition band between the pass and stop bands

See 4.2.1 for a typical low pass filter specification.

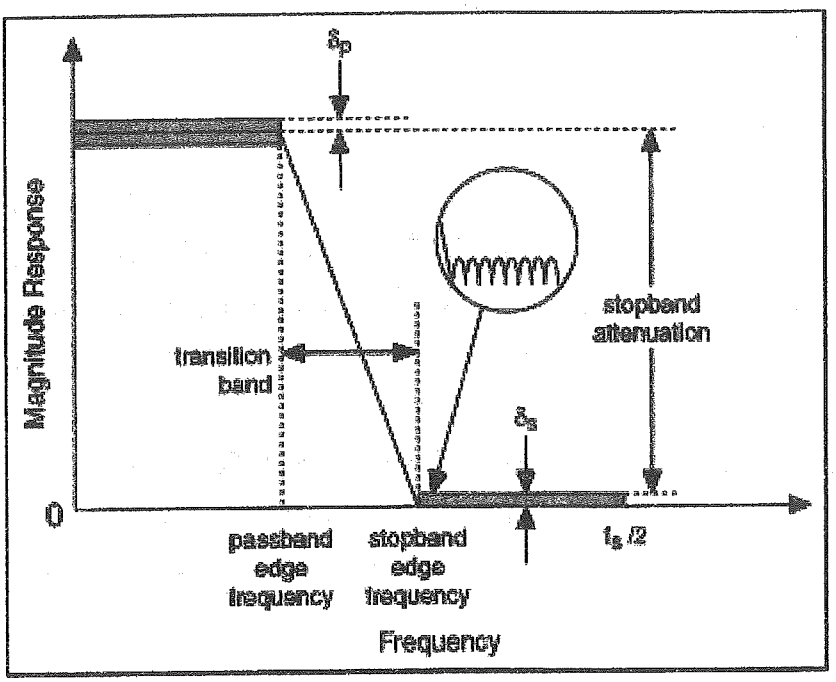


Figure 4.2.1: A real world low-pass filter specification.

Q. So what can we do?

A. Oversample.

4.2.1.1 Oversampling

In theory a signal having its highest wanted frequency component at f_{max} (plus some other unwanted components) can be filtered by a brick-wall filter having cut-off frequency f_{max} and then sampled at a rate of $2 \times f_{max}$.

In practice, as brick-wall filters don't exist, we use a real world low pass filter with a pass-band edge frequency > f_{max}, and we sample at a rate $f_s > 2 \times f_{\text{stopband}}$, where f_{stopband} is the stop-band edge frequency ($> f_{max}$). This way the signal will be faithfully captured as too will some of the unwanted signal components - but importantly they will not interfere with the wanted parts.

Note that the sampling here is now $> 2 \times f_{max}$, hence the name *oversampling*.

See Figure 4.2.2 for an example. It is important to note that the sampling process in this case converts both wanted and unwanted signals but this is done in such a way that they don't interfere with each other. Perhaps the subsequent digital processing will further "clear-up" the signal.

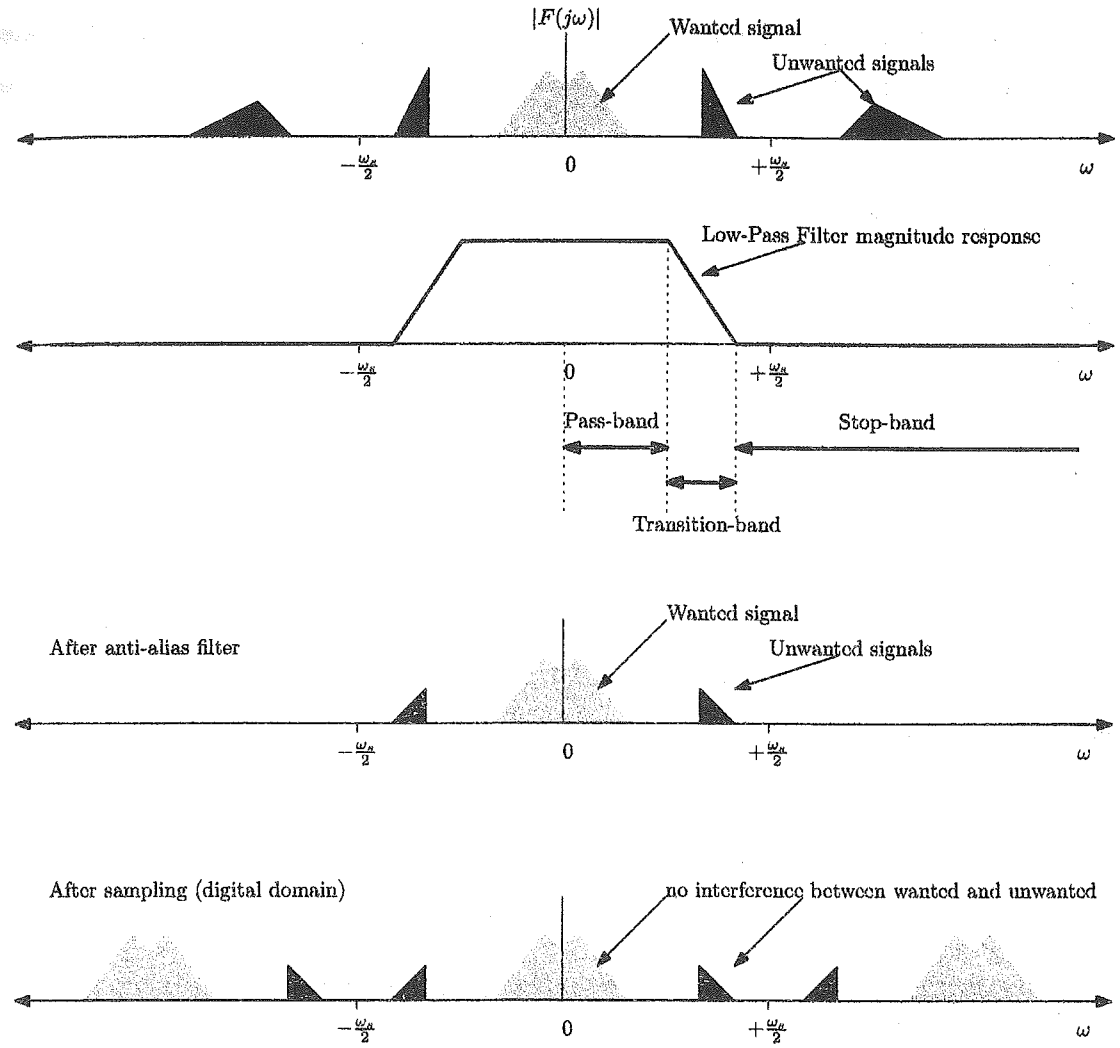


Figure 4.2.2: Real world sampling using a non-ideal low-pass anti-aliasing filter.

Note:

Analog filters can be big and bulky and can consume a lot of power, and the tighter the specification the bigger they become => we usually want to design small easy-to-implement anti-aliasing filters. This can be done by increasing the sampling rate higher, i.e. by over-sampling by a large factor. Today it is not uncommon to sample at $10 \times f_{max}$ instead of the $2 \times f_{max}$ that Nyquist would suggest.

4.2.2 Sample & Hold

Impulse sampling is NOT possible as it requires the generation of Dirac delta functions which are not possible[1].

[1] Not possible due to their infinite amplitude, and infinitesimally small duration.

We can however implement circuits (using transistors and capacitors) that can approximate well a "sample & hold" (S/H) function resulting in a waveform like the one shown in Figure 4.2.3.

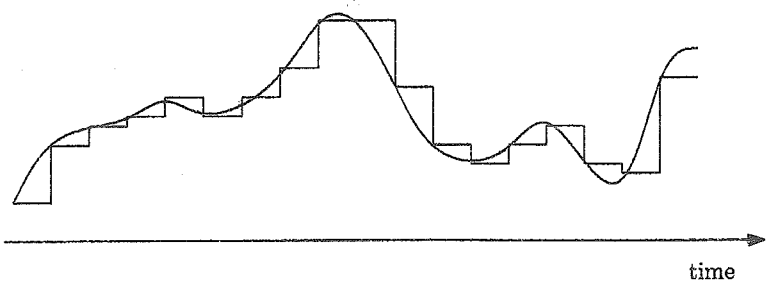


Figure 4.2.3: Analog signal input (smooth line) to a S/H device and the corresponding output (red discontinuous line).

If we follow this S/H operation with a device that converts the analog voltage in each time step to a numeric value (the core of the ADC), then we'll have a series of numbers which represent the value of the signal at the sample instances. For this to work correctly the following need to be true:

1. The value held on the S/H output must represent the voltage at the exact sampling time instant
2. The value held must be stable for the duration of the conversion

It is clear that the sampling rate in this scheme is limited by the speed of the conversion.

4.2.2.1 ADC chain

Based on the above discussions we can now draw the entire ADC chain as per Figure 4.2.4.

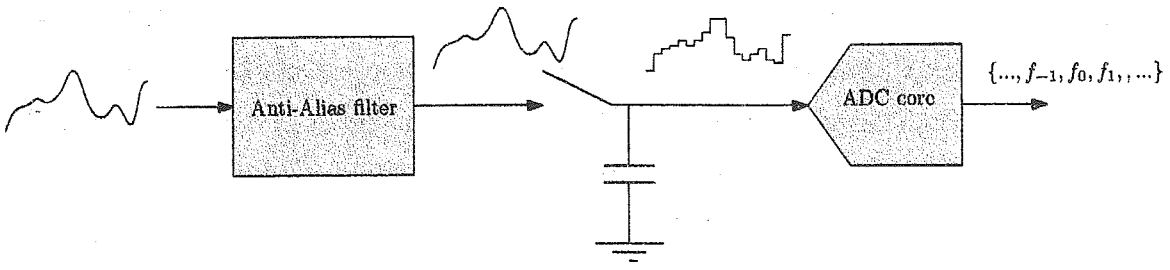


Figure 4.2.4: Typical Analog to Digital Conversion (ADC) chain.

4.3 Signal Reconstruction

Given a set of signal samples $\{f_n\}$ how can we reconstruct the corresponding analog signal? Well we saw in Section 3.2.2.2 that *perfect reconstruction* can be achieved with an ideal low-pass filter, known as a reconstruction filter. However, just as in the ADC case, if the sampling rate $f_s = 2 \times f_{max}$, then this filter needs to be a non-realizable brick-wall filter - not to mention that it is not possible to generate input to said filter (a train of delta impulse functions).

The solution is to do exactly as per the ADC above, but in reverse.

1. Have an DAC circuit that can generate a S/H type analog signal out based on a discrete series of numeric value (the $\{f_n\}$)
2. These are held constant for the sampling interval
3. This is applied to a non-ideal reconstruction filter.

This is shown in Figure 4.3.1, where it is important to note that the sampling rate $f_s > 2 \times f_{max}$

.

Also shown in Figure 4.3.1 is a mathematical way of modelling the entire system. In particular it is important to note that the final spectrum is that of the sampled spectrum (i.e. with all the repeated spectra) multiplied by a sinc response (arising from the rectangular impulse repsonse of the DAC core) and the reconstruction filter.

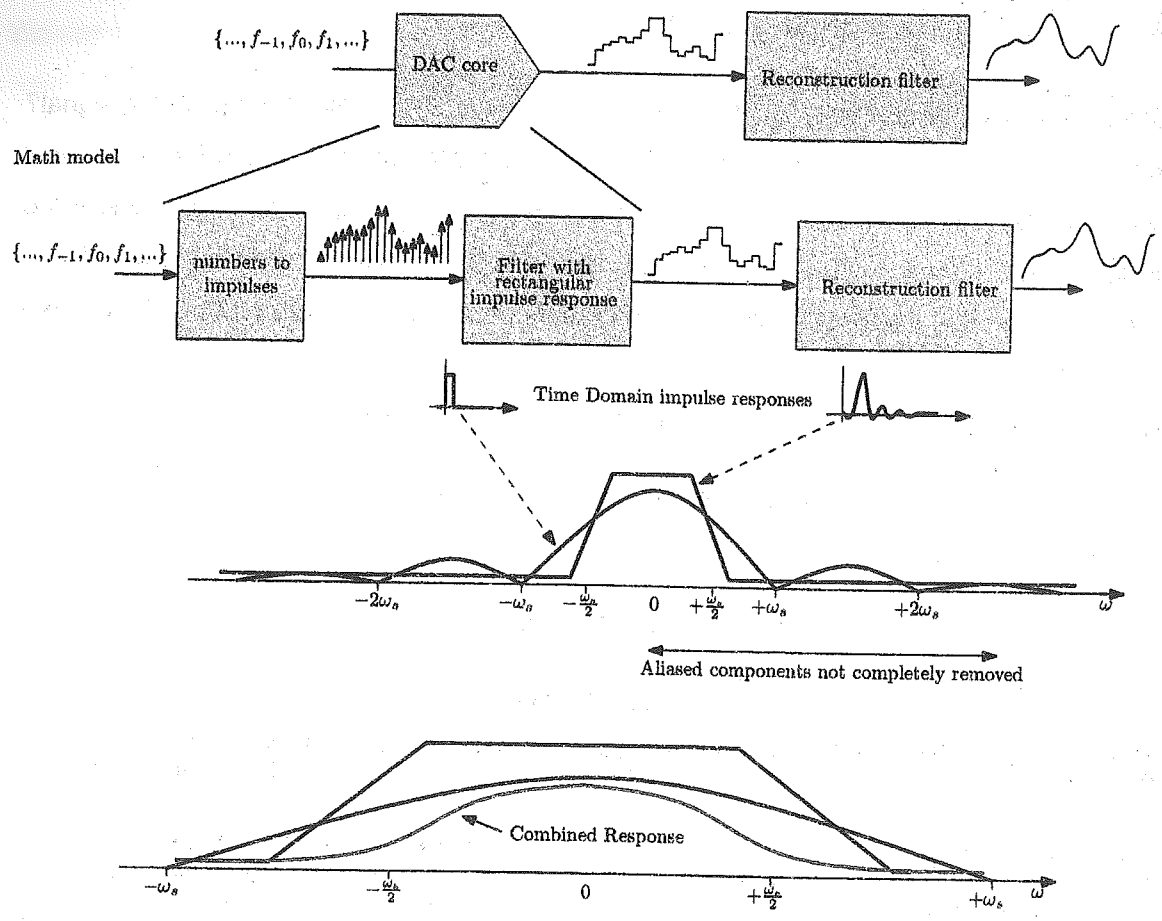


Figure 4.3.1: Typical Digital to Analog Conversion (DAC) chain.

4.3.1 Reconstruction filter

The reconstruction filter should have the following properties

1. The response, when combined with the S/H sinc response, should be ideal at the wanted signal frequencies

2. This combined response should be (ideally) zero at the aliased frequencies.

If the oversampling rate is high, then the above conditions are relatively easy to approximately achieve.

However if operating close to the theoretical $2 \times f_{max}$ then this becomes very difficult - sometimes the digital signal itself is pre-distorted so that the combined response of the pre-distortion, sinc filter, and reconstruction yields the desired behavior. This is an advanced topic.

4.3.2 Quantizer

After sampling, the signal is discrete in time, but is still a continuum in amplitude.

- The signal needs to be mapped (i.e. rounded) to discrete levels
- This process is called quantization, see figure 4.3.2

As can be seen from Figure 4.3.2, quantization introduces irreversible rounding errors, i.e. once done can never be undone, and these can impact system performance, e.g. cause an audible noise in a speech signal. The following section will quantify this noise.

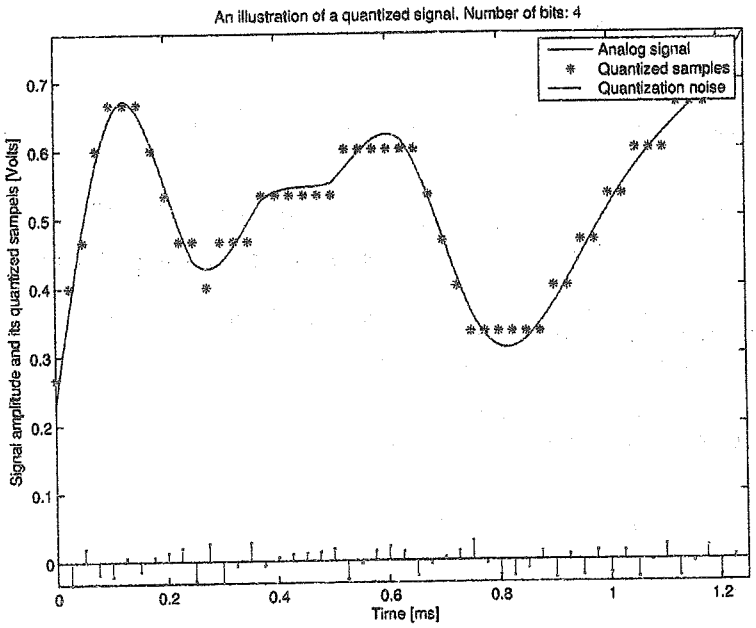


Figure 4.3.2: An analog signal sampled and quantized

4.3.2.1 Quantization noise

Taking the units of amplitude as being Volts, and by assuming a *uniform spacing*, we can define Δ Volts as that voltage separating adjacent quantized levels, see figure 4.3.3. Note that if binary words are associated with quantization levels, then a input change of Δ Volts causes a Least Significant Bit (LSB) change in the resulting binary word, hence Δ is known as the LSB Voltage.

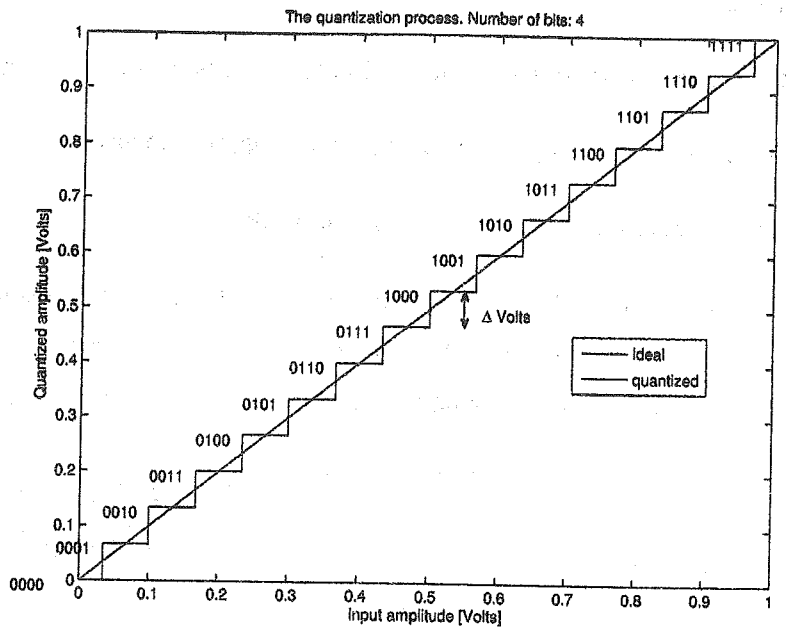


Figure 4.3.3: Typical input-output response of an ADC

We can consider the quantization noise as being equivalent to adding a random variable (RV) to the sampled signal. This random variable has the following characteristics:

- Its magnitude is never greater than $\pm\Delta/2$
- Any value in the range $\pm\Delta/2$ is possible
- Each of its possible values are equally likely

Basically it is a uniformly distributed RV on the range $\pm\Delta/2$, and so its probability density function (pdf) is:

$$f(n) = \begin{cases} \frac{1}{\Delta} & \text{for } -\Delta/2 < n < \Delta/2 \\ 0 & \text{elsewhere} \end{cases}$$

We can now calculate the Root Mean Squared (RMS) voltage of the noise introduced by the quantization process by noting that:

$$\text{RMS Noise Voltage} = \sqrt{E[n^2]}$$

where $E[\cdot]$ is the expectation operator (i.e. mean). It is a well known result in probability and

statistics (see Theorem 2 in Appendix A) that:

$$E\left[n^2\right] = \frac{\Delta^2}{12}$$

Thus:

$$\text{Noise Power} = \frac{\Delta^2}{12} \text{ Watts}$$

We note that the noise power depends only on Δ, i.e. the separation between the quantization level, and does not depend on the signal level.

4.3.2.2 Linear model

In theory, of course, the quantization noise is correlated to the signal - but if the signal amplitude changes randomly by amounts that are large compared to Δ then it can be the case that the quantization noise can be considered as additive white noise that is uncorrelated from the signal.

This is especially true if there is Gaussian thermal noise, with a variance $\gg \frac{\Delta^2}{12}$, present in the original signal.

Note that the quantization noise is however uniformly distributed (not Gaussian).

A linear noise model, as illustrated in Figure 4.3.4 can be used.

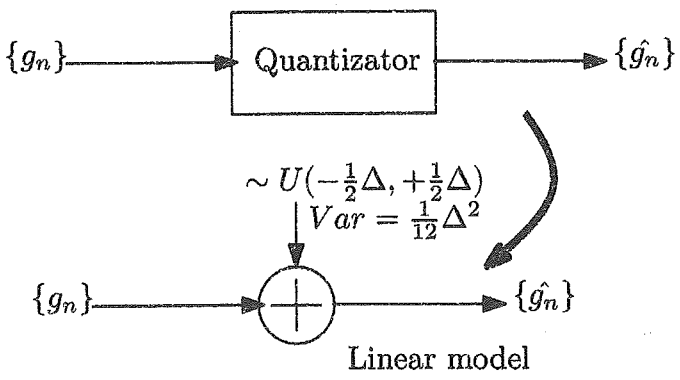


Figure 4.3.4: Approximate linear noise model for quantization noise.

4.3.2.3 Signal to Noise Ratio due to Quantization

A figure of merit often associated with an ADC is its Signal to Noise Ratio (SNR) normally measured when the largest allowable sine-wave is applied. Assuming the only source of noise is Quantization, then we can derive an analytical expression for the SNR.

It the ADC can take any input between $\pm V_{peak}$, then the largest possible sine wave has amplitude V_{peak} , and thus the signal power is $\frac{1}{2}V_{peak}^2$.

We can relate V_{peak} to Δ by noting that a b-bit ADC represents the full input range $2V_{peak}$ with 2^b levels, each separated by $\Delta = \frac{2V_{peak}}{2^b-1}$ Volts.

Thus the noise power is:

$$\begin{aligned} \frac{1}{12}\Delta^2 &= \frac{1}{12}\frac{4V_{peak}^2}{(2^b-1)^2} \\ &\approx \frac{1}{3}\frac{V_{peak}^2}{2^{2b}} \end{aligned}$$

Where the approximation is true when $2^b \gg 1$, as is usually the case.

Now we can easily compute the signal to noise ratio:

$$\text{SNR} = \frac{\frac{1}{2}V_{peak}^2}{\frac{1}{3}\frac{V_{peak}^2}{2^{2b}}} = \frac{3}{2}2^{2b}$$

In communication systems it is usual to quote noise ratios in terms of dB :

$$\text{dB} = 10 \times \log_{10}(\text{linear power ratio})$$

So our SNR due to quantization becomes:

$$\begin{aligned} \text{SNR(dB)} &= 10 \times \log_{10}\left(\frac{3}{2}2^{2b}\right) \\ &= 10 \times \log_{10}\left(\frac{3}{2}\right) + 20 \times \log_{10}\left(2^b\right) \\ &= 10 \times \log_{10}\left(\frac{3}{2}\right) + 20 \times \frac{\log_2\left(2^b\right)}{\log_2(10)} \\ &= 1.76 + 6.02 \times b \quad \text{dB} \end{aligned} \tag{4.3.1}$$

Observations:

- SNR improves by 6dB with every extra bit (b)
- In practice the SNR varies according to frequency as well as other factors
- Quantization noise effects low and high signal voltages equally.
  - $\Rightarrow$Instantaneous SNR can be very bad for low signal levels.
  - This can be a problem for some signals, e.g. Audio.

4.3.2.4 Selection of quantization level

Lets begin by evaluating equation 4.3.1 for several difference numbers of bits:

#bits = b	#level = 2^b	SNR [dB]
8	256	49.92
10	1,024	61.96
12	4,096	74
14	16,384	86.04
16	65,536	98.08
18	262,144	110.12

A word on terminology:

For the 12 bit case, we might say: "The quantization noise is 74dB down on the signal", or "The quantization is -74dB", or "The SNR due to quantization is 74dB".

The choice of how many bit to use really depends on the application. If for example, we are dealing with an audio signal that has been picked up by a cheap microphone and a noisy amplification circuit (as might be the case in a mobile phone for example), then it might be that the signal being presented to the ADC circuit already has -40dB of noise present. Noting that dB are on a log scale, we can see that -40dB of noise is far noisy than -74dB. Note the 24dB difference is 8×3dB, and each 3dB corresponds to a doubling of power. Therefore the analog audio signal being presented to the 12-bit ADC would already have 8 times the amount of noise present than what the ADC is going to add.

4.3.2.5 Examples

Compact Disc (CD)

- Stereo, i.e. 2 ADC channels
- $F_s = 44.1\text{kHz}$ per channel
- $b = 16\text{bits}$ resolution per channel

Digital part of Plain Old Telephone Service (POTS)

- Single channel
- $F_s = 8\text{kHz}$
- $b = 8\text{bits}$ resolution

- Analog still used on local loop (to / from the home)

Chapter 5

Discrete (Time) Fourier Transform

5.1 Introduction

In deriving the Sampling Theorem we saw what the spectrum of a sampled signal looks like (its main feature is that it is periodic). We now wish to explore this further. We will find an expression for the spectrum in terms of the values of the time samples and then we will examine how we might compute the spectrum using a digital machine.

5.2 DTFT definition

The Discrete Time Fourier Transform (DTFT), $\bar{F}(j\omega)$, is simply the Fourier Transform of $\bar{f}(t)$, a discrete time signal, i.e.:

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

where $f_n \triangleq f(nT)$ are the discrete times samples of the analog signal $f(t)$, T is the sampling interval, and $f_s = \frac{1}{T}$ is the sampling rate in samples per second.

Exercise:

Prove the following: $\bar{F}(j\omega) = \sum_{n=-\infty}^{+\infty} f_n e^{-jn\omega T}$

Solution:

So we have the very important result:

$$\bar{F}(j\omega) = \sum_{n=-\infty}^{+\infty} f_n e^{-jn\omega T} \tag{5.2.1}$$

Notation

- The formula in the box 5.2.1, is the definition of the *Discrete Time Fourier Transform (DTFT)*. In general $\bar{F}(j\omega)$ is complex.

- The magnitude (or modulus, or amplitude) of the spectrum, $\left|\bar{F}(j\omega)\right|$ is the *magnitude spectrum* (or amplitude spectrum).

- The phase $\angle\bar{F}(j\omega)$, is known as the *phase spectrum.*

5.2.1 Properties

We can write $\bar{F}(j\omega)$ as:

$$\bar{F}(j\omega) = \sum_{n=-\infty}^{+\infty} f_n \cos(n\omega T) - j \sum_{n=-\infty}^{+\infty} f_n \sin(n\omega T)$$

From this we can deduce the following four properties:

1. For real $\{f_n\}$, we have $\bar{F}(j\omega)$ is real and even, if and only if $\{f_n\}$ are even (i.e. if $f_n = f_{-n}$)
   i.e. if $f_n = f_{-n}$ then $\Im\left(\bar{F}(j\omega)\right) = 0$, and $\bar{F}(j\omega) = \bar{F}(-j\omega)$.

2. For real $\{f_n\}$, we have $\bar{F}(j\omega)$ is imaginary and odd, if and only if $\{f_n\}$ are odd (i.e. if $f_n = -f_{-n}$)
   i.e. if $f_n = -f_{-n}$ then $\Re\left(\bar{F}(j\omega)\right) = 0$, and $\bar{F}(j\omega) = -\bar{F}(-j\omega)$.

3. For real $\{f_n\}$, we have $\bar{F}(j\omega)$ and $\bar{F}(-j\omega)$ are complex conjugates.

4. For any $\{f_n\}$ (i.e. real or complex) then $\bar{F}(j\omega)$ is periodic in ω with period $\frac{2\pi}{T}$
   i.e. $\bar{F}(j\omega) = \bar{F}\left(j\left(\omega + m\frac{2\pi}{T}\right)\right)$, where m is any integer.

Exercise:

Confirm the correctness of these four properties.

Properties 1, 2, & 3 are shared with the conventional Fourier transform.
Property 4 is not in general held by the conventional Fourier Transform (apart of course for the transform of periodic functions which is what $\bar{F}(j\omega)$ is).

Again we emphasize, the spectrum (i.e. the DTFT) of a sampled signal is periodic. The spectrum repeats every f_s Hz (i.e. every $\frac{2\pi}{T}$ radians per second).

5.3 The Shift Theorem

Exercise:

If $\{f_n\}$ is delayed by K samples, as shown in Figure **??**, then the DTFT is multiplied by $e^{-jK\omega T}$.

Solution:

Note

Because $\left|e^{-jK\omega T}\right| = 1 \ \forall \omega$ the multiplication by $e^{-jK\omega T}$ does not change the *amplitude* spectrum (but it does change the *phase* spectrum).

5.4 The Discrete Fourier Transform (DFT)

There are two difficulties in performing the machine computation of the DTFT (equation 5.2.1)

1. The sum is infinite, $\sum_{n=-\infty}^{+\infty}$
2. The independent variable ω is a continuous variable (so there are an infinity number of points to compute).

For difficulty 1 we will just consider a finite number of time domain samples. Let us just use N samples (starting from $n = 0$):
The DTFT now becomes

$$\bar{F}(j\omega) \simeq \sum_{n=0}^{N-1} f_n e^{-jn\omega T}$$

For difficulty 2 we can argue as follows: although ω is a continuous variable, there are only N degrees of freedom involved in the calculation of $\bar{F}(j\omega)$; These come from the N independent values of $\{f_n\}$. So we need only consider N samples of $\bar{F}(j\omega)$.
Property 4 (periodicity) of the DTFT suggests that we take the N samples of $\bar{F}(j\omega)$ over one period, say from $\omega = 0$ to $\omega = \frac{2\pi}{T}$.

The convention is to use:

$$\omega = \frac{m}{N}\frac{2\pi}{T} \text{ for } 0 \leq m \leq N-1$$

as illustrated in Figure 5.4.1.

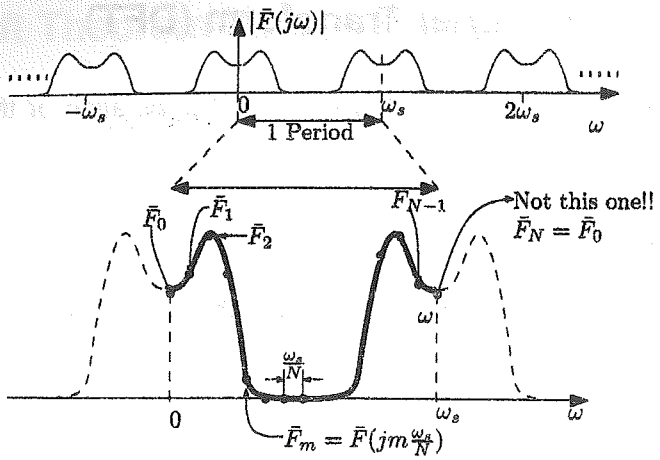


Figure 5.4.1: Usually the N samples of $\bar{F}(j\omega)$ are all taken from the first repeated spectrum.

The usual notation for the DFT is:

$$\bar{F}_m = \sum_{n=0}^{N-1} f_n e^{-j\frac{2\pi mn}{N}} \text{ for } 0 \leq m \leq N-1 \tag{5.4.1}$$

Where:

$$\begin{aligned} \bar{F}_m &\triangleq \bar{F}(j\omega)\big|_{\omega=\frac{m}{N}\frac{2\pi}{T}} \\ &= \bar{F}\left(j\frac{m}{N}\frac{2\pi}{T}\right) \end{aligned}$$

The $\{f_n\}$ are time domain samples

n is the time index

The $\{\bar{F}_m\}$ are frequency domain samples

m is the frequency index.

5.4.1 The inverse DFT

As with most good transforms, the DFT has an inverse given by:

$$f_n = \frac{1}{N}\sum_{m=0}^{N-1} \bar{F}_m e^{+j\frac{2\pi mn}{N}} \text{ for } 0 \leq n \leq N-1 \qquad (5.4.2)$$

Exercise:

Verify that the above formula is indeed the inverse of the DFT.

Solution:

Hint: you you most likely need the formula proven in appendix A.1:

$$\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} = \begin{cases} N & \text{when } k \text{ is a multiple of } N \\ 0 & \text{elsewhere} \end{cases}$$

The DFT and its inverse (the iDFT)

$$\bar{F}_m = \sum_{n=0}^{N-1} f_n e^{-j\frac{2\pi mn}{N}} \quad m = 0, 1, \cdots, N-1$$

and

$$f_n = \frac{1}{N} \sum_{m=0}^{N-1} \bar{F}_m e^{+j\frac{2\pi mn}{N}} \quad n = 0, 1, \cdots, N-1$$

5.5 The DFT in Matrix form

The N equations embodied in equation 5.4.1 can be compacted into a single matrix form which in some areas of study can prove useful (i.e. it allows us to easily use results from linear algebra). We have:

$$\begin{bmatrix} \bar{F}_0 \\ \bar{F}_1 \\ \vdots \\ \vdots \\ \bar{F}_{N-1} \end{bmatrix} = \begin{bmatrix} e^{-j\frac{2\pi.0.0}{N}} & e^{-j\frac{2\pi.0.1}{N}} & \cdots & \cdots & e^{-j\frac{2\pi.0.(N-1)}{N}} \\ e^{-j\frac{2\pi.1.0}{N}} & e^{-j\frac{2\pi.1.1}{N}} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ e^{-j\frac{2\pi(N-1).0}{N}} & \cdots & \cdots & \cdots & e^{-j\frac{2\pi.(N-1)(N-1)}{N}} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ \vdots \\ f_{N-1} \end{bmatrix}$$

or more compactly:

$$\bar{\mathbf{F}} = \Omega_{\mathbf{N}}\mathbf{f}$$

Where the terms in **bold** are vector or matrix versions of the corresponding terms in the previous equation. The matrix $\Omega_{\mathbf{N}}$ is known as the size N DFT matrix. On first sight it would appear that $\Omega_{\mathbf{N}}$ contains the following list of unique elements:

$$\left\{ 1 \quad , e^{-j\frac{2\pi}{N}} \quad , e^{-j\frac{2\pi.2}{N}} \quad , e^{-j\frac{2\pi.3}{N}} \quad \cdots \quad , e^{-j\frac{2\pi.(N-1)(N-1)}{N}} \right\}$$

i.e. $(N-1)^2$ unique elements. However we note that all terms are just integer powers of $W_N \triangleq e^{-j\frac{2\pi}{N}}$, which is a unit magnitude vector with angle $-\frac{2\pi}{N}$ radians.

$$\left\{ W_N^0 \quad , W_N^1 \quad , W_N^2 \quad , W_N^3 \quad \cdots \quad , W_N^{(N-1)(N-1)} \right\}$$

Thus every term in $\Omega_{\mathbf{N}}$ is a unit magnitude vectors with angle that is a integer multiple of $-\frac{2\pi}{N}$ radians. Plotting this on a unit circle, we see that there are in fact only N unique integer powers of W_N due to the effect of wrapping at 2π. This is illustrated in Figure 5.5.1 for the simple case where $N = 8$.

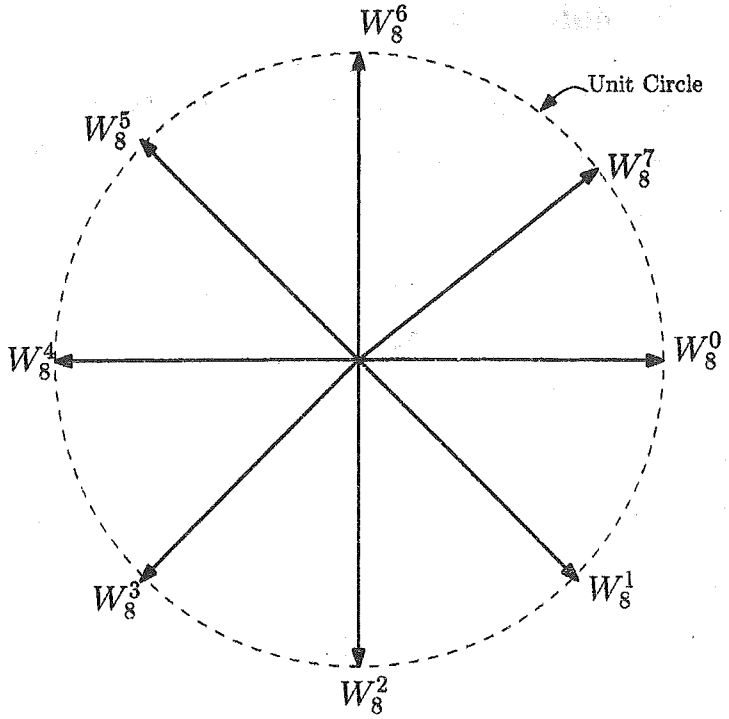


Figure 5.5.1: All possible integer powers of W_8 plotted on the units circle - note only first 8 are labeled but all others map to the same set of points.

Thus Ω_N has only N unique values:

$$\Omega_N = \begin{bmatrix} 1 & 1 & \cdots & \cdots & \cdots & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & \cdots & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & \cdots & W_N^{N-2} \\ 1 & W_N^3 & W_N^6 & W_N^9 & W_N^{12} & \cdots & W_N^{N-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{N-2} & W_N^{N-3} & \cdots & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

5.5.1 Graphical interpretation

We can view the DFT matrix graphically as shown in Figure 5.5.2

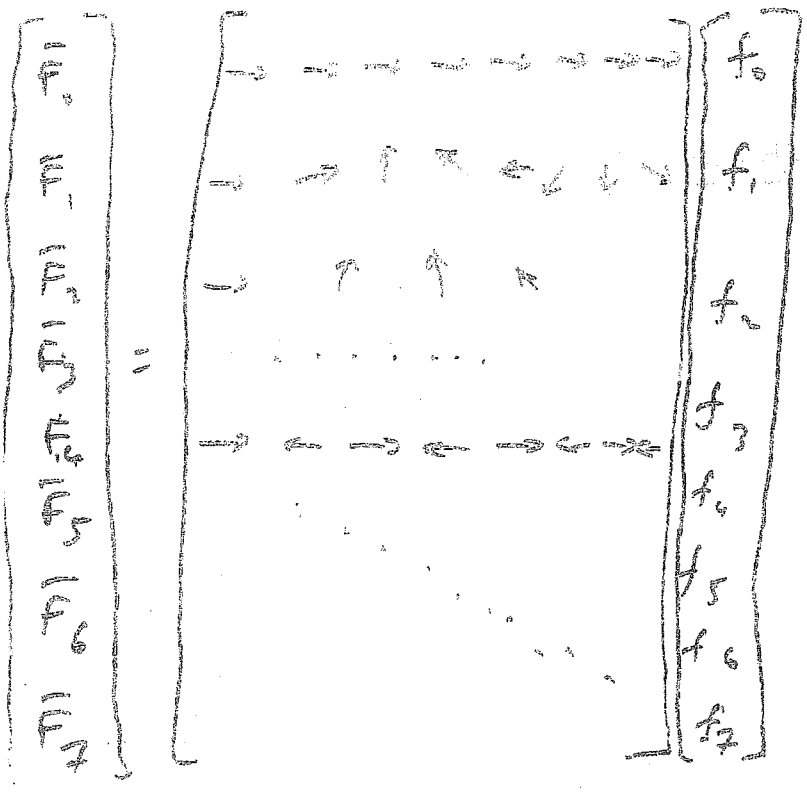


Figure 5.5.2: Graphical view of the matrix version of the DFT.

5.5.2 Example: size 4 DFT matrix

(a) Compute the size 4 DFT matrix, and (b) use it to compute the DFT of the signal $f_n = \{1, 2, 3, 4\}$.

Solution:

(a) The size 4 DFT matrix is given by:

$$\Omega_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^4 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix}$$

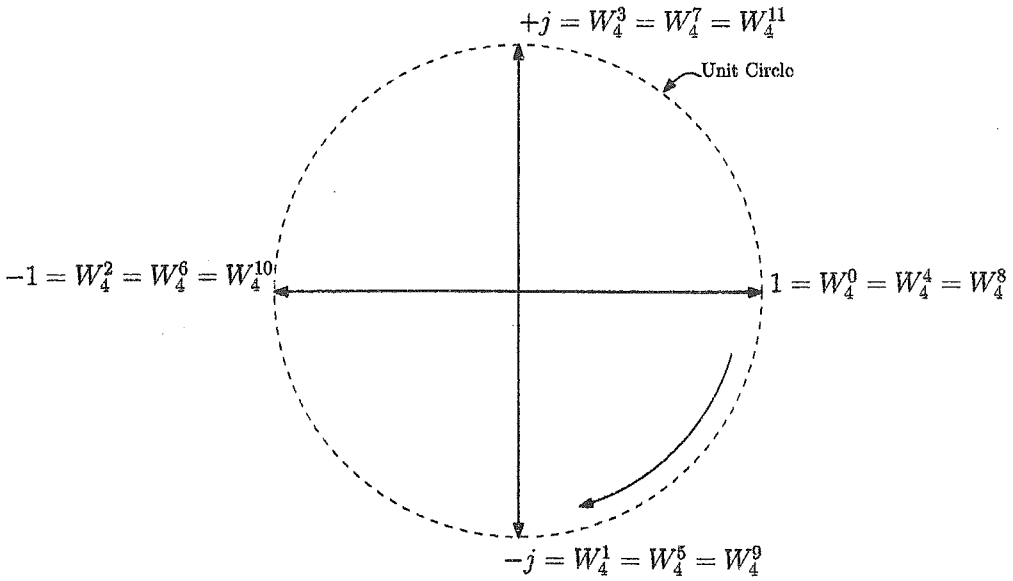


Figure 5.5.3: All possible integer powers of W_4 plotted on the units circle - Useful for computing the size 4 DFT (and inverse DFT) matrix.

Reading off the values of W_4 from Figure 5.5.3 we have:

$$\Omega_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$$

Note:

- The first row is always all-ones.
- The next row is obtained by starting at 1, and moving clockwise on Figure 5.5.3 one-step-at-a-time.
- The next row is obtained by starting at 1, and moving clockwise on Figure 5.5.3 two-steps-at-a-time.
- etc...

(b) The DFT of $f_n = \{1, 2, 3, 4\}$ is given by:

$$\begin{bmatrix} \bar{F}_0 \\ \bar{F}_1 \\ \bar{F}_2 \\ \bar{F}_3 \end{bmatrix} = \Omega_4 \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ -2+j2 \\ -2 \\ -2-j2 \end{bmatrix}$$

So the DFT of $\{1, 2, 3, 4\}$ is $\{10,\ -2+j2,\ -2,\ -2-j2\}$.
Note the conjugate symmetry and the fact that the DC and $\frac{1}{2}f_s$ terms are both real valued.

5.5.3 Exercises

Do the following:

1. Based on the definition of the iDFT compute the iDFT matrix, Ω_4^{-1}
2. Verify that it is indeed the inverse DFT by computing $\Omega_4^{-1}\Omega_4$
3. Use Ω_4^{-1} to compute the iDFT of $\{10,\ -2+j2,\ -2,\ -2-j2\}$. (the result from example 5.5.2)

Chapter 6

Graphical development of the DFT

6.1 one-pager!

All you need to know about the DFT and it relationship to the DTFT is compactly illustrated in Figure 6.1.1. It summarizes, in graphical form, the development of the Discrete Fourier Transform (we've already done this mathematically).

The development is given in the form of seven Fourier transform pairs labeled (a) to (g).

The bidirectional arrows $\Leftrightarrow$ are used to denote the Fourier transforms.

All the graphs on the left hand side are functions of time and on the right hand side we have functions of frequency ω (in radians per second).

Before continuing further we firstly have some reminders of basic theory.

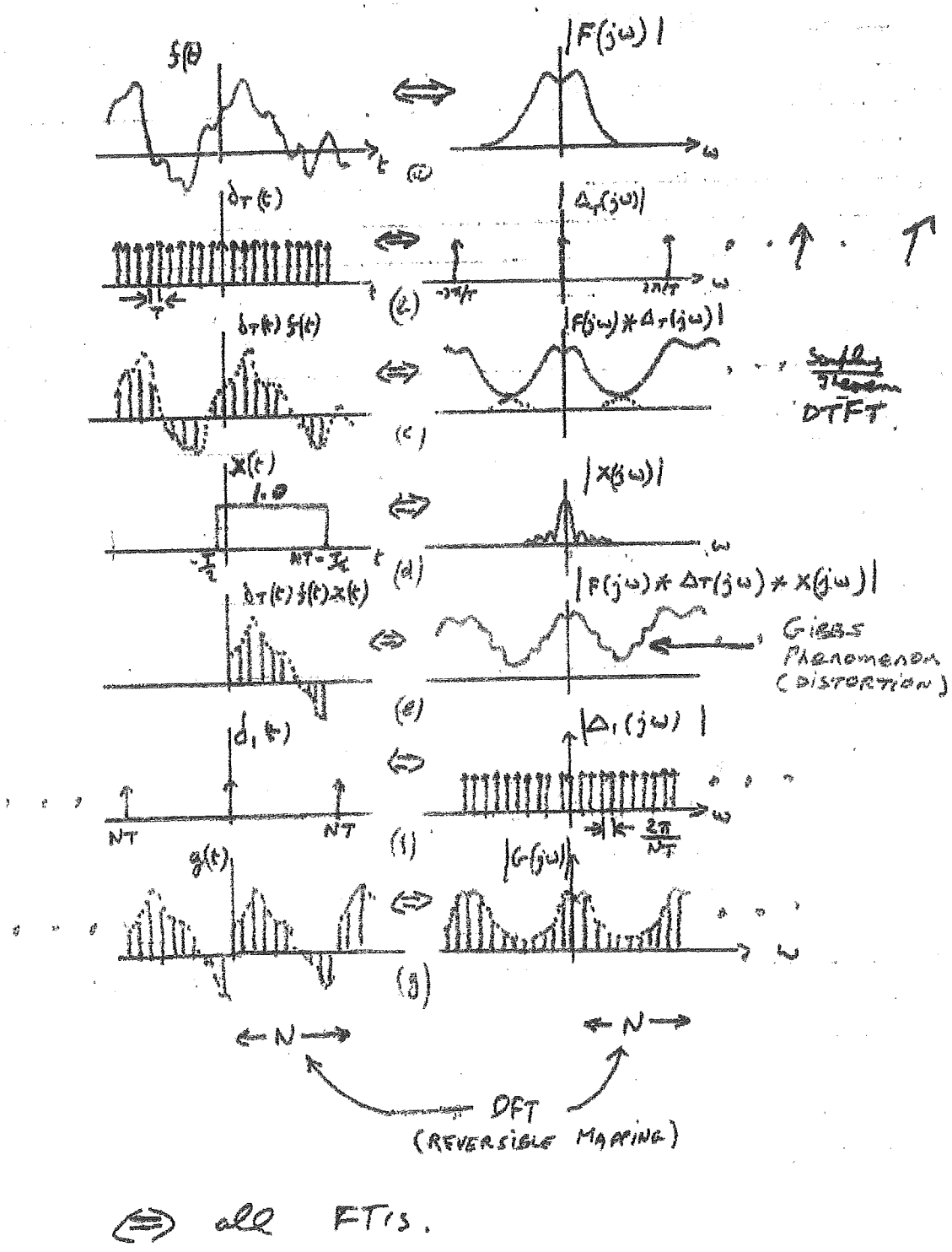


Figure 6.1.1: Graphical development of the DFT one pager!

6.2 Reminders

6.2.1 Convolution definition

Let $h(t)$ be the convolution of $f(t)$ and $g(t)$, so we have:

$$h(t) = \int_{-\infty}^{+\infty} f(\tau)\, g(t-\tau)\, d\tau$$

For shorthand we sometimes write $h = f \star g$, where $\star$ denotes convolution.

In words: convolving two functions involves reversing one of them and running (shifting) it past the other from $-\infty$ to $+\infty$.
For each value of shift we multiply the two functions together and integrate the product. The resulting third function is a function of the shift.

6.2.2 Convolution theorem

The Fourier transform of the product of two time functions is the convolution of the individual transforms.

The Fourier transform of the convolution of two time functions is the product of the individual transforms.

Multiply in time domain $\Rightarrow$ Convolve in frequency domain
Convolve in time domain $\Rightarrow$ Multiply in frequency domain

Duality! Multiply in one domain $\Leftrightarrow$ Convolve in the other

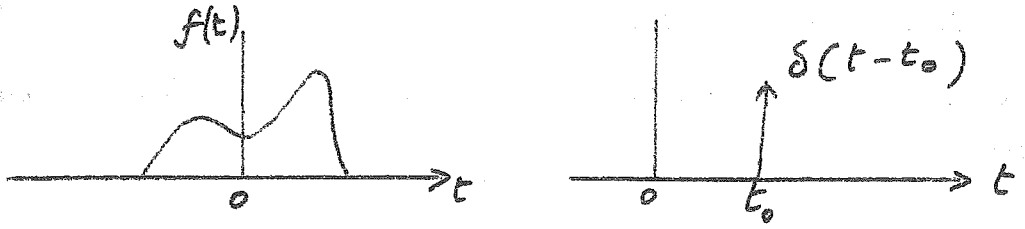


Figure 6.2.1: The two functions to be convolved together.

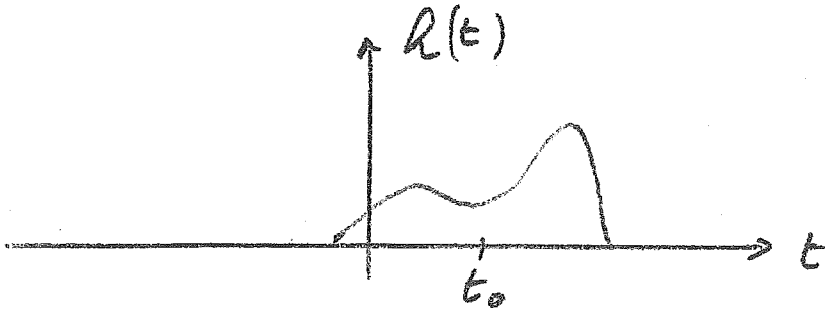


Figure 6.2.2: The result of convolving $f(t)$ with a delta function is that $f(t)$ is shifted in time.

6.2.3 Convolution with a delta function

Suppose we want to convolve the continuous function $f(t)$with a delta function centred on $t = t_o$ as per Figure 6.2.1

From the definition of convolution we have

$$h(t) = \int_{-\infty}^{+\infty} f(\tau)\,\delta(t-\tau-t_o)\,d\tau$$

Using the sifting property we get:

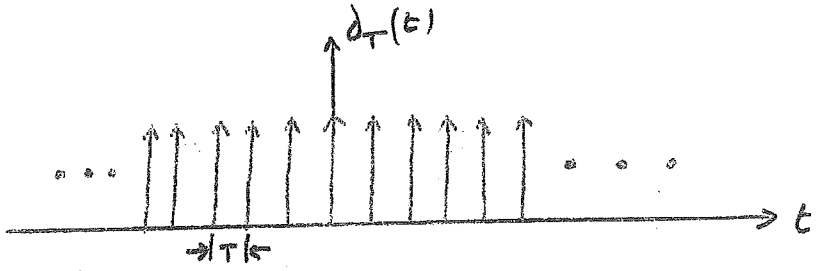
$$h(t) = f(t-t_o)$$

Thus convolving a continuous function with a delta function moves (shifts) the function to the position of the delta function.

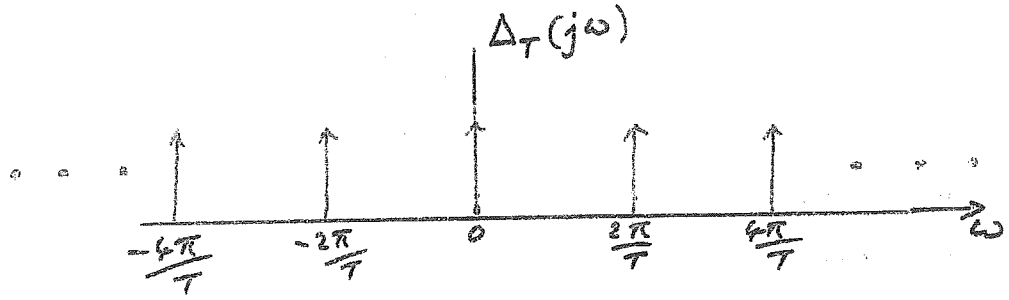
6.2.4 Train of impulses

The Fourier transform of a train of impulses is a train of impulses.

So the Fourier transform of



Is



We saw this in the derivation of the sampling theorem when we used the Fourier series to represent a train of impulses.

Here we see another example of the *duality* we often find in the Fourier transform. When we move the impulses closer together in one domain they move further apart in the other.

6.2.5 Rectangular pulse

The Fourier transform of a rectangular function is a sinc function.

Consider a rectangular pulse centered on the time origin as shown in Figure 6.2.3.

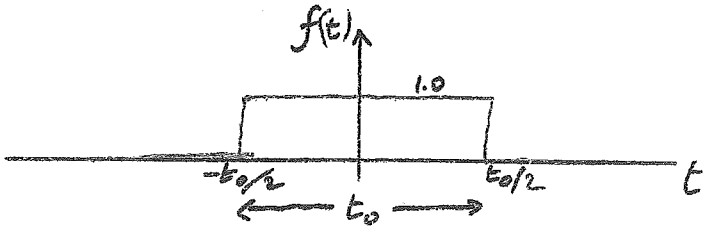


Figure 6.2.3: A rectangular pulse centered on the time origin.

The Fourier transform of this is:

$$\begin{aligned} F(j\omega) &= \int_{-\frac{t_o}{2}}^{+\frac{t_o}{2}} e^{-j\omega t} dt \\ &= t_o \left[\frac{\sin\left(\frac{\omega t_o}{2}\right)}{\frac{\omega t_o}{2}} \right] \\ &= t_o \text{sinc}\left(\frac{\omega t_o}{2}\right) \end{aligned}$$

where $\text{sinc}(x) \triangleq \frac{\sin(x)}{x}$.

$F(j\omega)$ is illustrated in Figure 6.2.4.

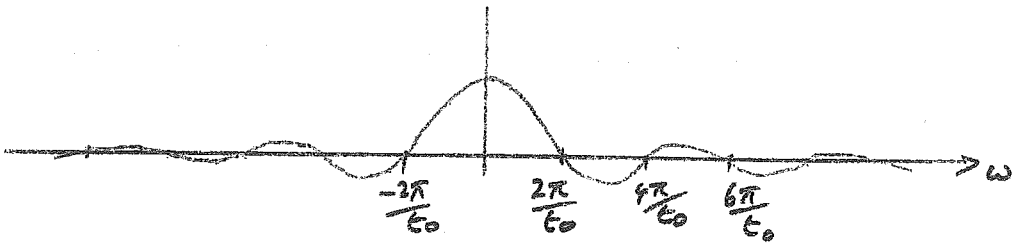


Figure 6.2.4: The Fourier transform of a rectangular pulse is a sinc.

Tip: The position of the first zero crossing in Hz of the sinc function is equal to the inverse of the pulse width expressed in seconds.

(Duality: Although we don't need it now, it is worthwhile noting that the inverse Fourier

transform of a rectangular function in the frequency domain is a sinc function in the time domain).

Exercise

Show that the magnitude of the Fourier transform is the same no matter where in the time domain the pulse occurs.

6.3 Back to the one pager!

With reference to Figure 6.1.1 we now consider each of the transform pair in turn.

6.3.1 (a) Random signal (to be sampled)

A random time signal and its Fourier transform.

We assume that the transform exists.

The signal may or may not be band-limited in the frequency domain.

6.3.2 (b) Train of delta functions

As per the reminder Section 6.2.4 the Fourier transform of a train of delta functions $\delta_T(t)$ is itself a train of delta functions denoted $\Delta_T(j\omega)$ as illustrated in Figure XX.

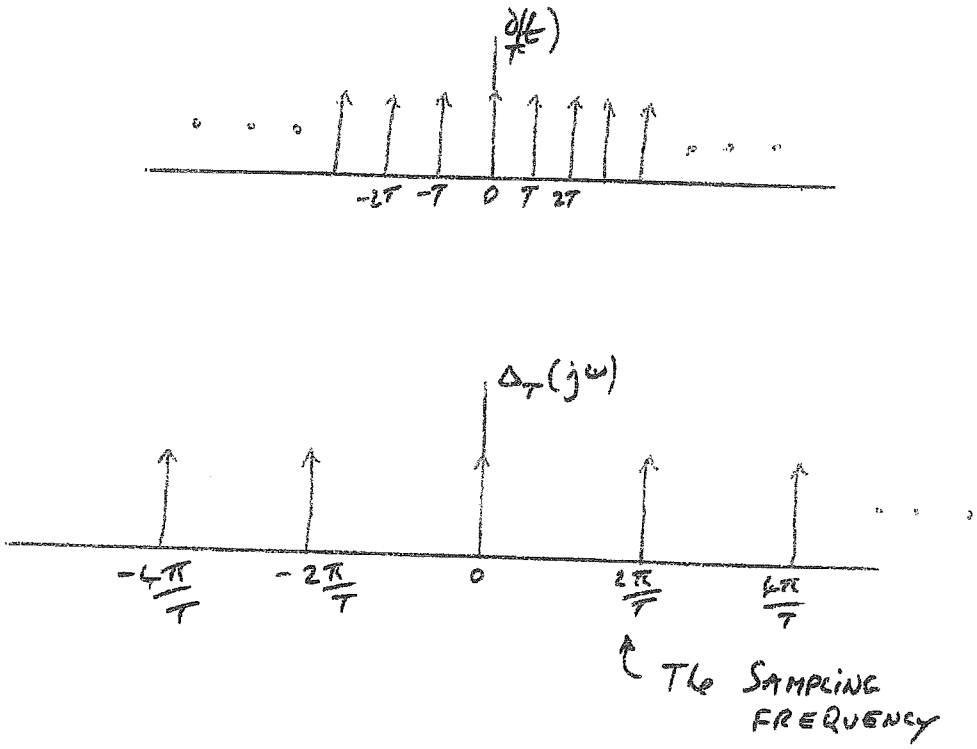


Figure 6.3.1: The Fourier transform of a train of delta functions is itself a train of delta functions.

6.3.3 (c) Sampled signal

The sampled signal and its transform.

From the definition of the DTFT, Equation 5.2.1, we have:

$$\bar{F}(j\omega) = \sum_{n=-\infty}^{+\infty} f_n e^{-jn\omega T}$$

A graphical of thinking about this is to note that sampling is achieved by multiplying the time domain functions of (a) and (b) which corresponds to convolution in the frequency domain (see the convolution theorem in Section 6.2.2).

In math: from our treatment of the sampling theorem (Section 3.2.1) we know that:

$$\bar{F}(j\omega) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} F\left(j\left(\omega - n\frac{2\pi}{T}\right)\right)$$

Hence Figure 6.3.2 illustrates the magnitude of the result in the frequency domain.

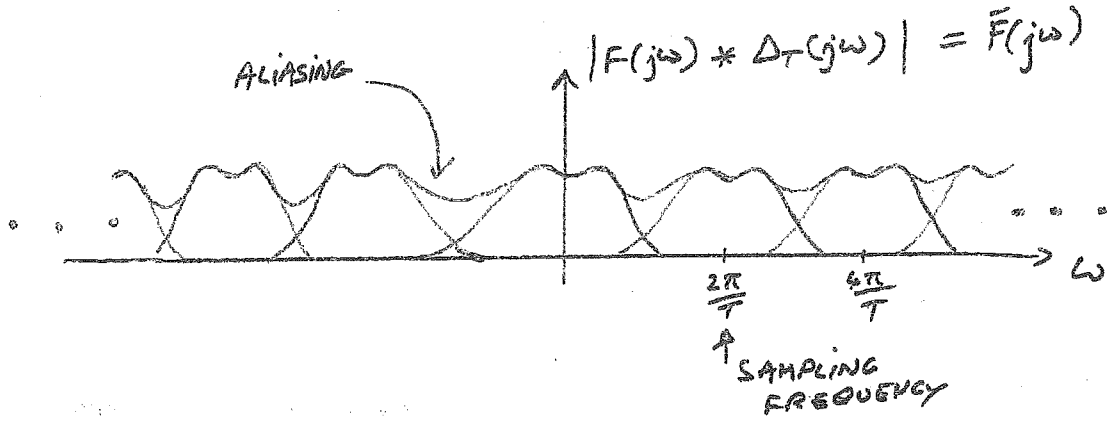


Figure 6.3.2: The magnitude of the frequency domain view of a sampled signal.

6.3.4 (d) Windowing in time domain

Moving from the DTFT to the DFT, we only consider the first N samples (starting at $n = 0$). This is the same as applying (multiplying) a time domain rectangular window function (Figure 6.3.3) to the signal.

The rectangular pulse is just wide enough to capture the first N time domain samples.

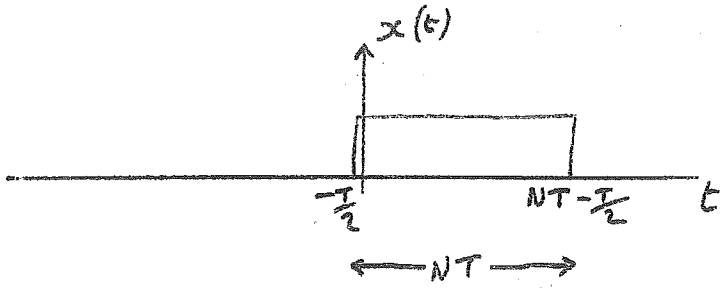


Figure 6.3.3: The rectangular window function used to limit the infinite series to a finite one (of length N)

As per Section 6.2.5, the Fourier transform of this time limiting function (a rectangular pulse) is a sinc (see Figure 6.3.4).

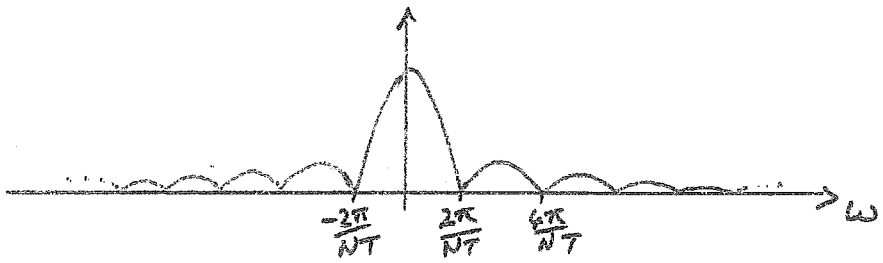


Figure 6.3.4: The Fourier transform of the rectangular windowing function is a sinc.

If N is large (i.e. a wide time domain window) then the corresponding frequency domain sinc is narrow, and vice versa - another example of Fourier transform duality.

(Note: If N is very large, then this sinc function can be regarded as a crude approximation to a delta function).

6.4 (e) Time limited function

The time limiting was achieved by multiplication by the rectangular pulse.
The corresponding action in the frequency domain is convolution with the sinc function.
These are both shown in Figure 6.4.1.

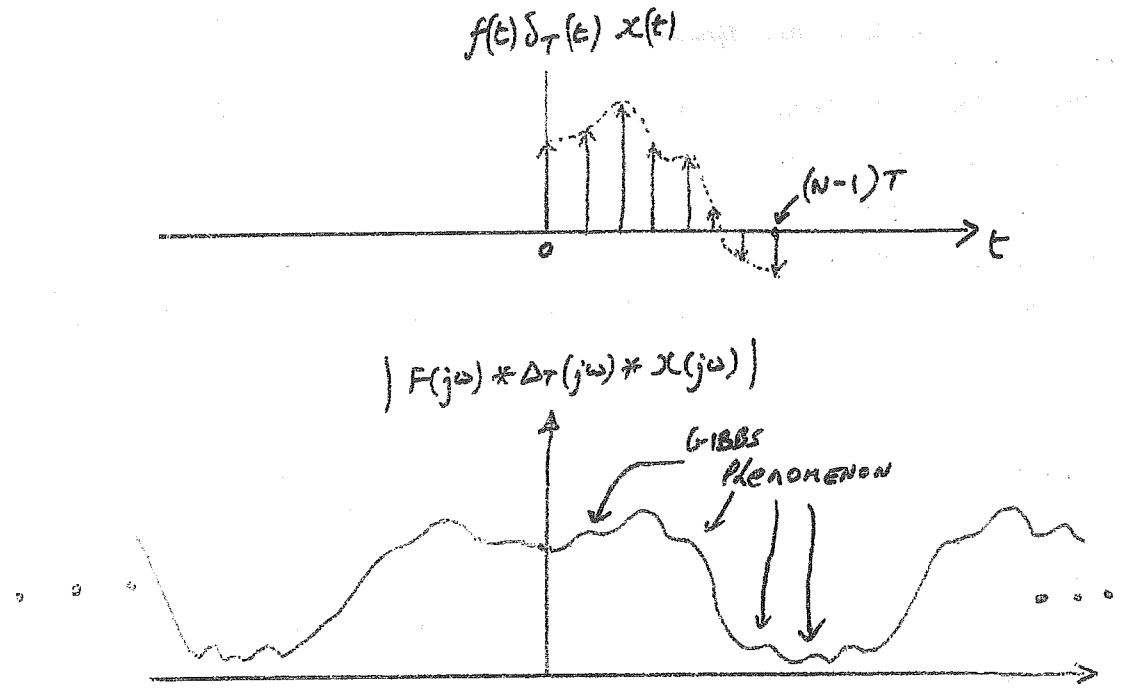


Figure 6.4.1: The time limited function (top) and its Fourier transform (bottom)

The distortion, seen here in the frequency domain, is called the Gibb's Phenomenon. This happens because the sinc function is not a delta function (The sinc function has a main lobe of non-zero width and it has side lobes).
It was noted earlier that the larger N the closer the sinc approximates a delta function and thus the Gibbs effect can be reduced (but not fully eliminated) by increasing the value of N.

6.4.1 (f) Frequency sampling

In the development of the DFT we also just used a sub-set of the frequency domain (as ω is a continuous variable) - this is the same as sampling in the frequency domain which is multiplication (in the frequency domain) by a train of delta pulses. These (and their inverse Fourier transform) are shown in this Fourier pair.

As before the Frequency sampling function (train of impulses) and its inverse transform (also a train of impulses).

6.4.2 (g) The Sampled spectrum

Frequency domain sampling is achieved by multiplication by a train of impulses.
The corresponding time domain action is convolution by a train of impulses.

This convolution causes the time domain function to repeat indefinitely.
We have chosen our parameters so that this repetition occurs without gaps and without overlap.

We note that, in transform pair (g), both functions are discrete (sampled),
But we know that being sampled in one domain implies periodicity in the other and vice versa.

⇒in transform pair (g), both functions are discrete (sampled) and periodic. The sample spacing in the time domain is T seconds.

The sample spacing in the frequency domain is $\frac{2\pi}{NT}$ radians per second i.e. $\frac{1}{NT}\,\mathrm{Hz} = \frac{f_s}{N}\,\mathrm{Hz}$.

The duration of one time domain period is NT seconds.

The width of one period in the frequency domain is $\frac{2\pi}{T}$ radians per second i.e $\frac{1}{T}\,\mathrm{Hz}$.

6.5 The DFT and its inverse

Finally: The DFT is a reversible mapping from the first N time domain samples to the first N frequency domain samples.

The calculation in either direction can be done numerically as follows:

The DFT and its inverse (the iDFT)

$$\bar{F}_m = \sum_{n=0}^{N-1} f_n e^{-j\frac{2\pi mn}{N}} \quad m = 0, 1, \cdots, N-1$$

and

$$f_n = \frac{1}{N} \sum_{m=0}^{N-1} \bar{F}_m e^{+j\frac{2\pi mn}{N}} \quad n = 0, 1, \cdots, N-1$$

The second formula, known as the *inverse DFT*, or iDFT is used to compute the time domain samples from the frequency domain ones (it's correctness was proven in Section 5.4.1).

Chapter 7

Fast Fourier Transform (FFT)

7.1 Introduction

The Fast Fourier Transform (FFT), despite its name, is not actually a transform is its own right. The FFT is an efficient algorithm for computing the DFT.
We know that a size N DFT (and N-point DFT) can be computed by multiplying an $N \times N$ matrix, Ω_N , by a length N column vector.

$$\bar{\mathbf{F}} = \Omega_N \mathbf{f}$$

Where $\mathbf{f}$ is the length N time domain vector to be transformed, and $\bar{\mathbf{F}}$ is the desired DFT result.
If we don't apply any cleverness this N-point DFT would require N^2 multiplications; we sometimes say its complexity has order N^2, compactly written as $\mathcal{O}(N^2)$. This means that if we $\times 2$ the size of N the complexity increases by $\times 4$ etc...
The FFT algorithm can improve on this by exploiting the structure of Ω_N.

Real signal (in the time domain)

One obvious complexity reduction technique occurs when the time domain vector is real valued. In this case the DFT values exhibit conjugate symmetry, i.e. $\bar{F}_m = \bar{F}^*_{N-m} \ \forall m$. Therefore we need only compute $\bar{F}_m$ for $m = 0, 1, ..., \frac{1}{2}N - 1$, and the other half are obtained for free (i.e. taking the conjugate involves simply negating the imaginary part), i.e. the complexity is halved.
This is illustrated in Figure 7.1.1.

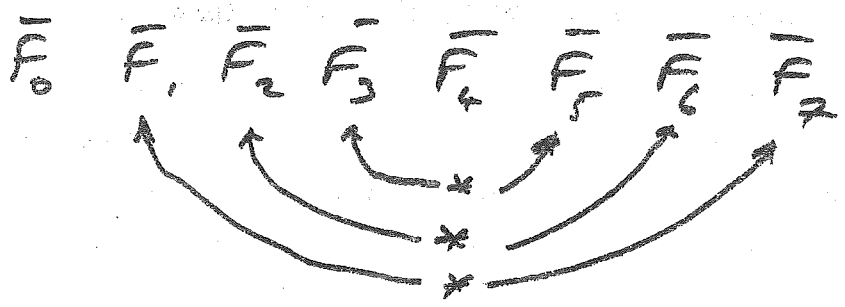


Figure 7.1.1: Example of conjugate symmetry for $N = 8$

Whilst a halving in complexity is great, it is still $\mathcal{O}(N^2)$.

The FFT algorithm used here assumes f_n is a complex signal yet it, as we will see, achieves a complexity order of $\mathcal{O}(N \times \log_2 N)$.

Exercise:

Prove that if the f_n are real valued, then it is always the case that $\bar{F}_m = \bar{F}^*_{N-m} \ \forall m$, i.e. $\bar{F}_1 = \bar{F}^*_{N-1}$ and $\bar{F}_2 = \bar{F}^*_{N-2}$ etc...

Solution:

Principle of time decomposition by divide and conquer

Suppose we divide $\mathbf{f}$ (the time domain vector) into two length $\frac{1}{2}N$ vectors.
Then to compute the two $\frac{1}{2}N$-point DFTs (by multiplication by $\Omega_{\frac{1}{2}N}$) would require $\left(\frac{1}{2}N\right)^2 = \frac{1}{4}N^2$ multiplications. If it takes M multiplications to combine these two DFT results into the N-point DFT result we require, then the total number of multiplications is:

$$\frac{1}{2}N^2 + M$$

This compares to N^2 using the inefficient matrix multiplication method.
Thus, if $M < \frac{1}{2}N^2$ then there is a reduction in computational complexity reduction!

We could imagine an iterative process where we compute the two length $\frac{1}{2}N$-point DFTs by firstly computing $\frac{1}{4}N$-point DFTs, and so on...
This is the basis of divide and conquer methods.

Note: here, in the first iteration, we have assumed that N is even (i.e. $\frac{1}{2}N$ is an integer). If we continue to apply many iterations then, as we will see later, the restriction means that N must be a power-of-two.
There is a loss in generality here, but as we shall see later, there are ways around this.

The FFT is actually a family of algorithms - the one thing they have in common is that they are all computationally very efficient. What was demonstrated above was the idea of dividing (or decomposing) the time domain vector $\mathbf{f}$ into smaller parts to obtain a complexity advantage; hence the name *time decomposition algorithm*. A frequency decomposition algorithm is also possible, but we'll focus on time decomposition.

7.2 Time decomposition FFT algorithm

We'll follow the usual notation:
$f_n,\ n = 0, 1, \cdots, N-1$ are the time domain *complex* samples
$\tilde{F}_m,\ m = 0, 1, \cdots, N-1$ is the DFT of $\{f_n\}$.

From the definition of the DFT we have:

$$\bar{F}_m = \sum_{n=0}^{N-1} f_n W_N^{mn} \quad m = 0, 1, \cdots, N-1$$

(this is what we want to compute)

where $W_N \triangleq e^{-j\frac{2\pi}{N}}$.

Exercise:

Prove the following:

- $W_N^n = -W_N^{n-\frac{1}{2}N}$, and
- $(W_N^n)^m = W_{\frac{1}{m}N}^n$

Solution:

These will be used in the development of the FFT.

Using the divide-and-conquer method let's divide f_n into two length $\frac{1}{2}N$ sequences as fol-

lows:

$$\left.\begin{array}{rcl} a_n & = & f_{2n} \\ b_n & = & f_{2n+1} \end{array}\right\} n = 0, 1, \cdots, \frac{N}{2} - 1$$

In words:

a_n are the even indexed elements of f_n and,

b_n are the odd indexed elements of f_n.

Each of these half sized sequences have their own DFTs:

$$\left.\begin{array}{rcl} \bar{A}_m & = & \sum_{n=0}^{\frac{1}{2}N-1} a_n W_{\frac{1}{2}N}^{mn} \\ \bar{B}_m & = & \sum_{n=0}^{\frac{1}{2}N-1} b_n W_{\frac{1}{2}N}^{mn} \end{array}\right\} m = 0, 1, \cdots, \frac{N}{2} - 1$$

For this to work we require a formula that expresses $\bar{F}_m$ in terms of $\bar{A}_m$ and $\bar{B}_m$; lets try to do that:

$$\begin{aligned} \bar{F}_m &= \sum_{n=0}^{N-1} f_n W_N^{mn} \quad m = 0, 1, \cdots, N-1 \\ &= \sum_{\substack{n=0 \\ n,\ even}}^{N-1} f_n W_N^{mn} + \sum_{\substack{n=0 \\ n,\ odd}}^{N-1} f_n W_N^{mn} \\ &= \sum_{n=0}^{\frac{1}{2}N-1} f_{2n} W_N^{m(2n)} + \sum_{n=0}^{\frac{1}{2}N-1} f_{2n+1} W_N^{m(2n+1)} \\ &= \sum_{n=0}^{\frac{1}{2}N-1} f_{2n} \left(W_N^2\right)^{mn} + W_N^m \sum_{n=0}^{\frac{1}{2}N-1} f_{2n+1} \left(W_N^2\right)^{mn} \end{aligned}$$

But $W_N^2 = W_{\frac{1}{2}N}$, thus:

$$\begin{aligned} \bar{F}_m &= \sum_{n=0}^{\frac{1}{2}N-1} f_{2n} w_{\frac{1}{2}N}^{mn} + w_N^m \sum_{n=0}^{\frac{1}{2}N-1} f_{2n+1} w_{\frac{1}{2}N}^{mn} \\ &= \bar{A}_m + w_N^m \bar{B}_m \quad m = 0, 1, \cdots, N-1 \end{aligned}$$

Wait a minute! In the last line $m = 0, 1, \cdots, N-1$, but $\bar{A}_m$ and $\bar{B}_m$ were only defined on the range $m = 0, 1, \cdots, \frac{N}{2} - 1$. This is valid if we recognize that the DFT is a periodic function and so they are just repeated outside of this range.

So in summary:

$$\bar{F}_m = \bar{A}_m + W_N^m \bar{B}_m \quad m = 0, 1, \cdots, N-1$$

This equation is the basis of the time decomposition FFT.

Going back to our complexity analysis, we noted earlier that that the computation of $\bar{A}_m$ and $\bar{B}_m$ requires $\frac{1}{2}N^2$multiplications (using the matrix multiplication method). Here we also need to compute $W_N^m \bar{B}_m$ which requires an additional N multiplications.

Thus the N-point DFT requires $\frac{1}{2}N^2 + N$ multiplications.

But we can still do better..., but for now lets look at what we've done so far in a bit more detail.

7.2.1 Signal flow diagrams

In signal processing we often represent a computation in diagrammatic form. Such a diagram is called a signal flow diagram (or sometimes a signal flow graph). Figure 7.2.1 shows an example.

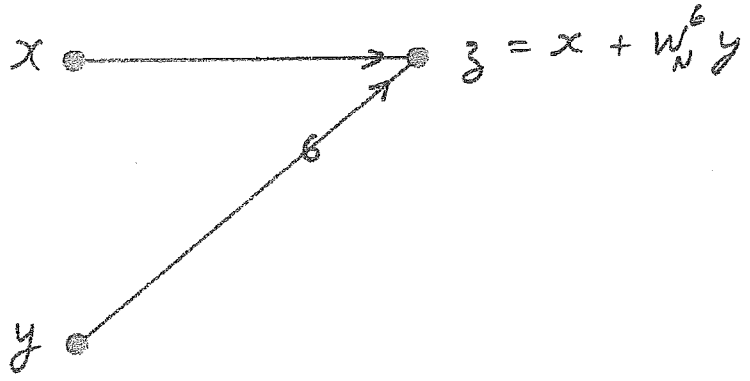


Figure 7.2.1: Signal flow diagram example

In Figure 7.2.1 the signal flow is from left to right.
Where the line merge the signal add.
The number written on the line represents a multiplication by that power of W_N.
Using this notation we can draw a signal flow diagram from the computation of a N-point DFT from two $\frac{1}{2}N$-point DFT, this is done for $N = 8$ in Figure 7.2.2.

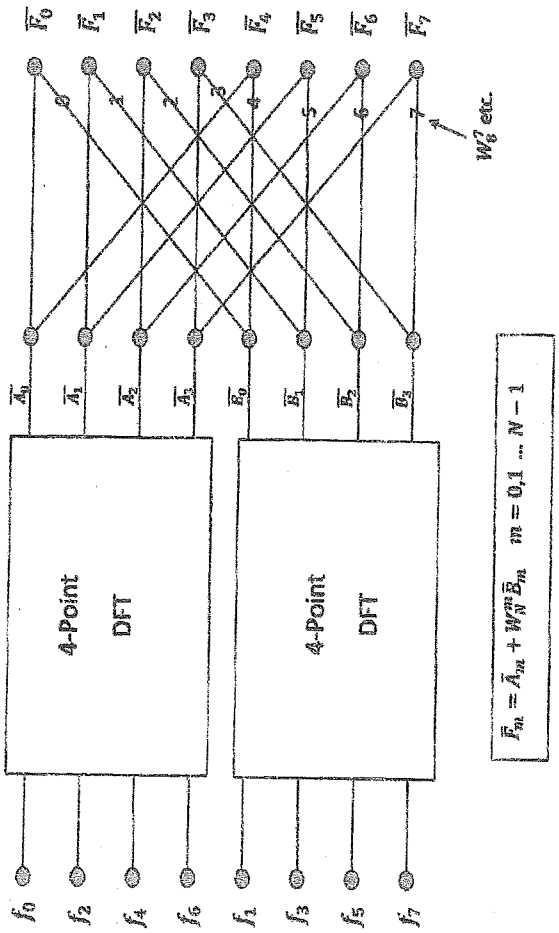


Figure 7.2.2: First stage of the time decomposition FFT algorithm for $N = 8$

7.2.2 Why stop there?

Consider Figure 7.2.2, it should be clear that the same "trick" can be applied to the two 4-point DFT. This is shown in Figure 7.2.3.

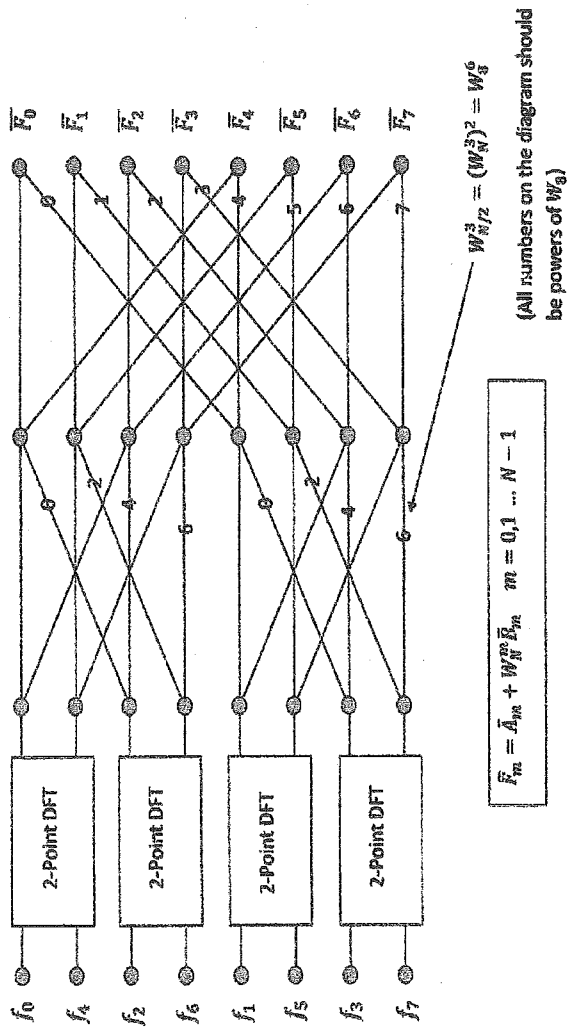


Figure 7.2.3: First & second stages of the time decomposition FFT algorithm for $N = 8$.

Clearly we can go again, this is shown in Figure 7.2.4.

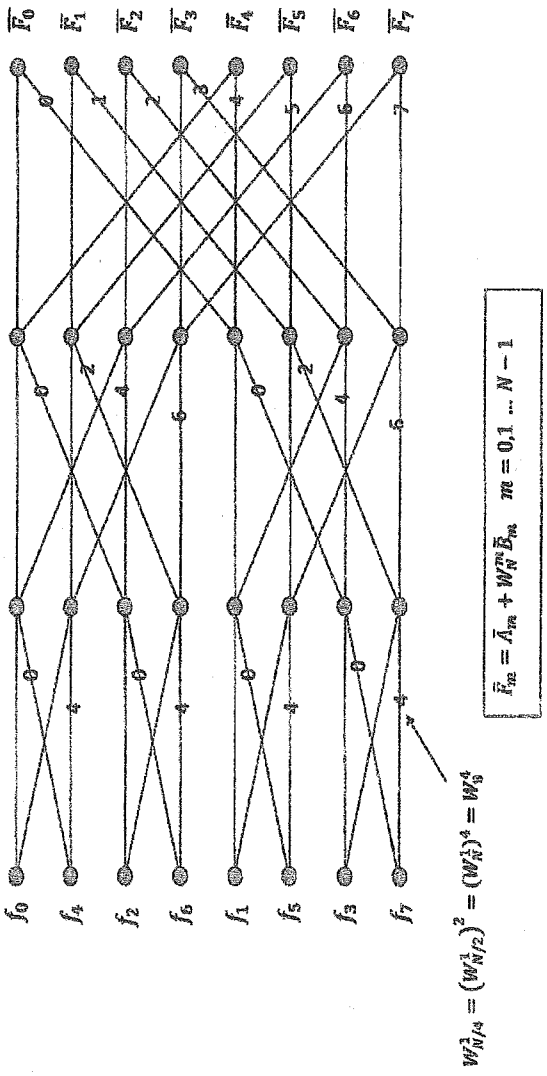


Figure 7.2.4: Complete time decomposition FFT algorithm for $N = 8$

7.2.3 Power-of-two

In this algorithm, the 1^{st} stage requires that N be even. The 2^{nd} requires that $\frac{1}{2}N$ be even, and the 3^{rd} stage requires that $\frac{1}{4}N$ be even.

Another way of saying this is that we require $\frac{1}{2}N$, $\frac{1}{4}N$ and $\frac{1}{8}N$ to all be integers with the last of these requirements being sufficient, i.e. we only require $\frac{1}{8}N$ to be an integer.

This was for 3 stages. In general if there are q stages, then we require that $\frac{1}{2^q}N$ be an integer.

Let q be the number of stages used in the complete FFT and if we look at the size of the DFTs implemented at each stage we see that the last stage implements 2-point DFTs, then previous one implements 4-point DFTs , the previous one implements 8-point DFTs etc.. until the last stage implements a 2^q-point DFT. But the output from the last stage should be

a N-point DFT, thus:

$$\begin{aligned} N &= 2^q \\ \Rightarrow q &= \log_2(N) \end{aligned}$$

For example, when $N = 8$, we required $\log_2(8) = 3$ stages.

Obviously, as we must have an integer number of stages, q, this also implies that N (being $= 2^q$) be an integer power-of-two, e.g. 8,16, 32, 64, etc...

Note that during the development of the FFT it was notes that the first stage requires that N be even. By the same argument we can say that the 2^{nd} requires that $\frac{1}{2}N$ be even, and the 3^{rd} stage requires that $\frac{1}{4}N$ be even, etc...
Another way of saying this is that we require $\frac{1}{2}N$, $\frac{1}{4}N$, ... $\frac{1}{2^q}N$ to all be integers with the last of these requirements being sufficient, i.e. we require $\frac{1}{2^q}N$ to be an integer, but $N = 2^q$, so clearly $\frac{1}{2^q}N = 1$ and so selecting N as a power of two simultaneously satisfies the two constraints:

- An integer number of stages can implement a full FFT
- Each stage implements integer sized DFTs.

In the real world there are many FFT implementations, e.g. in you mobile phone, and digital TV, and it is not uncommon for them to be of size 4096 or bigger.

7.3 Complexity analysis

So how well have we done? Lets count the number of multiplications. Refer to Figure 7.2.4 and we can see the following number of multiplications[1]:

Stage	# multiplication
1	8
2	8
3	8
Total	16

[1]For stage 3, in theory there are 8 multiplications, but actually these multiplications are trivial and some implementations don't add to the complexity.

In general there are N multiplications per stage, and there are $q = \log_2(N)$ stages, thus there are $N \times \log_2(N)$ multiplications required to implement a N-point DFT using this FFT algorithm.

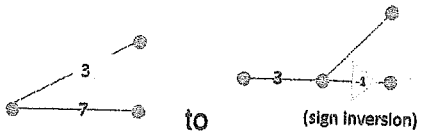
7.3.1 We can do better (by half)!

There is one further simplification that is generally done. It is inspired by noting that:

$$W_N^n = -W_N^{n-\frac{1}{2}N}$$

for example: $W_8^7 = -W_8^3$.

Thus, referring to Figure 7.2.4, we can change:



i.e. every pair of multiplications can be replaced by just 1 multiplication, resulting in half the complexity.

Thus the overall complexity is $\frac{1}{2}N \times \log_2 N$

Note, despite this in $\mathcal{O}$ notation, we still say it has complexity $\mathcal{O}(N \times \log_2 N)$.

7.3.2 Examples

Putting some number to these complexity equations we have:

N	N^2	$\frac{1}{2}N \times \log_2(N)$	%
32	1024	80	7.8%
256	65,536	1,024	1.6%
4096	16,777,216	24,576	0.146%

Wow! We can really see that, especially if N is large, the reduction is complexity by using this FFT algorithm is huge! (99.85% for the 4096 case). This is also illustrated in Figure 7.3.1

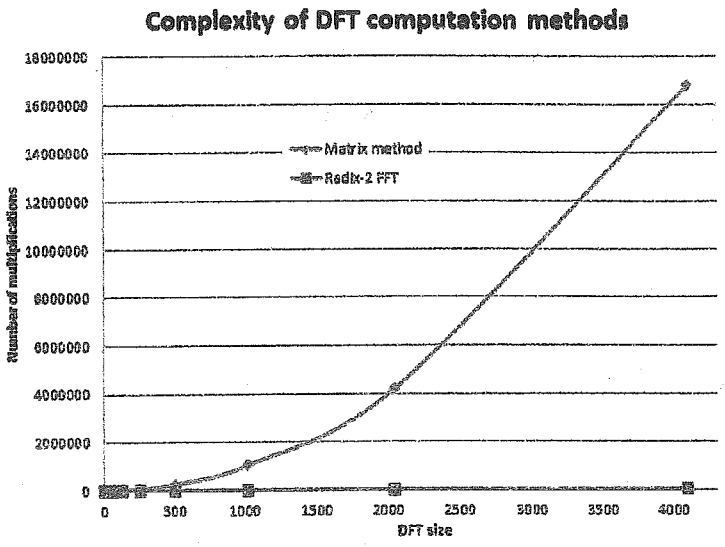


Figure 7.3.1: Complexity comparison between matrix and FFT methods of computing the DFT for various sizes.

7.3.3 Other Radix algorithms

The FFT described above worked by dividing the time vector in half at each stage; this is known as a radix-2 algorithm (as there are 2 halves).

Other options are possible too. We could have, for example, split the vector into 4 quarters; this would be known as a radix-4 algorithm.

A complete DFT computation using only radix-4 stages is an extremely efficient algorithm.

Many implementations contain a mixture of radix-2 and radix-4 stages. Of course each radix-2 stage requires that N be divisible by 2 etc.. This has an immediate implication on the factors of N, as illustrated in the following examples.

Examples

- 64 has factors 4,4 and 4 and so can be very efficiently implemented using 3 radix-4 stages.
- 32 has factors 4,4 and 2 and so can be implemented using 2 radix-4 stages and 1 radix-2 stage.

Note that, in the 32-point DFT case, it might be better to implement 5 radix-2 stages which mean that we'd only have to implement one type of stage.

It is often the case that the more factors N has the more efficient we can make the FFT algorithm, but it is also possible to have efficient algorithms for N prime. However we will not cover any of these.

7.4 Zero padding

What can we do if the number of time samples we have is not a power of two but we only have a radix two algorithm available? We can use a technique known as zero padding (or augmentation with zeros). The technique is very simple: we just append zero valued samples onto our discrete time signal to bring the number of samples up to the next highest power of two and take the FFT of this extended sequence.

Example:

Suppose we have 11 samples in our time signal and we have available a size 16 FFT, then we zero pad with 5 extra zeros as shown in Figure 7.4.1. We then take the FFT of the length 16 sequence.

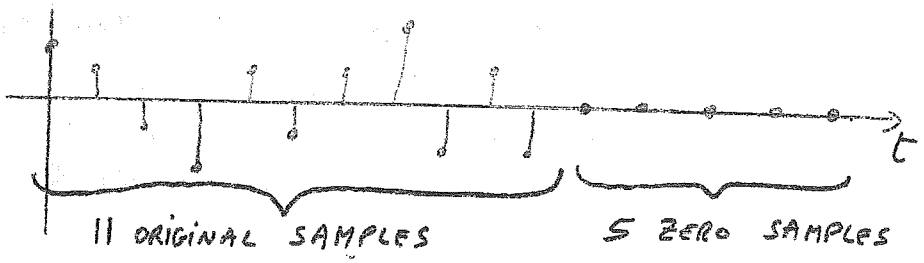


Figure 7.4.1: An example of zero padding

7.4.1 Justification

Is it ok to simply just append some zeros to the end of a sequence? what is the implication of doing so?

To answer these questions we need to look deeper, let us start by asking what exactly is the meaning each of the DFT values $\bar{F}_m$?

The $\bar{F}_m$ are the samples of $\bar{F}(j\omega)$ taken at $\omega = m\frac{2\pi}{NT}$, where $\bar{F}(j\omega)$ is the DTFT of the finite length sequence f_n

$$\bar{F}(j\omega) = \sum_{n=0}^{N-1} f_n e^{-jn\omega T}$$

where each term has its usual meaning.

If we zero pad $\{f_n\}$ to be a new sequence $\{g_n\}$ of length M we can define this new sequence as:

$$g_n = \begin{cases} f_n & \text{for } n = 0, 1, \cdots, N-1 \\ 0 & \text{for } n = N, \cdots, M-1 \end{cases}$$

Exercise:

Prove that: $\bar{G}(j\omega) = \bar{F}(j\omega)$, i.e. the original sequence $\{f_n\}$ and the augmented sequence $\{g_n\}$ have the same DTFT.

Solution:

Now the $\bar{G}_m$, DFT of g_n, are the first M set of samples of $\bar{G}(j\omega) = \bar{F}(j\omega)$ taken at frequency values $\omega = m\frac{2\pi}{MT}$.

But the $\bar{F}_m$ are the first N samples also from $\bar{F}(j\omega)$ taken at frequency values $\omega = m\frac{2\pi}{NT}$.

So, $\bar{F}_m$ and $\bar{G}_m$ are samples of the same function $\bar{F}(j\omega)$ only taken at different spacing; this is illustrated in Figure 7.4.2.

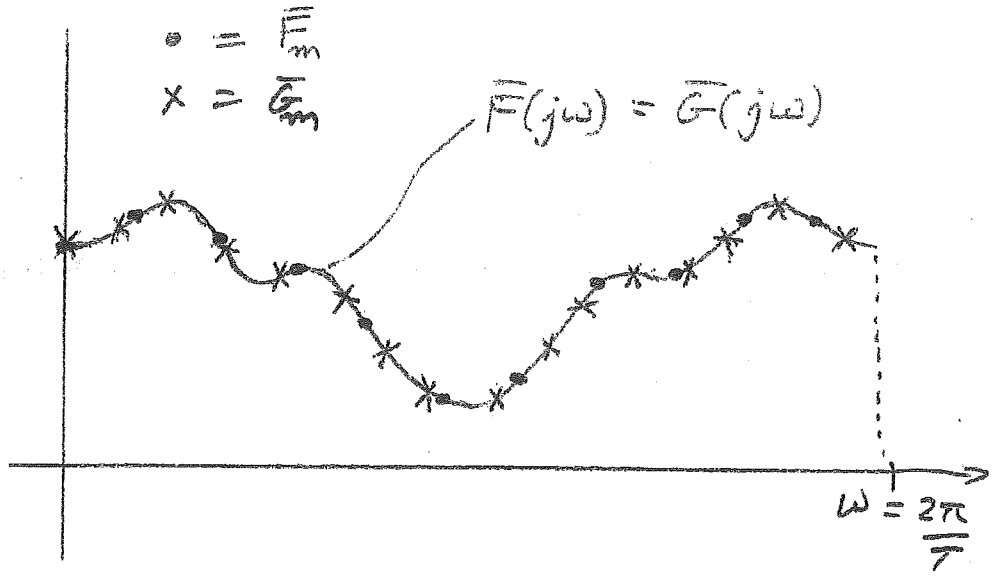


Figure 7.4.2: Effect of zero padding in the time domain is same as interpolation in the frequency domain.

So, augmentation with zero valued samples (*zero padding*) in the discrete time domain corresponds to *interpolation* in the frequency domain.

This is the justification for using zero padding.

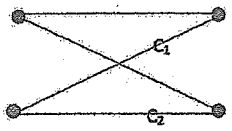
7.5 Implementation issues

7.5.1 The FFT Butterfly

The entire radix-2 FFT signal flow diagram can be constructed from repeated use of the butterfly structure shown in Figure 7.5.1.

Each stage uses $\frac{1}{2}N$ butterflies, so the total number of butterflies is $\frac{1}{2}N \times \log_2(N)$.

In a hardware implementation a smaller number of butterflies can be used in a multiplexed (shared) arrangement. There is a trade-off between speed, power consumption and chip area. The designer has a lot of freedom.



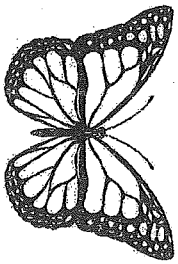


Figure 7.5.1: The FFT butterfly

7.5.2 Pipelined architectures

FFTs are often implemented in hardware and execution speed is a fundamental metric. As with many algorithms we would implement either:

1. A completely serial architecture, where only one butterfly is implemented and it is used over and over again with some additional logic and used to reconfigure the inputs and outputs as the algorithm progresses. Assuming 1 cycle per butterfly this approach would take $\frac{1}{2}N \times \log_2(N)$ cycles to execute (same speed as a single core software implementation on a Digital Signal Processor (DSP) chip).
   Or,

2. A parallel architecture where all butterflies are instantiated. Each stage can run in a single cycle but each stage has to wait for data from the previous stage before it can execute, so it takes $q = \log_2(N)$ cycles to get the result out.
   Or,

3. A combination of the serial and parallel approaches.

An important point to note is that in the parallel architecture, as specified above, only one stage is ever active at a time. This is wasteful! In practice typically the architecture is modified to allow *pipelined* operation. In this case all stages are active all the time, then are just operating on different data sets - essentially several DFTs are being computed at the same time - nice. This is shown graphically in Figure 7.5.2.

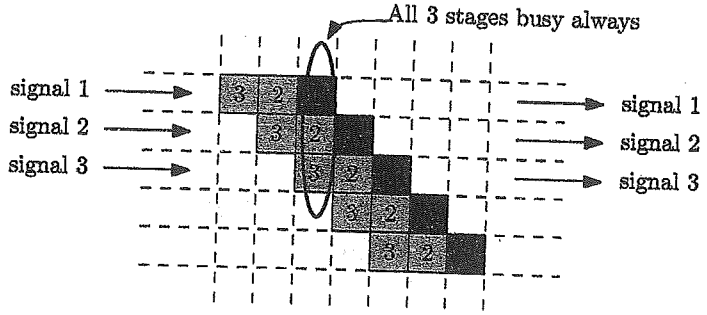


Figure 7.5.2: Pipelined architectures

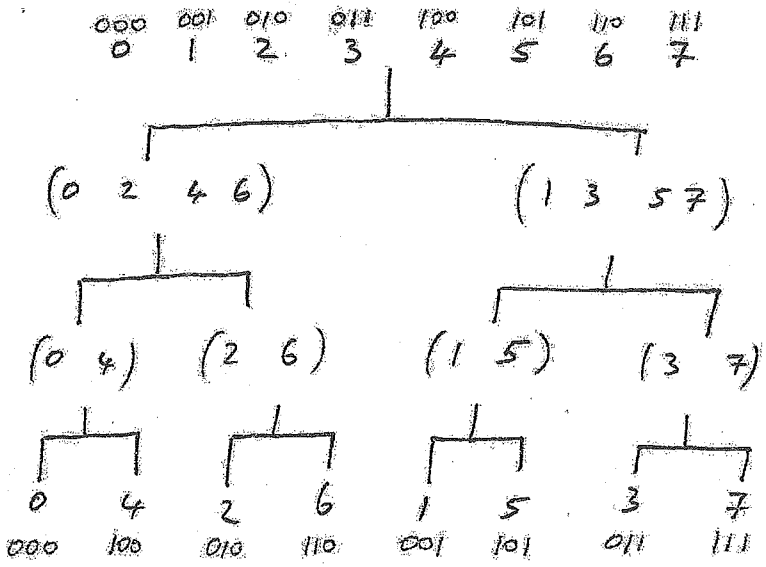


Figure 7.5.3: Top row = output $\bar{F}_m$ ordering, next row is the even and odd ordering from the previous stage of the FFT algorithm and so on. Shown also is the binary representation of the indexes.

7.5.3 Bit reversal

In the decomposition in time FFT algorithm the outputs $\bar{F}_m$ appear in their natural order (from $m = 0$ to $m = N-1$), but the input samples f_n must be presented to the algorithm in jumbled order as illustrated by the tree diagram in Figure 7.5.3:

Notice how the binary versions of the input indexes have their bits in the reverse order compared to the natural order.

So, if the input samples are stored in natural order, then we index them using the bit-reversal indexing to obtain the inputs in the jumbled order.This gives the desired sequencing. The process is known as bit reversal.

Chapter 8

Parseval's Theorem

8.1 Background

For a continuous time domain signal $x(t)$ with Fourier transform $X(j\omega)$ we have Parseval's theorem:

$$\int_{-\infty}^{+\infty} |x(t)|^2 \, dt = \int_{-\infty}^{+\infty} |X(j\omega)|^2 \, d\omega \tag{8.1.1}$$

We seek a similar formulation for discrete time systems.

8.2 Discrete version

Let's consider the length N time domain sequence f and its DFT $\bar{F}$. The discrete version of the integral on the Left-Hand-Side of equation 8.1.1 is

$$\sum_{n=0}^{N-1} |f_n|^2$$

where $|f_n| = f_n^* f_n$.

Exercise:

Prove the following:

$$\sum_{n=0}^{N-1} |f_n|^2 = \frac{1}{N} \sum_{m=0}^{N-1} |\bar{F}_m|^2$$

Solution:

Hint: you will need the following:

$$\sum_{n=0}^{N-1} W_N^{(m-k)n} = \begin{cases} N & \text{if } m = k \\ 0 & \text{elsewhere} \end{cases}$$

which is proven in Appendix A.1.

Parseval's Theorem

$$\sum_{n=0}^{N-1} |f_n|^2 = \frac{1}{N} \sum_{m=0}^{N-1} |\bar{F}_m|^2$$

Chapter 9

Discrete convolution

9.1 Background

The output, $g(t)$ from a continuous LTI system can be described by the convolution of its input, $f(t)$, and the impulse response, $h(t)$.

$$\begin{aligned} g(t) &= \int_{-\infty}^{+\infty} f(\tau)\, h(t-\tau)\, d\tau \\ &= \int_{-\infty}^{+\infty} h(\tau)\, f(t-\tau)\, d\tau \end{aligned}$$

We seek a similar formulation for discrete time systems.

9.2 Discrete convolution

The discrete convolution, $\{g_n\}$, of two sequences $\{f_n\}$ and $\{h_n\}$, written as $\{g_n\} = \{f_n\} \star \{h_n\}$, is defined by the following equation:

$$g_n \triangleq \sum_{k=-\infty}^{+\infty} f_k h_{n-k} \quad \forall n$$

An example of of this is shown in Figure 9.2.1.

The above convolution is sometimes referred to as *direct* discrete convolution (to distinguish it from other types of discrete convolution).

We will see later that direct discrete convolution can be used to compute the output, g,

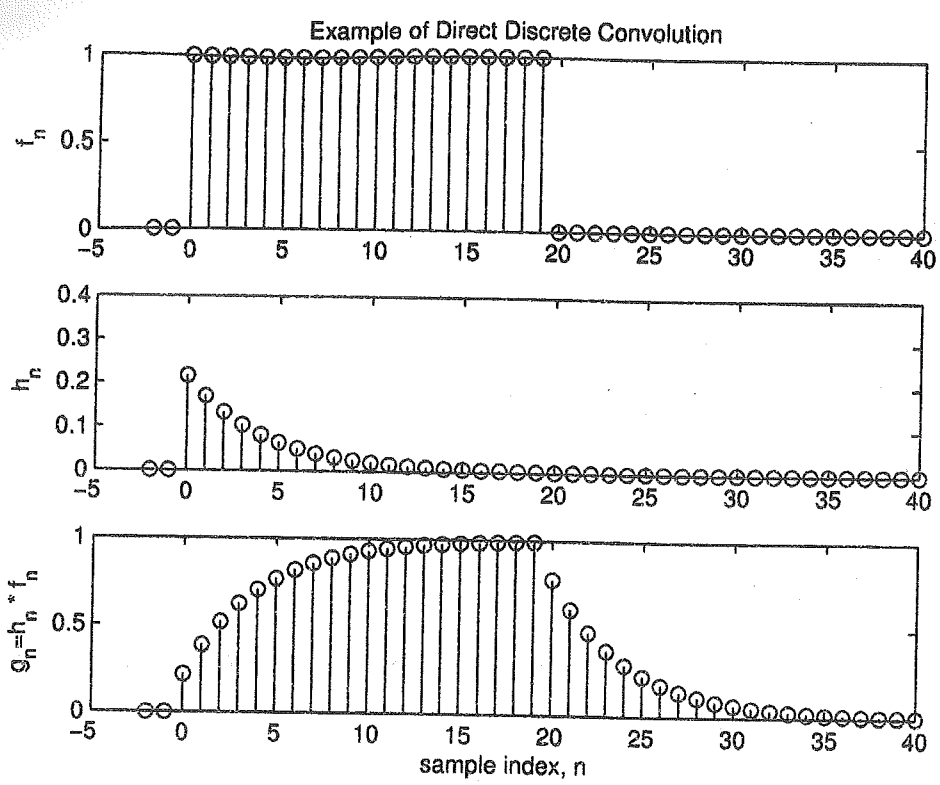


Figure 9.2.1: Direct discrete convolution example.

from a digital LTI system with input f and response h

9.2.1 Properties

Commutative:

$$\{g_n\} = \{f_n\} \star \{h_n\} = \{h_n\} \star \{f_n\}$$

Exercise:

Prove the Commutative property.

Solution:

Duality with the DTFT:

if $\{g_n\} = \{f_n\} \star \{h_n\}$, then $\bar{G}(j\omega) = \bar{F}(j\omega)\,\bar{H}(j\omega)$ where $\bar{F}(j\omega)$, $\bar{G}(j\omega)$ and $\bar{H}(j\omega)$ are the DTFT of f, g and h respectively.

Exercise:

Prove the duality with the DTFT property.

Solution:

9.2.2 Finite length convolution

All of the above is based on the fact that the sequences are infinitely long, and the transformation (i.e. the DTFT) are also infinite sums. However typically we are dealing with finite length signal, so we should really consider these in more detail.

Notation used here:

Let $\{f_n\}$ have length N_1 (starting at $n = 0$) and
Let $\{h_n\}$have length N_2

Exercise:

Show that $\{g_n\} = \{f_n\} \star \{h_n\}$ has length $N_1 + N_2 - 1$.

Solution:

Connection to Multiplication

It is sometime said that convolution is a type of multiplication, and hence the similar $\star$ notation. One of the areas where this is particularly true is that of polynomial multiplication,

consider for example:

$$\begin{aligned} f(x) &= a.x^2 + b.x + c, \quad \text{and} \\ h(x) &= d.x^2 + e.x + f \end{aligned}$$

- Try computing $g(x) = f(x) \times h(x)$ and group the terms into their respective powers of x.
- Now just consider the direct discrete convolution of the two sequences made up from the polynomial coefficients $f_n = \{a, b, c\}$ and $h_n = \{d, e, f\}$, i.e. compute $g = f \star h$
- What do you notice?

9.2.3 No duality with the DFT !

The convolution theorem that was true for the DTFT is, sadly, **not** the true for the finite length DFT. i.e. the product of two DFTs is **not** the DFT of the convolution of the time domain versions!! However we can define a different kind of discrete convolution where this property does hold.

9.3 Periodic discrete convolution

Before defining the periodic discrete convolution we must define what we mean by the *periodic extension* of a signal.

9.3.1 Periodic extension

Suppose f_n is zero outside the range $n = 0, 1, \cdots, N-1$, i.e. f_n is of length N samples. Then the periodic extension of f_n , which is written as $\mathring{f}_n$, is defined as:

$$\mathring{f}_{n+lN} \triangleq f_n \text{ for } \begin{array}{l} 0 \leq n \leq N-1 \text{ and} \\ l = 0, \pm 1, \pm 2, \cdots \end{array}$$

This is an infinitely long signal.

Now we ready to define periodic discrete convolution.

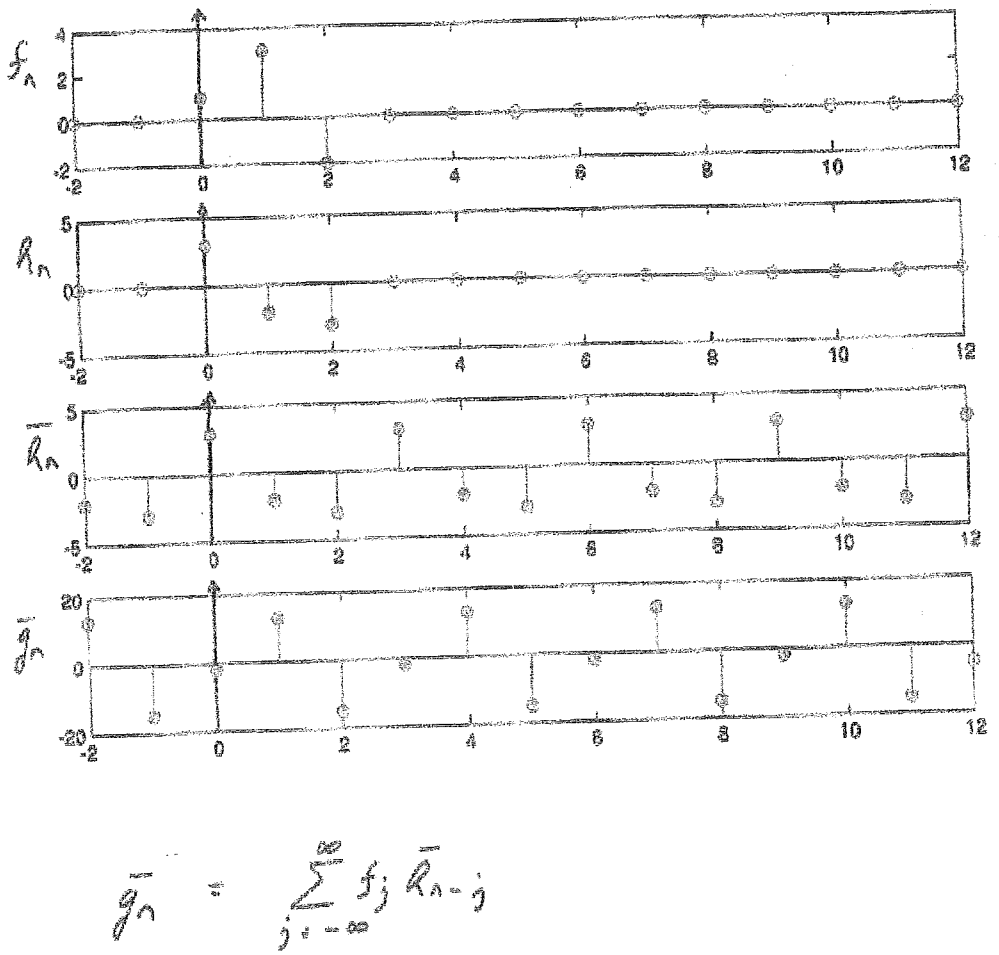


Figure 9.3.1: Periodic discrete convolution example.

9.3.2 Periodic discrete convolution

The periodic discrete convolution is defined as the discrete convolution of one signal with the periodic extended version of the other:

$$\begin{aligned} \{f_n\} \circledast \{h_n\} &\triangleq \{f_n\} \star \left\{\mathring{h}_n\right\} \\ &= \sum_{k=-\infty}^{+\infty} f_k \mathring{h}_{n-k} \end{aligned}$$

This is shown in Figure 9.3.1.

Whilst the definition of $\{f_n\} \circledast \{h_n\}$ is valid for any length signals $\{f_n\}$ and $\{h_n\}$, many of the useful properties of this type of convolution are only valid if they are both the same length, N, and care is usually taken to ensure that this is usually the case.

Exercise:

Prove that if $\{f_n\}$ and $\{h_n\}$ have equal length N then $\{g_n\} = \{f_n\} \circledast \{h_n\}$ is periodic with length N.

Solution:

Exercise:

Prove that if $\{f_n\}$ and $\{h_n\}$ have equal length N then $\{g_n\} = \{f_n\} \circledast \{h_n\} = \{h_n\} \circledast \{f_n\}$.

Solution:

Exercise:

Prove that if $\{f_n\}$ and $\{h_n\}$ have equal length N then the DFT of their periodic convolution is the product of the respective DFTs, i.e.:

$$\bar{G}_m = \bar{F}_m \bar{H}_m \quad \text{for } 0 \leq m \leq N-1$$

where $\bar{F}$, $\bar{H}$ and $\bar{G}$ are the length N DFTs of f, h and g respectfully[1]

Solution:

Hint: This (I think) is the most difficult proof we'll do in this course.
You will most likely need the formula in appendix A.1:

$$\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} = \begin{cases} N & \text{when } k \text{ is a multiple of } N \\ 0 & \text{elsewhere} \end{cases}$$

[1] Of course g is infinitely long (and periodic with period N), so really we mean to say that $\bar{G}$ is the DFT of the signal $\{g_n\}$ for $0 \leq n \leq N-1$.

Exercise

Prove that if $\{f_n\}$ and $\{h_n\}$ have equal length N then the DFT of their product is the periodic convolution of their DFTs scaled by $\frac{1}{N}$, i.e. if

$$g_n = f_n h_n \quad \text{for } 0 \leq n \leq N-1$$

then

$$\{\bar{G}_m\} = \frac{1}{N}\left(\{\bar{F}_m\} \circledast \{\bar{H}_m\}\right)$$

Solution:

9.3.3 Duality

Again we see a nice duality[2]:

$$g_n = f_n h_n \quad \Longleftrightarrow \quad \{\bar{G}_m\} = \frac{1}{N}\left(\{\bar{F}_m\} \circledast \{\bar{H}_m\}\right)$$

and

$$\{g_n\} = \{f_n\} \circledast \{h_n\} \quad \Longleftrightarrow \quad \bar{G}_m = \bar{F}_m \bar{H}_m$$

[2]The $\frac{1}{N}$ factor raises its ugly head again!

Chapter 10

Spectrum estimation

10.1 Introduction

Here we look at some results from spectrum estimation theory. They are not derived here, just presented as methods you can use.

10.2 Energy Spectra

Signal with finite energy can be characterized in the frequency domain by their energy spectra (energy Spectral Density).
For infinitely long discrete we have the Fourier transform of the sampled signal, i.e. the DTFT:

$$\bar{F}(j\omega) = \sum_{n=-\infty}^{+\infty} f_n e^{-jn\omega T}$$

We are interested in the energy spectrum $S_f(\omega) \triangleq \left|\bar{F}(j\omega)\right|^2$.
(the subscript indicates which signal the spectrum refers to).
Clearly we can't actually compute this infinite sum, so we define $\tilde{S}(\omega)$ as an estimate of the energy spectrum computed using just a finite number of elements:

$$\tilde{S}_f(\omega) \triangleq \left|\sum_{n=0}^{N-1} f_n e^{-jn\omega T}\right|^2$$

Note: If the signal is actually finite in duration and completely contained within the range of the above summation, then approximation is exact, i.e. $\tilde{S}_f(\omega) = S_f(\omega)$.

A difficulty with the above formulation is that ω is a continuous variable and we the computation of $\tilde{S}_f(\omega)$ in a digital machine is impossible. Thus we typically only compute a sampled version of $\tilde{S}_f(\omega)$ using the DFT instead, i.e.

$$\left(\tilde{S}_f\right)_m = \left|\bar{F}_m\right|^2$$

10.3 Power Spectra

Infinitely long random signal do not have finite energy and can only be characterized in the frequency domain by their power spectra (a.k.a. Power Spectral Density, PSD).

In continuous systems, the PSD $P_x(\omega)$ of a signal $x(t)$ is defined in such a way to ensure that its integral over any frequency band is the average power in the signal over that frequency band:

$$\int_A^B P_x(\omega)\, d\omega = \text{average power between A and B rads/sec}$$

The word "average" here is important as it is a random signal and each instance of $x(t)$ will be different. $P_x(\omega)$ is not itself random, it is a statistical description of all possible $x(t)$. The precise definition of $P_x(\omega)$ is given by:

$$P_x(\omega) = \int_{-\infty}^{+\infty} E\left[x(t)\, x^*(t-\tau)\right] e^{-j\omega\tau} d\tau$$

i.e. the Fourier transform of the auto-correlation function.
(E here is the expected value operator).

10.4 PSD estimation techniques

As the definition of $P_x(\omega)$ contains an integration over infinite time it is clear that exact computation is not in general possible; however there are techniques that can be used to estimate it. Broadly speaking these can be catheterized into either a) parametric or b) non-parametric techniques.

Parametric techniques:

If you have a parameterizable model for the random signal then a parametric techniques should be used to estimate the parameters and hence the PSD. For example: suppose you (somehow) know that your signal consists of a single sine wave having unknown amplitude, frequency and phase plus white noise, i.e. the whole of the random signal can be described by only 4 parameters - thus a parametric technique could be used to estimate the unknown 4 parameters, and the complete PSD be constructed from these.
There are many parametric techniques, e.g. Auto-Regressive (AR), Moving-Average (MA), ARMA. In the example given here a possible techniques could be principal component analysis to estimate the parameters of the sine wave followed by a simple RMS calculation to estimate the density of the remaining white noise.

Non-Parametric techniques:

In the absence of a parameterizable model (or if you are lazy) there are many non-parametric (or direct) methods available to PSD estimation. Here some of them are presented without proof. In all cases the time domain signal is $x(t)$ unless otherwise states - hence the x subscript is often dropped.

10.4.1 The Periodogram

Estimate $P_x(\omega)$ using

$$P_x(\omega) \approx P_{periodgram}(\omega) \triangleq \frac{1}{N}\left|\sum_{n=0}^{N-1} x_n e^{-jn\omega T}\right|^2$$

This is basically the same as the energy spectrum with the extra $\frac{1}{N}$ scaling factor (produces an average over time).
There are two important properties of note:

Bias:

It can be shown that the periodogram yields a biased estimate of $P_x(\omega)$, i.e.

$$E[P_{periodogram}(\omega)] = \frac{1}{2\pi} P_x(\omega) \star W_B(\omega)$$

where $W_B(\omega)$ is the spectrum of the length N Bartlett window an example of which is shown in Figure 10.4.1:

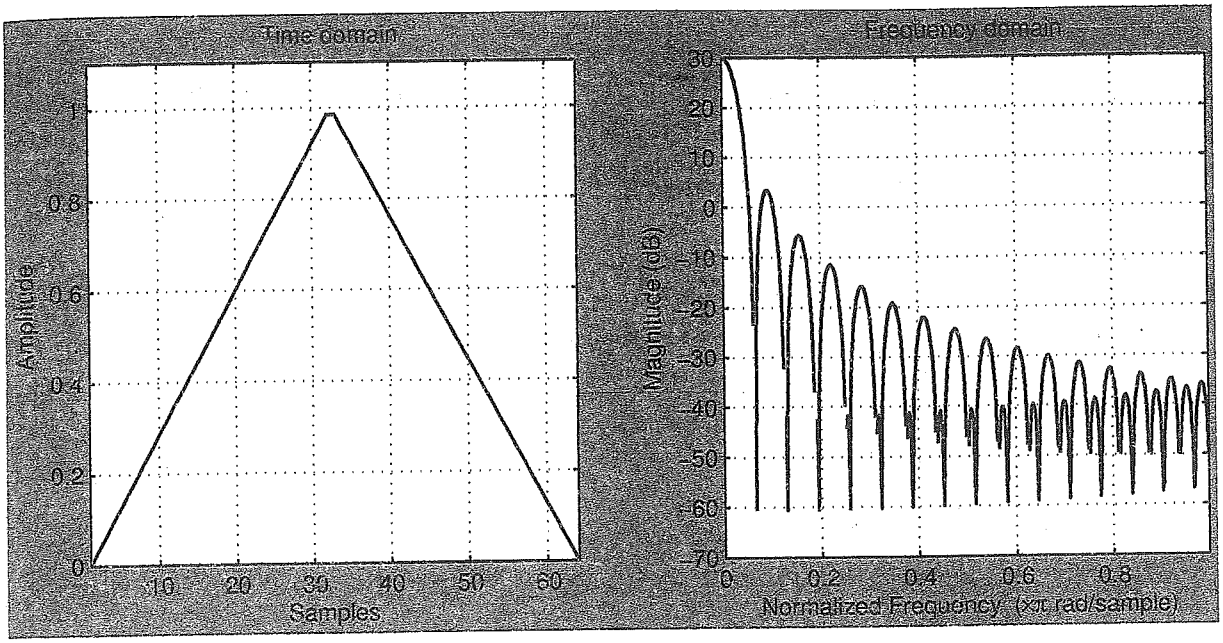


Figure 10.4.1: Length 65 Bartlett window

As $N \to \infty$ then time domain window get wider, and by duality the frequency domain version $W_B(\omega)$ gets narrower (approaches a single delta function) and so the error (bias) becomes smaller.

Variance: It can be shown that the variance of the periodogram is:

$$Var\left[P_{periodgram}(\omega)\right] = P_x^2(\omega)$$

Importantly this is independent of N !!!
i.e. it doesn't matter how many samples you take, the results you obtain are always going to be noisy.
More samples implies finer frequency resolution, but not better estimation accuracy.

10.4.2 The modified periodogram

Based on the periodogram, but instead of just taking samples $n = 0$ to $N-1$, i.e. applying a time domain rectangular window, we could use a different shape window - typically one that has tapered edges.

The exact choice of window allows a trade off between main lobe width and side lobe height, see Figure 10.4.2 for some examples.

Whilst often better than the unmodified periodogram it still has the same problems:

- biased
- variance not reduced with increasing N

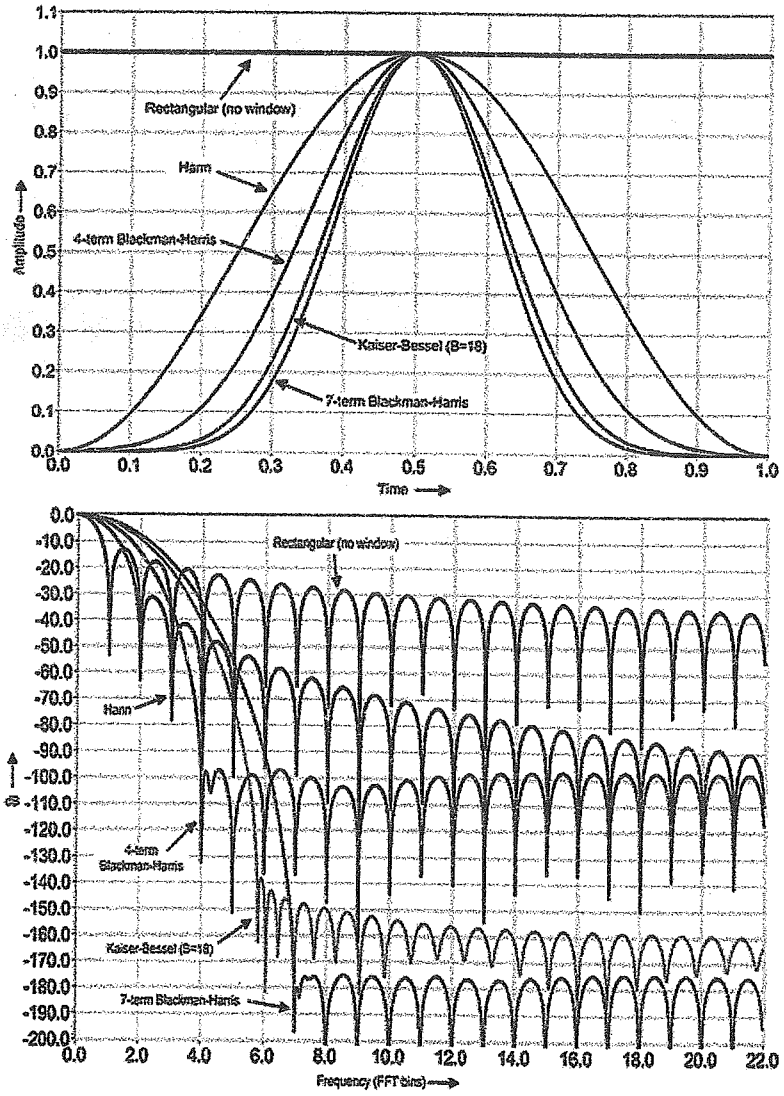


Figure 10.4.2: Some typical windows in the time and frequency domains.

10.4.3 Bartlett's method

Based on the idea of periodogram averaging:

- Divide the length N sequence into K smaller sequences, each of length L, i.e. $N = L.K$
- Using the DFT, compute the periodogram of each of the K smaller sequences
- Average them to obtain final estimate.

This is illustrated in Figure 10.4.3.

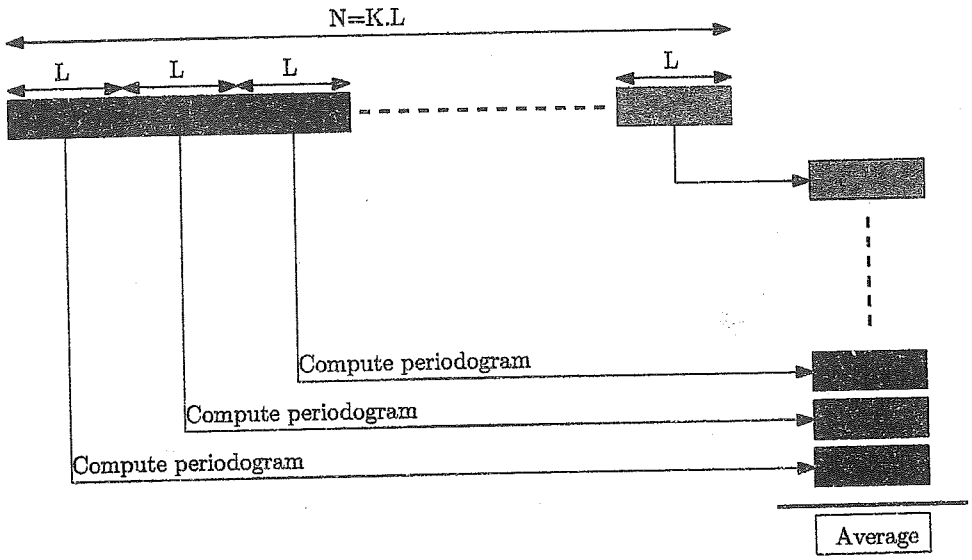


Figure 10.4.3: Illustration of Bartlett's method for PSD estimation.

Bias The bias can be shown to be:

$$E\left[P_{Bartlett}\left(\omega\right)\right] = \frac{1}{2\pi}P_x\left(\omega\right) \star W_B\left(\omega\right)$$

i.e. the same as the periodogram.

Variance the variance can be shown to be:

$$Var\left[P_{Bartlett}\left(\omega\right)\right] = \frac{L}{N}P_x^2\left(\omega\right)$$

This is important, as the variance has been reduced by a factor of $K = \frac{N}{L}$ compared to the previous two methods. If $N \to \infty$ then the variance reduces to zero!

10.4.4 Welch's method

Based on Bartlett's method with two modifications:

- We use overlapping sequences in time
- We use the modified periodogram instead of the regular periodogram, i.e we apply a window to each segment before computing the periodogram.

This is illustrated in Figure 10.4.4.

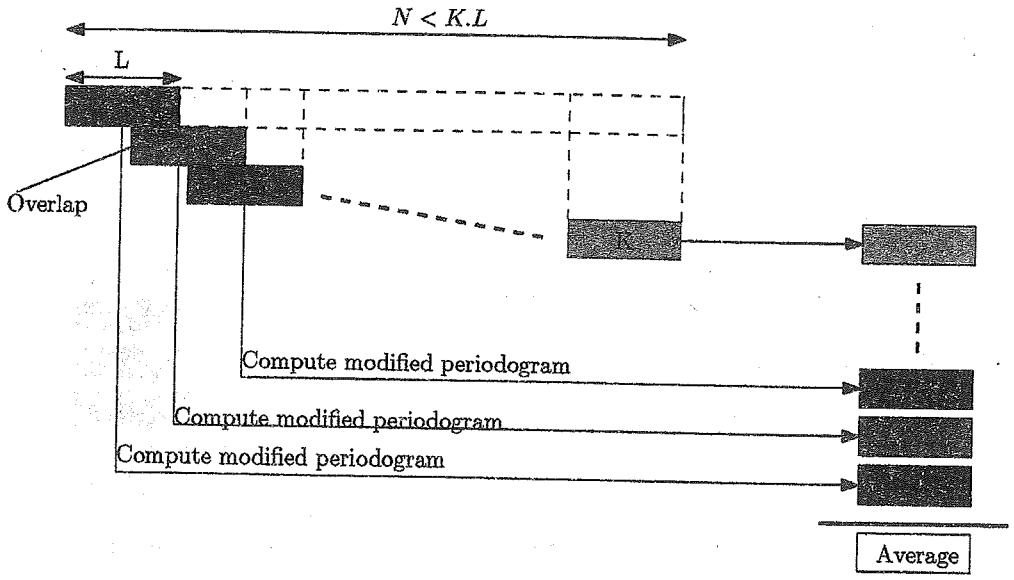


Figure 10.4.4: Illustration of Welch's method for PSD estimation.

The Hanning (Raised Cosine) window is a very popular choice for Welch's method.
With 50% overlapping Hanning windows each samples (apart from some at the start and end of the sequence) all contribute equally to the final estimate.

Bias The bias can be shown to be:

$$E\left[P_{Welch}\left(\omega\right)\right] = \frac{1}{2\pi L.U} P_x\left(\omega\right) \star \left|W\left(\omega\right)\right|^2$$

Where $U = \overline{W_n^2}$ is the average of the sampled window function (a constant for a given window).
Importantly this bias depends on $\frac{1}{L}$, and so can be controlled.

Variance Assuming a 50% overlap, the variance can be shown to be:

$$Var\left[P_{Bartlett}\left(\omega\right)\right] = \frac{9}{16}\frac{L}{N}P_x^2\left(\omega\right)$$

Chapter 11

The z-transform

11.1 Background

For a continuous time systems we had the Fourier and (double sided) Laplace transforms - why both? Well lets look at their definitions:

$$\text{Fourier} \quad F(j\omega) = \int_{-\infty}^{+\infty} f(t)\, e^{-j\omega t} dt$$

$$\text{Laplace} \quad F(s) = \int_{-\infty}^{+\infty} f(t)\, e^{-st} dt$$

Where the Laplace variable $s = \sigma + j\omega$.

The major difference between these is that in the Laplace transform the variable $s = \sigma + j\omega$ is an arbitrary complex number, whereas the Fourier variable ω is purely real. This change allows solutions other than sums of infinite complex sinusoids to exist.

The solutions to steady-state systems can be represented by the sum of complex sinusoids terms e.g. $e^{j\omega t} = \cos\omega t + j\sin\omega t$ for $-\infty < t < +\infty$, but the solution to transient systems requires more elaborate formulations that can't be represented simply by $e^{j\omega t}$ terms, hence the Laplace variable $s = \sigma + j\omega$ includes the extra σ terms thereby permitting solutions to have exponential decaying (or increasing) terms, i.e. $e^{st} = e^{\sigma t}(\cos\omega t + j\sin\omega t)$. Clearly if $\sigma < 0$, then this term $\to 0$ as $t \to \infty$ and the solution converges to some steady state conditions (the transient effect disappears).

11.1.1 Example RC circuit

Consider the RC circuit in Figure 11.1.1.

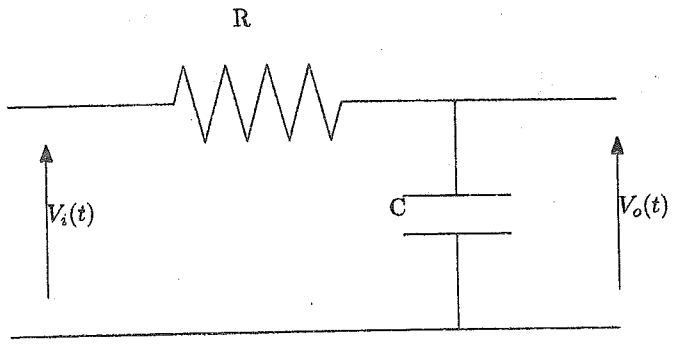


Figure 11.1.1: Simple RC circuit

The Fourier analysis of this circuit allows use to compute the output (infinitely long) sinusoid when presented with an (infinitely long) sinusoid input:

$$\begin{aligned} v_i(t) &= \sin\omega t \quad -\infty < t < +\infty \\ \Rightarrow v_o(t) &= \frac{1}{1+(RC\omega)^2}(\sin\omega t + RC\omega\sin\omega t) \quad -\infty < t < +\infty \end{aligned}$$

However what if the input $v_i(t) = v_0(t) = 0$ for $t < 0$, i.e. the circuit is "turned on" at time $t = 0$, and all voltages are assumed to be zero prior to that time?, i.e.

$$v_i(t) = \begin{cases} 0 & t < 0 \\ \sin\omega t & t > 0 \end{cases}$$

Well Fourier analysis no longer works.

Laplace analysis yields the following result:

$$v_o(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{1+(RC\omega)^2}\left[RC\omega e^{-\frac{t}{RC}} + (\sin\omega t - RC\omega\cos\omega t)\right] & t > 0 \end{cases}$$

as illustrated in Figure 11.1.2.

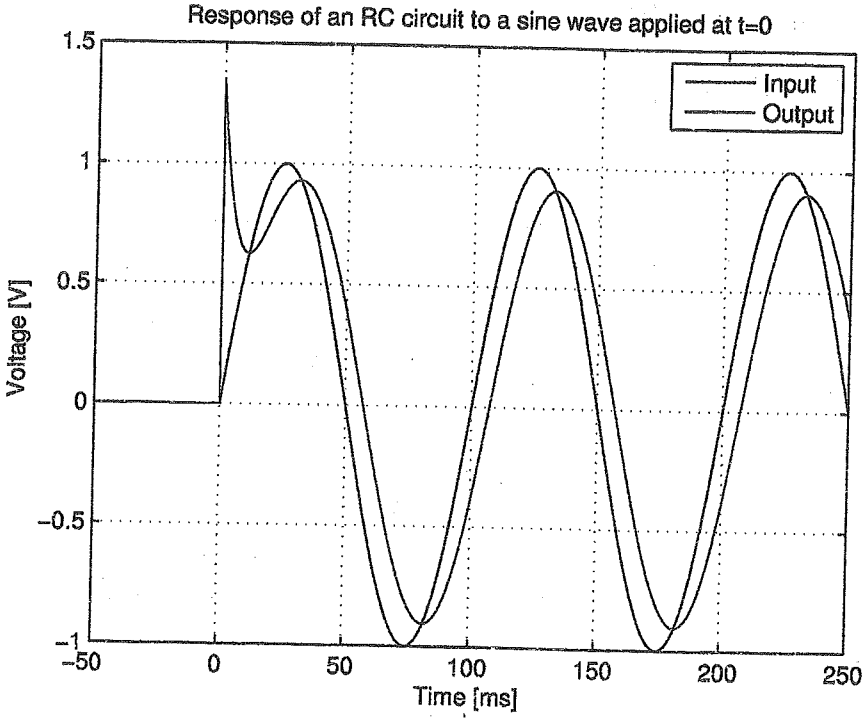


Figure 11.1.2: Result from analysis of a simple RC circuit

Note that as $t \to \infty$, the above solution becomes:

$$\lim_{t\to\infty} v_o(t) = \frac{1}{1+(RC\omega)^2}\left(\sin\omega t - RC\omega\cos\omega t\right)$$

exactly as per the Fourier analysis!

11.1.2 Discrete time systems

In discrete time systems we have a similar situation. We've already seen the DTFT which is the Fourier transform of a sampled signal. Now we introduce the *z-transform* which is basically the Laplace transform of a sampled signal.

11.2 (Two-sided) z-transform

First we'll provide the definition and then we'll show that this is indeed the Laplace transform of the sampled signal. The definition is:

$$\tilde{F}(z) \triangleq \mathcal{Z}(\{f_n\}) \triangleq \sum_{n=-\infty}^{+\infty} f_n z^{-n} \tag{11.2.1}$$

You must remember this formula - it is very very important!!!

11.2.1 Relationship to the Laplace transform

Consider the impulse sampled signal:

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

(remember the DTFT was the Fourier transform of $\bar{f}(t)$).

Exercise:

Prove that the (two-sided) Laplace transform of $\bar{f}(t)$ with the change of variable $z = e^{sT}$ is the z-transform.

Solution:

So we've shown that the (two-sided) z-transform of the discrete signal $\{f_n\}$ is the (two-sided) Laplace transform of the impulse sampled signal $\bar{f}(t)$, with the small notation change $z = e^{sT}$.

The notation change from s to $z = e^{sT}$ is really just for mathematical convenience but it does turn out to be very useful and is used throughout all of Digital Signal Processing.

11.2.2 Sequence becomes a polynomial

The z-transform turns a sequence $\{f_n\}$ (i.e. a collection of samples) into a polynomial by attaching a unique power of z^{-1} to each sample and adding all the terms together, see Figure 11.2.1 for an illustration of this.

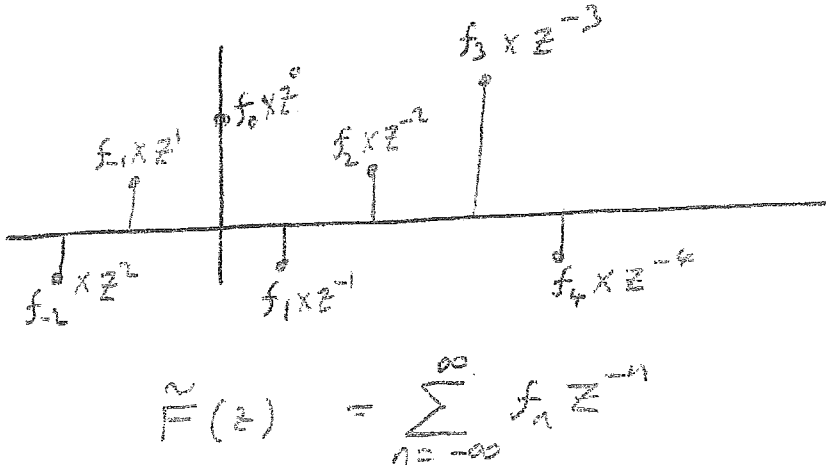


Figure 11.2.1: Example of how the z-transform makes a polynomial from a collection of samples

Note how the z-transform is independent of the sampling interval T, so the following two signal have the same z-transform:

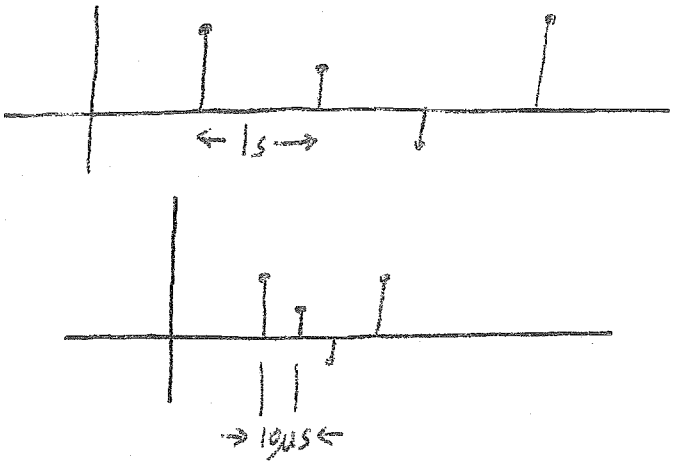


Figure 11.2.2: The z-transform of the two sequence above are the same despite the different sampling internal.

11.3 Examples

11.3.1 Exponential decay exercise:

If $\{f_n\}$ is an exponential decay starting at time index $n = 0$, i.e.

$$f_n = \begin{cases} c^n & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

Show that the z-transform is:

$$\tilde{F}(z) = \frac{1}{1 - cz^{-1}} \quad |z| > |c|$$

Solution:

Hint:You will need to make use of the geometric series:

$$\sum_{k=0}^{N-1} r^k = \frac{1 - r^N}{1 - r}$$

Figure 11.3.1 illustrates what is called the Region Of Convergence (RoC) for the z-transform of this signal.

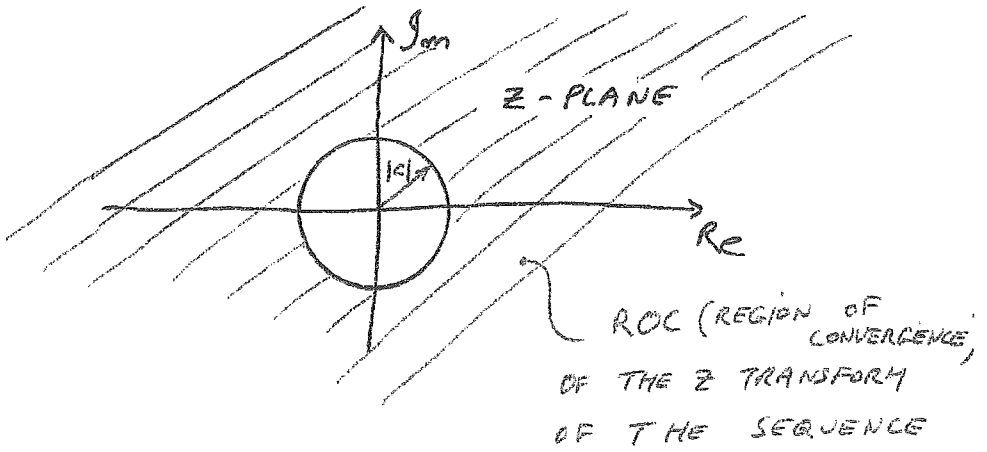


Figure 11.3.1: The RoC for the z-transform of signal $f_n = \begin{cases} c^n & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$

11.3.2 More z-transform exercises

Verify the following z-transforms:

Name	time domain	z-transform	RoC
Exponential decay	$f_n = \begin{cases} c^n & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$	$\frac{1}{1-cz^{-1}}$	$\|z\| > \|c\|$
Unit step	$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
Unit impulse	$\delta_n = \begin{cases} 1 & n = 0 \\ 0 & \text{elsewhere} \end{cases}$	1	everywhere
Delayed unit impulse	$\delta_{n-N} = \begin{cases} 1 & n = N \\ 0 & \text{elsewhere} \end{cases}$	z^{-N}	everywhere

Table 11.1: Some z-transform pairs

These are shown in Figure 11.3.2.

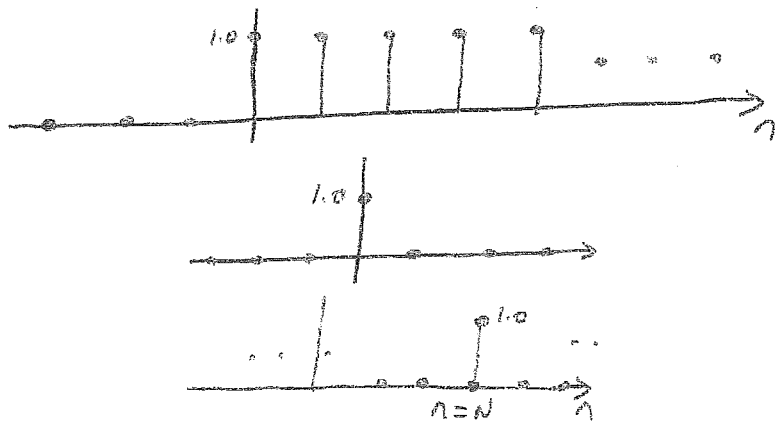


Figure 11.3.2: From top to bottom, (a) the unit step, (b) the unit impulse, and (c) the delayed unit impulse

Solution:

Exponential decay:

Already done above in Section 11.3.1.

$$\tilde{F}(z) = \frac{1}{1 - cz^{-1}} \quad |z| > |c|$$

Unit step:

Unit impulse:

Delayed unit impulse:

11.4 Properties of the z-transform

11.4.1 Linear

The z-transform is linear iff the following are true:

$$\begin{aligned} \mathcal{Z}(\lambda\{f_n\}) &= \lambda\tilde{F}(z) \quad \forall\lambda \in \mathbb{C} \text{ and} \\ \mathcal{Z}(\{f_n + g_n\}) &= \tilde{F}(z) + \tilde{G}(z) \end{aligned}$$

Exercise:

Show that the z-transform is linear.

Solution:

Note: A useful consequence of all linear transforms is that weighted sums of signals are easy to compute, i.e.:

$$\mathcal{Z}\left(\{a.f_n + b.g_n\}\right) = a\tilde{F}(z) + b\tilde{G}(z)$$

Which can be very very useful.

11.4.2 Shift theorem

Exercise:

If $\tilde{F}(z)$ is the z-transform of $\{f_n\}$, show the z-transform of $\{f_{n-K}\}$ is $z^{-K}\tilde{F}(z)$

Solution:

It is because of the above theorem that z^{-1} is often called the *delay operator*, i.e. just multiply by z^{-1} every time you want to delay a signal by one sample; in fact in the DSP world we often represent a signal delay line as shown in Figure. 11.4.1

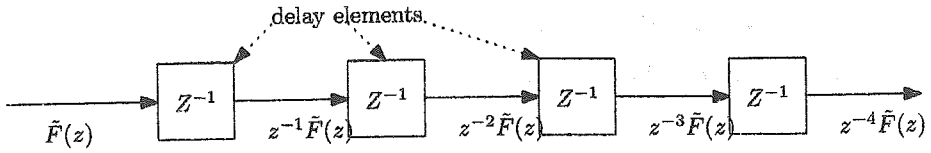


Figure 11.4.1: A delay line is often drawn with z^{-1} written in each delay element.

11.4.3 Relationship to the DTFT

The importance of this is that, once we know the z-transform then we can easily compute the DTFT. We also note the the DC ($\omega = 0$) response can be computed by letting $z = e^0 = 1$, the response at $\frac{1}{2}f_s$ can be computed by letting $z = e^{j2\pi\frac{f_s}{2}T} = e^{j\pi} = -1$, these along with a couple of others are shown in Figure 11.4.2.

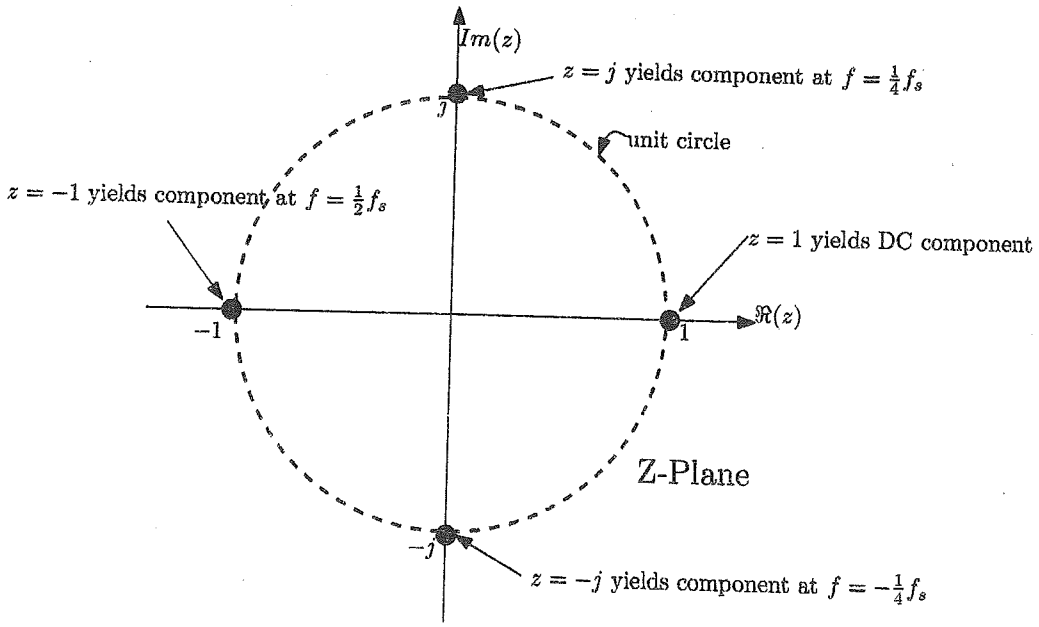


Figure 11.4.2: The DTFT is the z-transform evaluated along the unit circle in the z-plane

Finally we note that this is only true in general of the two-sided z-transform[1] (as the DTFT is also two sided).

[1] Well it is true for the single sided z-transform if the signal itself is causal, i.e. zero for sample index $n < 0$.

11.4.4 Convolution

Exercise:

Show that if $\{g_n\} = \{f_n\} \star \{h_n\}$, then $\tilde{G}(z) = \tilde{F}(z)\tilde{G}(z)$, where $\star$ denotes direct digital convolution.

Solution:

Duality:

Again we see the usual duality, i.e. convolution in one domain corresponds to convolution in the other.

11.5 Inverse z-transform

Formally the inverse z-transform is given (without proof) by:

$$f_n = \frac{1}{2\pi j} \oint_C z^{n-1} \tilde{F}(z)\, dz$$

where C is any counterclockwise closed contour in the z-plane encircling the origin within the Region Of Convergence (RoC).

We rarely use this form of the inverse z-transform. For almost all practical situations there are usually much easier techniques that can be applied.

Chapter 12

Introduction to Digital Filtering

12.1 Introduction

Digital filters can be used to filter analog signals provided a mechanism is provided to sample the input, and reconstruct the output as shown in Figure 12.1.1.

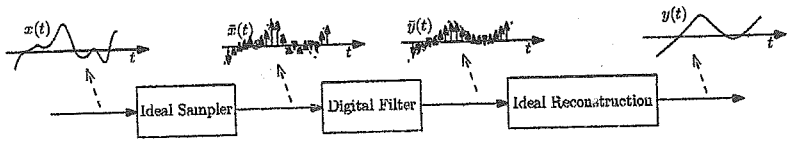


Figure 12.1.1: Example system where an analog signal is sampled, filtered digitally and then reconstructed.

Digital filters can be constructed to designed to do all sorts of things, for example the it can be used to smooth out a noisy signal by essentially removing high frequency components - a low pass filter. An example of such a filter in operation is illustrated in Figure 12.1.2.

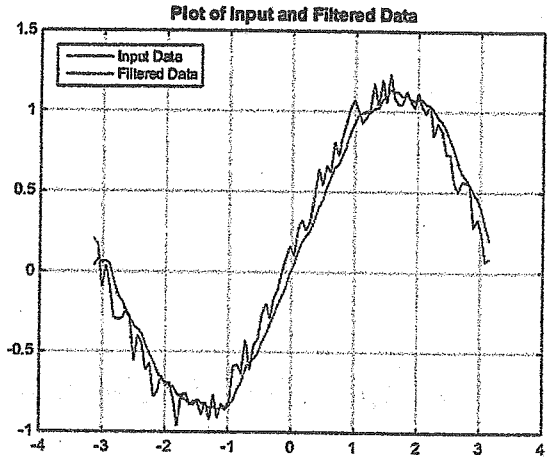


Figure 12.1.2: Example input and output signals for a low pass filter - note the delay. (picture comes from Mathwoks).

In this chapter we will look at a generic equation describing the time domain relationship between the input and output of a digital filter. We will then examine some of it's properties and consider it's z-domain representation.
This will server as a good basis for more advance filter design and analysis later in the course.

12.2 Digital filter equation

A large class of linear, time invariant, single input single output (SISO), discrete time systems can be described by the equation:

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} - \sum_{l=1}^{L} a_l y_{n-l} \qquad (12.2.1)$$

This is a Difference Equation and is the basis of digital filtering.
The values (sometimes known as coefficients) of $\{a_n\}$ and $\{b_n\}$ are constants, i.e. the system is time invariant.
$\{x_n\}$ is the input, and
$\{y_n\}$ is the output, i.e. the filtered signal.
An obvious way to implement such a filter is shown in Figure 12.2.1.

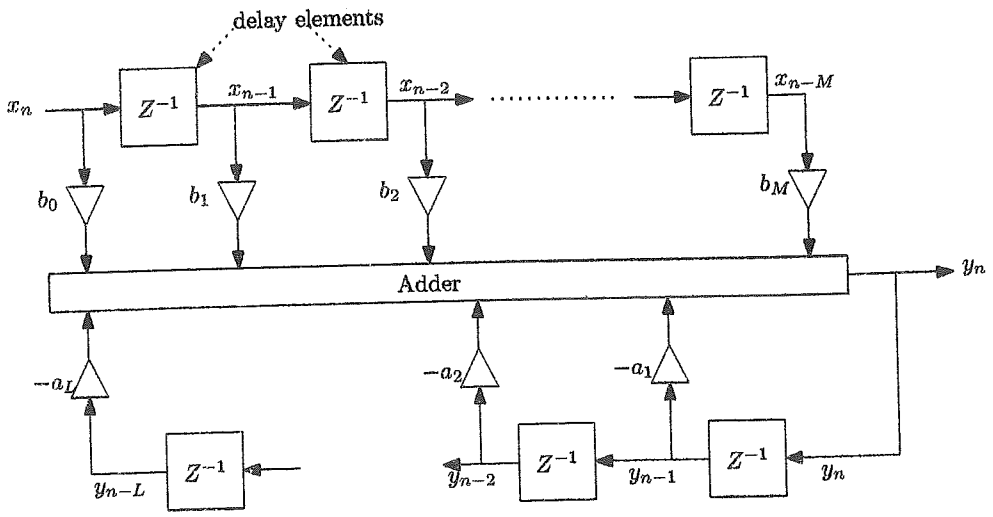


Figure 12.2.1: Direct form implementation of equation 12.2.1

By adjusting the $\{a_n\}$ and $\{b_n\}$ coefficients filters can be altered to perform different types of filtering actions. This ability obviously offers huge flexibility for the one piece of hardware of software to be easily reconfigured to suit many tasks and is one of the many reasons why digital signal processing had before so widespread today.

12.2.1 Linear

Exercise:

Show that the z-transform is linear.
i.e. if inputs $\{x_n\}$ and $\{w_n\}$ produces outputs $\{y_n\}$ and $\{v_n\}$ respectively, then
(a) input $\lambda\{x_n\}$ produces output $\lambda\{y_n\}$, and
(b) input $\{x_n + w_n\}$ produces an output $\{y_n + v_n\}$

Solution:

12.2.2 Time invariant

Exercise:

Show that the z-transform is time invariant.

i.e. if input $\{x_n\}$ produces output $\{y_n\}$, then input $\{x_{n-K}\}$ produces output $\{y_{n-K}\}$.

Solution:

12.3 Transfer function

Just as the Laplace transform was used in continuous time systems to get a transfer function of the system, here we'll use the z-transform. If

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} - \sum_{l=1}^{L} a_l y_{n-l}$$

Then we define the transfer function as:

$$\tilde{H}(z) \triangleq \frac{\tilde{Y}(z)}{\tilde{X}(z)}$$

allowing us to write the input / output relationship simply as:

$$\tilde{Y}(z) = \tilde{H}(z)\tilde{X}(z)$$

You should be able to see clear parallel here with the Laplace transform descriptions of continuous time signals and systems.

Exercise:

Show that the transfer function is simply the ratio of the z-transform of $\{b_k\}$ and $\{a_k\}$ with the additional definition that $a_0 = 1$.

Solution:

12.3.1 Ratio of polynomials

We can write the transfer function out "long-hand":

$$\tilde{H}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L}} \tag{12.3.1}$$

i.e. this is just a ratio of two polynomials, a Numerator $\tilde{B}(z)$ and a Denominator $\tilde{A}(z)$.

12.3.2 Poles and zeros

The poles and zeros of $\tilde{H}(z)$ are defined as:

- **Zeros**: The points in the z-plane where $\tilde{H}(z) = 0$.
- **Poles**: The points in the z-plane where $\tilde{H}(z)$ has a divide by 0 (i.e. it explodes to infinity).

Note, as defined, $\tilde{B}(z)$ only contains negative powers of z, so we can factorize it as follows:

$$\begin{aligned} \tilde{B}(z) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M} \\ &= z^{-M}\left(z^M b_0 + b_1 z^{M-1} + b_2 z^{M-2} + \cdots + b_M\right) \end{aligned}$$

i.e. a leading z^{-M} term and a polynomial in positive powers of z. From this form we can now factorize it again in a collection of roots in the z plane:

$$\tilde{B}(z) = b_0 z^{-M}(z - z_1)\ldots(z - z_M) = \prod_{m=1}^{M}(z - z_m)$$

where the z_m are the positions in the z plane (not the z^{-1}plane) where $\tilde{B}(z)$ is zero, i.e. the zeros of $\tilde{B}(z)$. Also we note that there are M poles of $\tilde{B}(z)$ at position $z = 0$.
Likewise for the denominator:

$$\tilde{A}(z) = z^{-L}(z - p_1)\dots(z - p_L) = \prod_{l=1}^{L}(z - p_l)$$

(note as $a_0 = 1$ it is not shown above above).
Combining these we have:

$$\tilde{H}(z) = b_0 z^{L-M} \frac{\prod_{m=1}^{M}(z - z_m)}{\prod_{l=1}^{L}(z - p_l)} \tag{12.3.2}$$

We see that at $z = 0$ there is either $L - M$ zeros if $L > M$, or $M - L$ poles if $M > L$.

Assuming $\tilde{H}(z)$ is irreducible, we can say that apart from the poles or zeros at $z = 0$, there are M zeros ad L poles.

You should be easily be able to verify that

- If $z = z_m$ (for any m), then the numerator is $= 0$ and consequently $\tilde{H}(z) = 0$.
- Likewise, if $z = p_l$ for any l then the denominator is $= 0$ and we have a divide by zero situation which normally results in $\left|\tilde{H}(z)\right| = \infty$.

12.3.2.1 Pole-zero diagram

The position of the poles and zeros of a filter have a huge impact on both the frequency response and stability and form the basis of much analysis and synthesis of digital filters. Often we like to visualize them on a pole-zero diagram like the one shown in Figure 12.3.1.
Notation:

- A "O" is used to represent a zero, i.e. where $\tilde{H}(z) = 0$, and
- A "X" is used to represent a pole.

For clarity we usually don't draw the poles or the zeros at the origin because a) they are always there for causal filters and just serve to confuse the issue, and b) it can be shown that they don't affect the magnitude response nor the stability (however they do effect the

phase response by adding or subtracting a fixed group delay - see the shift property of the z-transform).

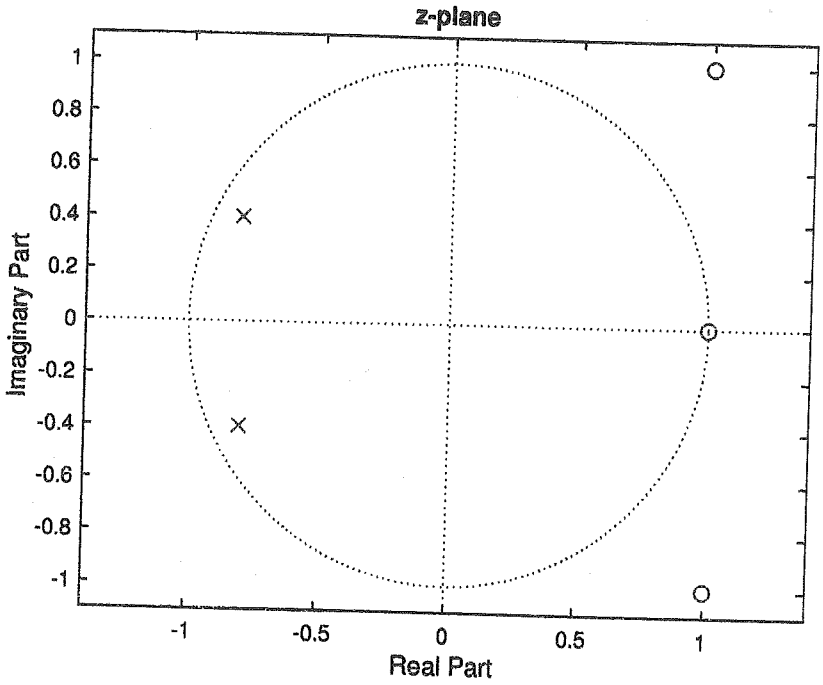


Figure 12.3.1: Example pole-zero diagram. Here we have two poles (the x's) and three zeros (the o's). The "extra" zero at the origin is not shown.

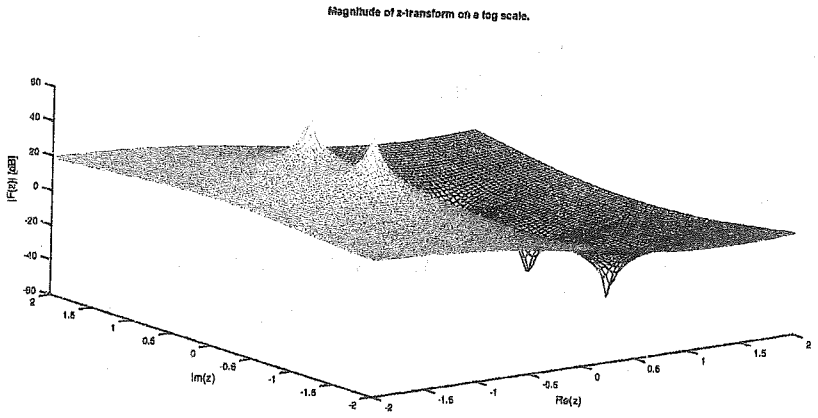


Figure 12.3.2: The magnitude of the z-transform of the system shown in Figure 12.3.1

12.3.3 Polynomials with Real coefficients

Exercise:

Show that if the coefficients of *any* polynomial are real valued then the roots are either real or they occur in conjugate pairs.

Solution:

Theorem. *If the coefficients of* any *polynomial are real valued then the roots are either real or they occur in conjugate pairs.*

12.3.4 Filters with real coefficients

Given that the transfer function of a digital filter is the ratio of two polynomials, the coefficients of which are just the filter coefficients themselves, i.e.:

$$\tilde{H}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L}}$$

We can apply the above theorem and state that:

If the filter coefficients are all real valued, then the poles and zero must either be real valued or occur in conjugate pairs.

This is illustrated in Figure 12.3.1, where there are the 2 poles that occur as a conjugate pair (at $z = -0.8 \pm j0.4$), and the 3 zeros are composed of 1 real valued zero (at $z = 1$), and 2 others which occur as a conjugate pair (at $z = 1 \pm j$). Thus we can say with certainty that the coefficients are this filter are all real valued.

12.4 Frequency Response

12.4.1 Digital filtering of an analog signal

As mentioned in the introduction to this chapter it is possible to filter an analog signal using a digital filter.

Of course we would like to know the relationship between the analog transfer function $H(j\omega)$ and the digital transfer function just derived. We can derive this relationship graphically by looking at Figure 12.4.1.

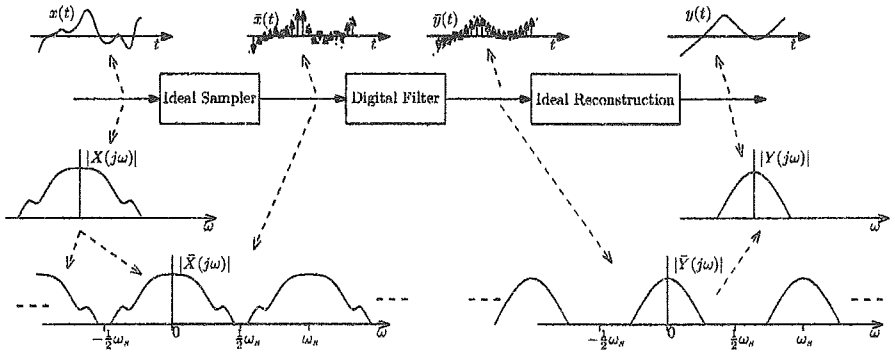


Figure 12.4.1: Example system where an analog signal is sampled, filtered digitally and then reconstructed.

The analog transfer function $H(j\omega)$ is defined as:

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} \quad \forall\omega$$

But, from the Figure 12.4.1 we can see that[1] on the range $-\frac{1}{2}\omega_s \leq \omega \leq \frac{1}{2}\omega_s$ the $H(j\omega)$ should be equal to the ratio of the sampled spectra, i.e.:

$$H(j\omega) = \frac{\bar{Y}(j\omega)}{\bar{X}(j\omega)} \quad -\frac{1}{2}\omega_s \leq \omega \leq \frac{1}{2}\omega_s$$

[1] Provided the sampling is higher than the Nyquist rate and no aliasing has occurred.

But we previously established that $\bar{Y}(j\omega)$, being the Fourier transform of $\bar{y}(t)$ can be obtained from the z-transform of $\{y_n\}$, i.e. $\tilde{Y}(z)$, if we evaluate it along the unit circle, i.e. when $z = e^{j\omega T}$ (see Section 11.4.3). Likewise for $\tilde{X}(z)$, thus we have:

$$H(j\omega) = \left.\frac{\tilde{Y}(z)}{\tilde{X}(z)}\right|_{z=e^{j\omega T}} \quad -\frac{1}{2}\omega_s \leq \omega \leq \frac{1}{2}\omega_s$$

or more compactly we have:

$$H(j\omega) = \tilde{H}\left(z = e^{j\omega T}\right) \quad -\frac{1}{2}\omega_s \leq \omega \leq \frac{1}{2}\omega_s$$

Notation Based on the above discussion, when we talk of the discrete system's **frequency response**, we mean $\tilde{H}\left(z = e^{j\omega T}\right)$ irrespective if there is an analog signal involved or not!

12.4.2 Periodic

Recap: The frequency response is now defined as

$$\tilde{H}\left(z = e^{j\omega T}\right) = \frac{\tilde{Y}\left(z = e^{j\omega T}\right)}{\tilde{X}\left(z = e^{j\omega T}\right)} \text{ or } \frac{\tilde{B}\left(z = e^{j\omega T}\right)}{\tilde{A}\left(z = e^{j\omega T}\right)}$$

Which is the ratio of two DTFTs, each of which is periodic, and therefore the frequency response is also periodic as shown in Figure 12.4.2.

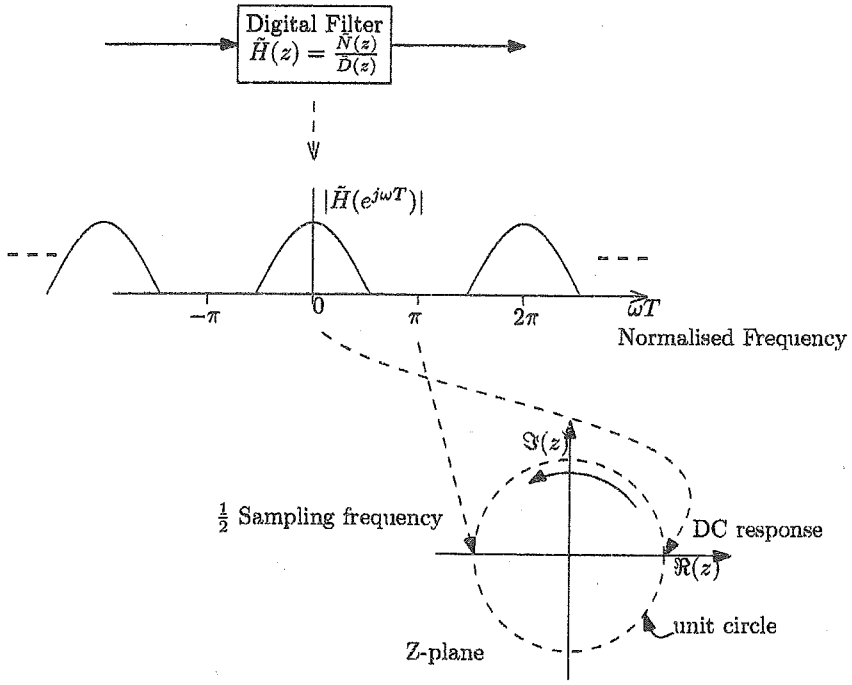


Figure 12.4.2: The frequency response, $\tilde{H}\left(z = e^{j\omega T}\right)$, of a digital filter is periodic. This is due to the wrapping around and around the unit circle.

Chapter 13

Finite Impulse Response (FIR) filters

13.1 Basic definition

Finite Impulse Response (FIR) filters, also known as non-recursive filters, are used in almost every DSP application. If we let all the $\{a_k\}$ coefficients in equation 12.2.1 be zero we get:

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} \tag{13.1.1}$$

i.e. the current output y_n is a weighted sum of the current and the M most recent inputs[1], as shown in Figure 13.1.1.

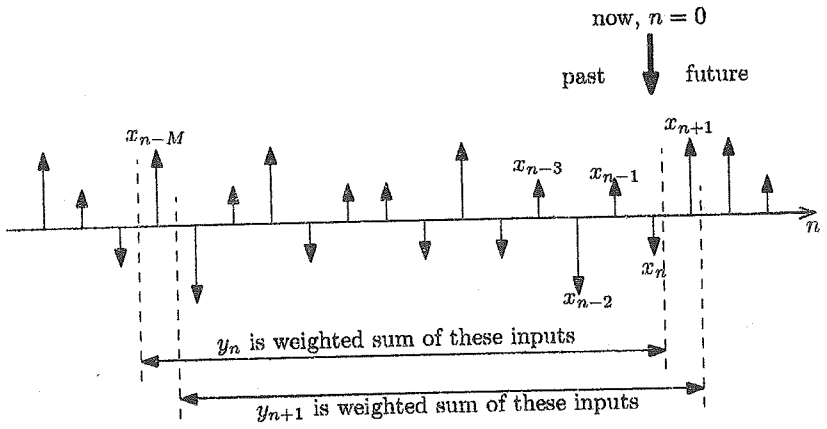


Figure 13.1.1: Each FIR output is a weighted sum current and previous M inputs.

Equation 13.1.1 can be implemented using a structure like the one shown in Figure 13.1.2.

[1]This is also called a weighted moving average process.

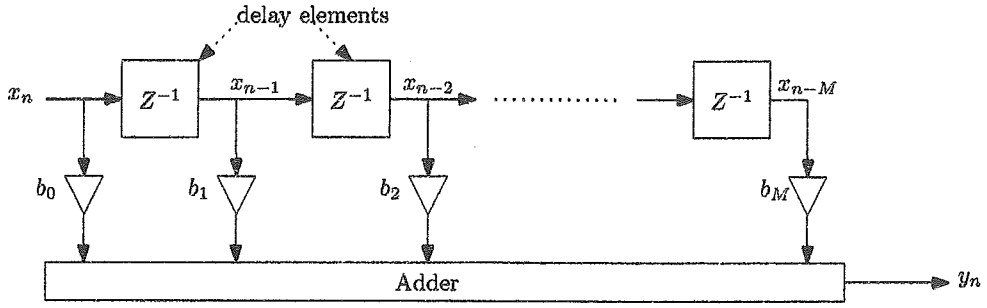


Figure 13.1.2: An example implementation of an FIR filter.

13.2 Properties

13.2.1 Impulse response

What is the output, y_n, from any FIR filter to the following "impulse" input?

$$x_n = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{elsewhere} \end{cases}$$

Well simply look at Figure 13.1.2, and follow the "1" along the delay line computing the output as you go, you should be able to create the following table:

n	Delay line contents $\{x_n, x_{n-1}, x_{n-2} \cdots, x_{n-M}\}$	$y_n = \sum_{k=0}^{M} b_k x_{n-k}$	Comments
$n < 0$	$\{0,0,0,0 \cdots, 0,0\}$	0	output =0
0	$\{\mathbf{1},0,0,0 \cdots, 0,0\}$	b_0	response equals the coefficients $\{b_k\}$
1	$\{0,\mathbf{1},0,0 \cdots, 0,0\}$	b_1	
2	$\{0,0,\mathbf{1},0 \cdots, 0,0\}$	b_2	
3	$\{0,0,0,\mathbf{1} \cdots, 0,0\}$	b_3	
$\vdots$	$\vdots$	$\vdots$	
$M-1$	$\{0,0,0,0 \cdots, \mathbf{1},0\}$	b_{M-1}	
M	$\{0,0,0,0 \cdots, 0,\mathbf{1}\}$	b_M	
$n > M$	$\{0,0,0,0 \cdots, 0,0\}$	0	output =0

Table 13.1: Response of an FIR filter to an impulse

So the output is simply the coefficients $\{b_k\}$.

$$\text{FIR Impulse response} = \{b_k\}$$

Note: This output is Finite in duration, i.e. $b_k = 0$ for $k < 0$ and $k > M$. Hence the name *Finite Impulse Response* (FIR) filter.

13.2.2 Unit step response

What is the output, y_n, from any FIR filter to the following "impulse" input?

$$x_n = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$$

Again simply look at Figure 13.1.2, and follow the "1's" along the delay line computing the output as you go, you should be able to create the following table:

n	Delay line contents $\{x_n, x_{n-1}, x_{n-2} \cdots, x_{n-M}\}$	$y_n = \sum_{k=0}^{M} b_k x_{n-k}$	Comments
$n < 0$	$\{0,0,0,0 \cdots, 0,0\}$	0	output =0
0	$\{1,0,0,0 \cdots, 0,0\}$	b_0	transient
1	$\{1,1,0,0 \cdots, 0,0\}$	$b_0 + b_1$	
3	$\{1,1,1,0 \cdots, 0,0\}$	$b_0 + b_1 + b_3$	
$\vdots$	$\vdots$	$\vdots$	
$M-1$	$\{1,1,1,1 \cdots, 1,0\}$	$b_0 + b_1 + \cdots + b_{M-1}$	
$n \geq M$	$\{1,1,1,1 \cdots, 1,1\}$	$\sum_{k=0}^{M} b_k$	steady state

Table 13.2: Response of an FIR filter to a unit step.

So the output is zero prior to $n = 0$ and then, during a (length M) transient time, becomes the set of partial sums of the coefficients, until finally at time $n = M$ the steady state output equal to the sum of all the coefficients is achieved - it will stay like this forever, i.e. the system is then in a steady state condition.

This steady state condition for the unit step input is also the DC response of the system, i.e. if you put $1's$ in forever you'd always get an output equal to the sum of the coefficients.

$$\text{FIR DC response} = \text{Unit step steady state condition} = \sum_{k=0}^{M} b_k$$

13.2.3 Transfer function

In Section 12.3 that the transfer function of a generic filter is:

$$\tilde{H}(z) = \frac{\tilde{B}(z)}{\tilde{A}(z)}$$

But in the case of an FIR filter, the $\{a_k\}$ coefficients are all zero except for a_0 which is always defined to be $= 1$, and thus $\tilde{A}(z) = 1$, and so:

$$\tilde{H}_{FIR}(z) = \tilde{B}(z) \tag{13.2.1}$$

13.2.4 Poles and zeros

We can factorize the FIR transfer function as follows:

$$\tilde{H}(z) = b_0 z^{-M} \prod_{m=1}^{M} (z - z_m)$$

(see Equation 12.3.2 with $L = 0$).

i.e. there are M poles at the origin, and M zeros.

13.3 Example two tap filter

Consider a simple example where:

$$\begin{aligned} M &= 1 \\ b_0 &= 1 \\ b_1 &= 1 \end{aligned}$$

As illustrated in Figure 13.3.1.

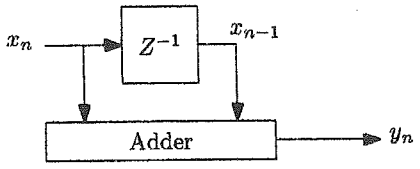


Figure 13.3.1: A very simple FIR filter.

The equation for the output is:

$$y_n = x_n + x_{n-1}$$

13.3.1 Impulse response

We know the impulse response of any FIR filter is just its coefficients, therefore the output is:

$$y_n = \begin{cases} 1 & n = 0 \text{ or } 1 \\ 0 & \text{elsewhere} \end{cases}$$

This is shown in Figure 13.3.2:

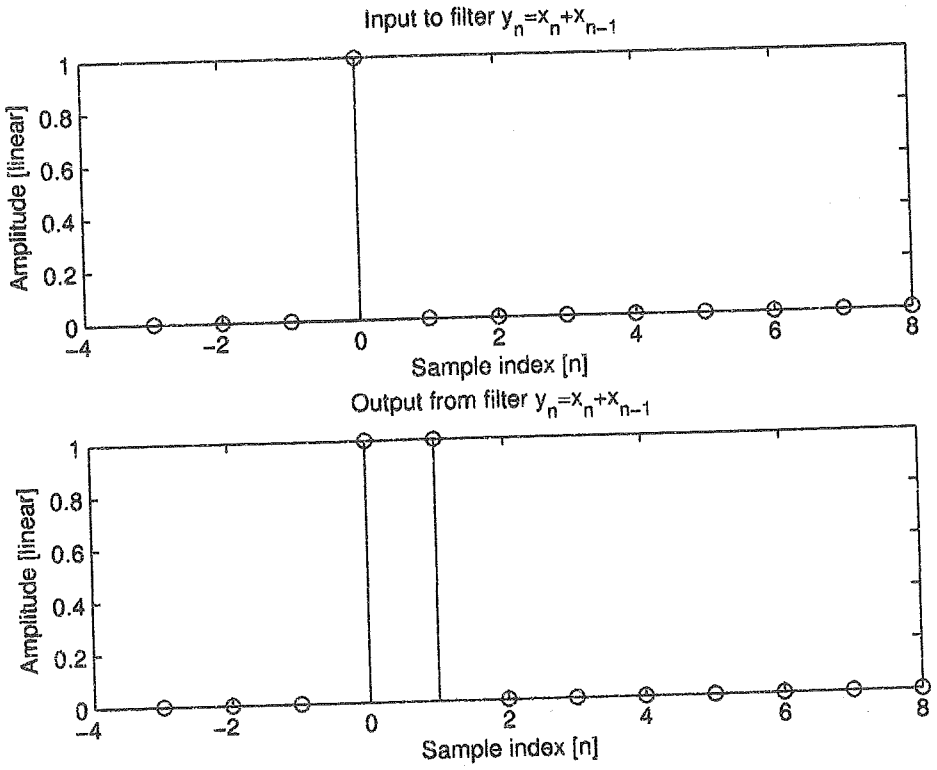


Figure 13.3.2: The impulse response from $y_n = x_n + x_{n-1}$

13.3.2 Unit step

Based on table 13.2 we have:

n	Delay line contents $\{x_n, x_{n-1}\}$	$y_n = \sum_{k=0}^{M} b_k x_{n-k}$	Comments
$n < 0$	$\{0, 0\}$	0	output =0
0	$\{1, 0\}$	1	transient
$n \geq 1$	$\{1, 1\}$	2	steady state

Table 13.3: The unit step response of the FIR filter, $y_n = x_n + x_{n-1}$

So after a, 1 sample, transient time the output settles to a value of 2.

This is shown in Figure 13.3.3:

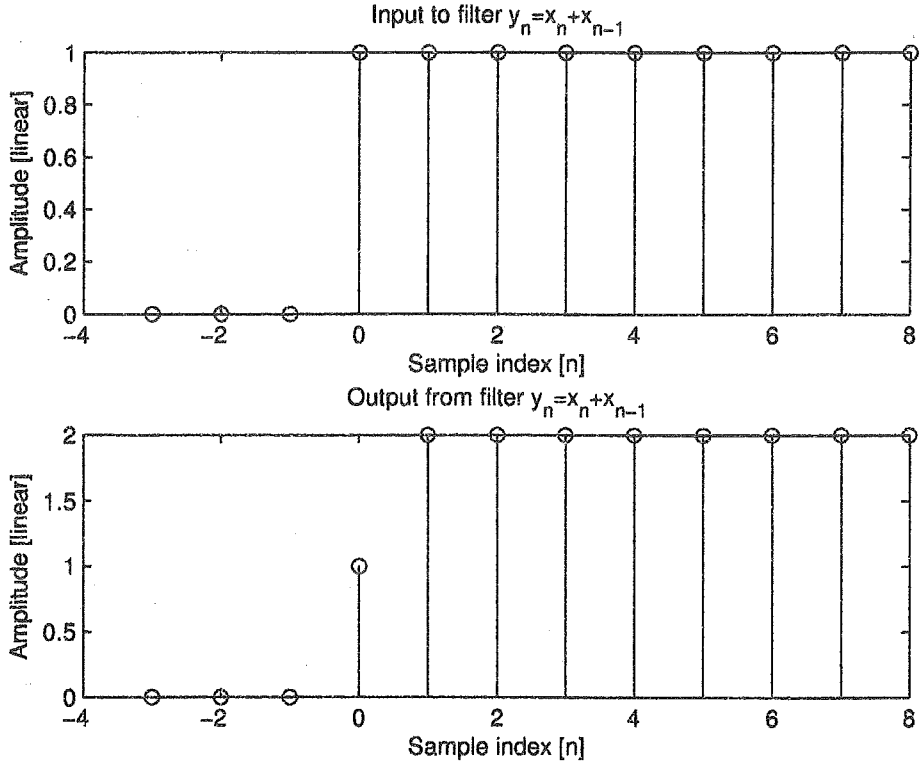


Figure 13.3.3: The unit step response from $y_n = x_n + x_{n-1}$

13.3.3 Response to a sine wave @ $\frac{1}{4}f_s$

|We seek the output of the filter with input:

$$x_n = \begin{cases} \sin\left(\frac{1}{4}\omega_s nT\right) & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

$$= \begin{cases} \sin\left(\frac{\pi}{2}n\right) & n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

But $\sin\left(\frac{\pi}{2}n\right)$ is zero for even n, and alternates between ± 1 on odd n.

Based on table 13.2 we have:

n	x_n	Delay line contents $\{x_n, x_{n-1}\}$	$y_n = \sum_{k=0}^{M} b_k x_{n-k}$	Comments
$n<0$	0	$\{0,0\}$	0	output =0
0	0	$\{0,0\}$	0	
1	1	$\{1,0\}$	1	Alternating between $\{1,1\}$ and $\{-1,-1\}$
2	0	$\{0,1\}$	1	
3	-1	$\{-1,0\}$	-1	
4	0	$\{0,-1\}$	-1	
5	1	$\{1,0\}$	1	
6	0	$\{0,1\}$	1	
7	-1	$\{-1,0\}$	-1	
⋮	⋮	⋮	⋮	

Table 13.4: The response of the FIR filter, $y_n = x_n + x_{n-1}$ to sine wave at $\frac{1}{4}f_s$

So the output in steady state (i.e. when $n \geq 1$) alternates between $\{1,1\}$ and $\{-1,-1\}$ which, as can can be seen from Figure 13.3.4, is a sine wave with period 4 sample, i.e. frequency $\frac{1}{4}f_s$ as expected.

The amplitude and phase can be computed noting that its amplitude is 1 when $n = 2$ and is $-1$when $n = 3$, i.e.:

$$\begin{aligned} A\sin\left(\frac{\pi}{2}2+\phi\right) &= 1 \text{ and} \\ A\sin\left(\frac{\pi}{2}3+\phi\right) &= -1 \end{aligned}$$

Solve for ϕ by equating the two left-hand-sides:

$$\sin(\pi+\phi) = -\sin\left(3\frac{\pi}{2}+\phi\right)$$

Subtract π from both angles

$$\sin(\phi) = -\sin\left(\frac{\pi}{2}+\phi\right)$$

Noting that *sine* is an odd function, the smallest angle where this can be true is $\phi = -\frac{\pi}{4}$.

Now we can compute A, i.e.

$$\begin{aligned} A\sin\left(\frac{\pi}{2}2+\phi\right) &= 1 \\ \Rightarrow A\sin\left(\frac{\pi}{2}2-\frac{\pi}{4}\right) &= 1 \\ \Rightarrow A &= \sqrt{2} \end{aligned}$$

So the final output is: $\sqrt{2}\sin\left(\frac{\pi}{2}n-\frac{\pi}{4}\right)$.

This is shown in Figure 13.3.4:

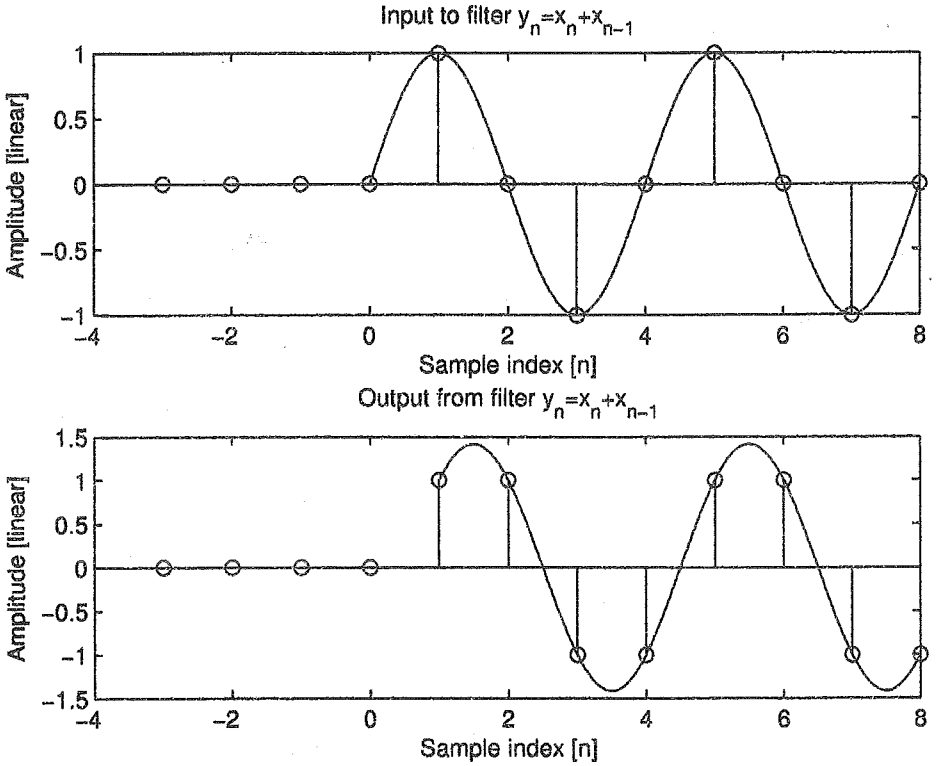


Figure 13.3.4: The response of $y_n = x_n + x_{n-1}$ to a sine wave at $\frac{1}{4}f_s$

13.3.4 Response to a cosine wave @ $\frac{1}{2}f_s$

We seek the output of the filter with input:

$$
\begin{aligned}
x_n &= \begin{cases} \cos\left(\frac{1}{2}\omega_s nT\right) & n \geq 0 \\ 0 & \text{elsewhere} \end{cases} \\
&= \begin{cases} \cos(n\pi) & n \geq 0 \\ 0 & \text{elsewhere} \end{cases} \\
&= \begin{cases} 1 & n \geq 0 \text{ and } n \text{ even} \\ -1 & n \geq 0 \text{ and } n \text{ odd} \\ 0 & n < 0 \end{cases}
\end{aligned}
$$

Based on table 13.2 we have:

n	x_n	Delay line contents $\{x_n, x_{n-1}\}$	$y_n = \sum_{k=0}^{M} b_k x_{n-k}$	Comments
$n < 0$	0	$\{0, 0\}$	0	output =0
0	1	$\{1, 0\}$	1	transient
1	0	$\{-1, 1\}$	0	Steady state
2	-1	$\{1, -1\}$	0	
3	0	$\{-1, 1\}$	0	
4	1	$\{1, -1\}$	0	
5	0	$\{-1, 1\}$	0	
6	-1	$\{1, -1\}$	0	
⋮	⋮	⋮	⋮	

Table 13.5: The response of the FIR filter, $y_n = x_n + x_{n-1}$ to cosine wave at $\frac{1}{2}f_s$

So the output in steady state (i.e. when $n \geq 1$) is always zero!.

This is shown in Figure 13.3.5:

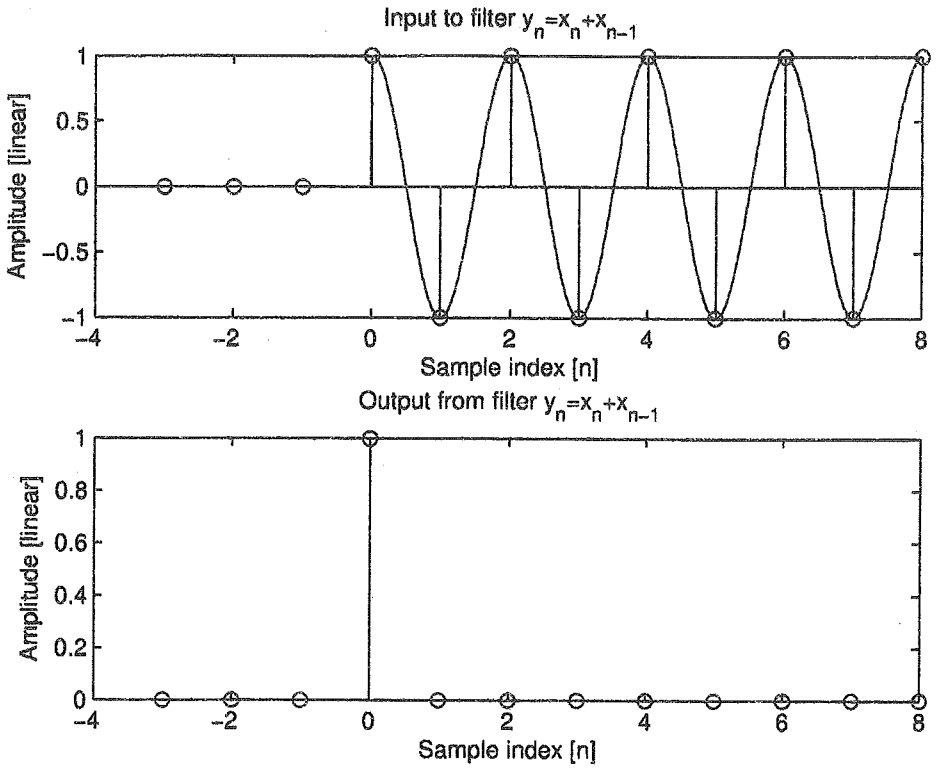


Figure 13.3.5: The response of $y_n = x_n + x_{n-1}$ to a cosine at $\frac{1}{2}f_s$. Notice the transient at $n = 0$.

13.3.5 Transfer function

For an FIR filter the transfer function is just the Fourier transform the coefficients which in this case are $\{1, 1\}$, so we have

$$\tilde{H}(z) = 1 + z^{-1}$$

The frequency response is obtained by evaluating the transfer function at $z = e^{j\omega T}$, i.e.:

$$\bar{H}(j\omega) = \tilde{H}\left(e^{j\omega T}\right) = 1 + e^{-j\omega T}$$

Exercise:

Show that:

$$\left|\bar{H}(j\omega)\right| = \sqrt{2(1+\cos\omega T)} \quad \text{and} \quad \angle\bar{H}(j\omega) = -\frac{1}{2}\omega T$$

as shown in Figures 13.3.6 and 13.3.6.

Solution:

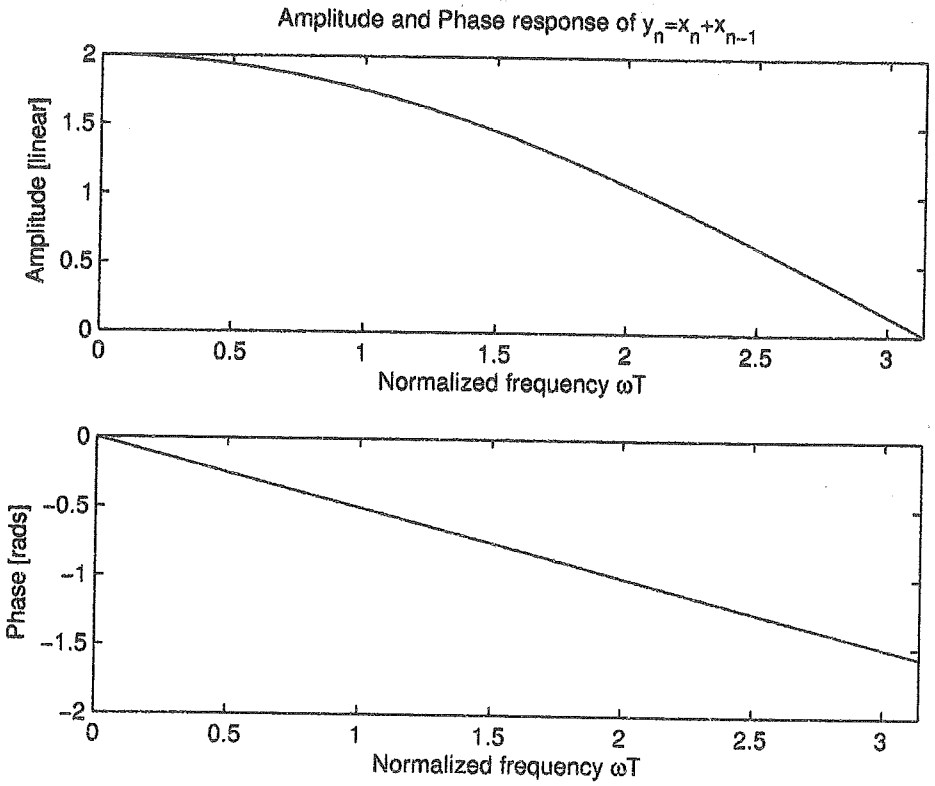


Figure 13.3.6: The response of the FIR filter, $y_n = x_n + x_{n-1}$.

The phase response is linear (with respect to ω) - this is a very important result. A linear phase response implies that each frequency component is delayed by a fixed amount (given by the slope of the phase response).
This fixed delay property is very important in many audio and video signal processing.

13.3.6 The easy way

In the preceding sections we computed the outputs for various types of input. If we are only interested in the steady state behavior, then we could simply use the transfer function $\tilde{H}(z)$ as computed in Section 13.3.5 instead. This is shown in Table 13.6, where it can be seen that this agrees with the previously derived results.

Frequency	ωT	$z = e^{j\omega T}$	$\left\|\tilde{H}\left(e^{j\omega T}\right)\right\| = \sqrt{2\left(1+\cos\omega T\right)}$	$\angle\tilde{H}\left(e^{j\omega T}\right) = -\frac{1}{2}\omega T$
DC response	0	$1+j0$	2	0
$\frac{1}{4}f_s$	$2\pi\frac{1}{4} = \frac{\pi}{2}$	$0+j$	$\sqrt{2}$	$-\frac{1}{4}\pi$
$\frac{1}{2}f_s$	$2\pi\frac{1}{2} = \pi$	$-1+j0$	0	$\frac{1}{2}\pi$

Table 13.6: Computation of steady state response of $y_n = x_n + x_{n-1}$.

Chapter 14

Linear Phase FIR filters

14.1 Why linear phase?

Imagine an analog filter that passes all frequency components of a particular input signal with the same delay, e.g. by τ_o sec, with unit gain. Then the input and output for several frequencies could be as per the following table:

frequency	input, $\cos(2\pi ft)$	output, delayed by τ sec $= \cos(2\pi f(t-\tau_o))$	phase [rads]
1Hz	$\cos(2\pi t)$	$\cos(2\pi(t-\tau_o)) = \cos(2\pi t - 2\pi\tau_o)$	$-2\pi\tau_o$
2Hz	$\cos(4\pi t)$	$\cos(4\pi t - 4\pi\tau_o)$	$-4\pi\tau_o$
3Hz	$\cos(6\pi t)$	$\cos(6\pi t - 6\pi\tau_o)$	$-6\pi\tau_o$
$\vdots$	$\vdots$	$\vdots$	
f Hz	$\cos(2\pi ft)$	$\cos(2\pi ft - 2\pi f\tau_o)$	$-2\pi f\tau_o$
ω rads/sec	$\cos(\omega t)$	$\cos(\omega t - \omega\tau_o)$	$-\omega\tau_o$

i.e. if the the phase response of a filter is a linear function of ω, i.e. if $\angle H(j\omega) = -\omega\tau_o$, then the *group delay* of said filter is τ_o seconds as all the frequency components are delay as a group by τ_o seconds.

A signal having all frequency components within the pass-band of an ideal filter will experience neither amplitude nor phase distortion save for a fixed delay or τ_o seconds. This is why we almost always try to design linear phase filters.

As an example, Figure 14.1.1 shows the ideal response of a low-pass filter.

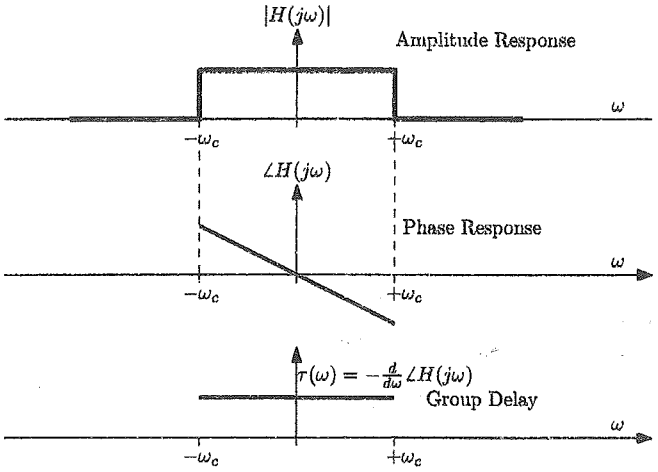


Figure 14.1.1: Ideal response of a low-pass filter

14.1.1 Group delay

More formally we define the group delay of a filter as minus the derivative of the phase response:

$$\tau(\omega) = -\frac{d}{d\omega}\angle H(j\omega)$$

Clearly, in the example given above where the filter introduces a fixed time delay, .e. if $\angle H(j\omega) = -\omega\tau_o$, we have $\tau(\omega) = \tau_o$ which makes sense and is also shown in Figure 14.1.1. Aside: Technically, $\angle H(j\omega) = -\omega\tau_o + K$, is also regarded as linear phase, but a non-zero constant K violates the nice delay properties we often want, but none-the-less these types of filters do find same applications, e.g. in differentiators and Hilbert transformers (beyond the scope of this course).

All the same arguments can be made for discrete time filters, so now we will consider what conditions are needed to ensure a linear phase response for Real valued filters.

14.2 Sufficient conditions for real filters

We wish to find the conditions under which a real valued FIR filter has an exactly linear phase response.
From Section 13.2.3 we know that the transfer function, $\tilde{H}(z)$, of an FIR filter having coefficients $\{b_k\}$ is:

$$\tilde{H}(z) = \tilde{B}(z)$$

Where $\tilde{B}(z)$ is the z-transform of the coefficients. Expanding this we have:

$$\tilde{H}(z) = \sum_{k=0}^{M} b_k z^{-k}$$

where there are a total of $M+1$ coefficients.
Again we stress that (for now) we'll assume the $\{b_k\}$ are all real valued.

14.2.1 Preliminaries

I happen to know that the group delay of the result is going to be $\frac{M}{2}$, so as a starting point we rewrite transfer function, $\tilde{H}(z)$, by factoring out that delay, which we know from the shift theorem involves a multiplication by a power of z, i.e.:

$$\tilde{H}(z) = z^{-\frac{M}{2}} \left(\sum_{k=0}^{M} b_k z^{-\left(k-\frac{M}{2}\right)} \right)$$

We are going to be interested in the angle of frequency response $\tilde{H}(j\omega)$ which is given by evaluating the $\tilde{H}(z)$ along the unit circle, i.e. when $z = e^{j\omega T}$, thus:

$$\begin{aligned}\bar{H}(j\omega) = \tilde{H}\left(e^{j\omega T}\right) &= e^{-j\frac{M}{2}\omega T}\left(\sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}\right) \\ \Rightarrow \angle\bar{H}(j\omega) &= -\frac{M}{2}\omega T + \angle\sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}\end{aligned}$$

This will be linear phase if the angle of the summation above is a linear function of ω.
i.e., we need to select the $\{b_k\}$ to ensure that $\angle\sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}$ is a linear function of ω.

14.2.2 Symmetry

We claim that if the real valued coefficients have symmetry, $b_k = b_{M-k}$ or anti-symmetry $b_k = -b_{M-k}$, the resulting filter will have linear phase. More generally we can write $b_k = \lambda b_{M-k}$ for $\lambda = +1$ or $\lambda = -1$.
To prove this we need to consider the two cases of an odd and an even number of coefficients, and for each examine the phase response for $b_k = \lambda b_{M-k}$ for $\lambda = +1$ and $\lambda = -1$.

14.2.2.1 An even number of coefficients

Suppose M is odd, i.e. we've an even number of coefficients, then we can break the summation in, $\angle\sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}$, into two equal halves:

$$\begin{aligned}&= \angle\left(\sum_{k=0}^{\frac{M-1}{2}} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T} + \sum_{k=\frac{M+1}{2}}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}\right) \\ &= \angle\left(\sum_{k=0}^{\frac{M-1}{2}} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T} + \sum_{l=\frac{M-1}{2}}^{0} b_{M-l} e^{j\left(l-\frac{M}{2}\right)\omega T}\right) \\ &= \angle\left(\sum_{k=0}^{\frac{M-1}{2}} \left(b_k e^{-j\left(k-\frac{M}{2}\right)\omega T} + b_{M-k} e^{j\left(k-\frac{M}{2}\right)\omega T}\right)\right) \qquad (14.2.1)\end{aligned}$$

Where in the second line above we substituted $l = M - k$ to make the range of the two summations the same. In the third line we merge the two summations.
Now we apply our assumption that $b_k = \lambda b_{M-k}$.

$$= \angle\left(\sum_{k=0}^{\frac{M-1}{2}} b_{M-k}\left(\lambda e^{-j\left(k-\frac{M}{2}\right)\omega T} + e^{j\left(k-\frac{M}{2}\right)\omega T}\right)\right)$$

Notice that the two exponential are complex conjugates of each other, and, that for any complex number ν we have $\nu + \nu^* = 2\Re(\nu)$ and $\nu - \nu^* = 2j\Im(\nu)$, therefore:

- If $\lambda = 1$ (i.e. symmetric case), the bracketed term above real valued.
- If $\lambda = -1$ (i.e. anti-symmetric case), the bracketed term above complex valued.

Therefore, the above equation reduces to:

$$= \begin{cases} \angle\left(\sum_{k=0}^{\frac{M-1}{2}} b_{M-k} 2\Re\left(e^{j\left(k-\frac{M}{2}\right)\omega T}\right)\right) & \text{for symmetric case} \\ \angle\left(j\sum_{k=0}^{\frac{M-1}{2}} b_{M-k} 2\Im\left(e^{j\left(k-\frac{M}{2}\right)\omega T}\right)\right) & \text{for anti-symmetric case} \end{cases}$$

For the symmetric case everything is real valued[1] $\Rightarrow$ phase $= 0$ or $= \pi$.
For the anti-symmetric case everything is imaginary $\Rightarrow$ phase $= \pm\frac{\pi}{2}$.
So finally we have:

$$\Rightarrow \angle\tilde{H}(j\omega) = -\frac{M}{2}\omega T + \begin{cases} 0 \text{ or } \pi & \text{for symmetric case} \\ \pm\frac{\pi}{2} & \text{for anti-symmetric case} \end{cases}$$

Which are all linear functions of ω.
Therefore either symmetric or anti symmetric coefficients, $b_k = \pm b_{M-k}$, is a sufficient condition for an FIR with an even number of coefficients to have a linear phase response. In either case the group delay is $\frac{M}{2}T$ seconds (this is half way between the two middle coefficients).

$$\tau = -\frac{d}{d\omega}\angle\tilde{H}\left(e^{j\omega T}\right) = \frac{M}{2}T$$

As an example, Figure 14.2.1 shows this condition for a length 10 FIR filter.

[1] A phase shift of $-\pi$ is also possible, but that's the same as a phase shift of π so its not listed as a separate option.

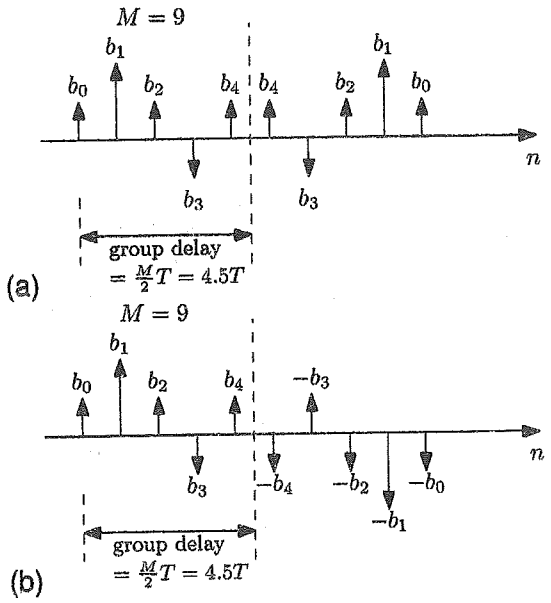


Figure 14.2.1: Example of an FIR filter with 10 real valued coefficients with (a) symmetry and (b) anti-symmetry. These will have a linear phase response with a group delay of 4.5 sampling intervals.

Note: for both odd and even cases there are two equations for the phase response, each differing by exactly π radians. This corresponds to a signal inversion, i.e. cases where the FIR filter has a positive and negative gain; remember the magnitude of the response is (by definition) positive, so signal inversion must be embodied in the phase response.

14.2.2.2 An odd number of coefficients

The rule $b_k = \pm b_{M-k}$ also work for an odd number of coefficients as is easily proven in exactly the same way as above except that we must consider the middle coefficient $b_{\frac{M}{2}}$:

- If $b_k = b_{M-k}$, then the middle term $b_{\frac{M}{2}}$ can have any value, but
- If $b_k = -b_{M-k}$, then the middle term $b_{\frac{M}{2}}$ must be $= 0$ as this is the only way $b_{\frac{M}{2}} = -b_{\frac{M}{2}}$ assuming real values.

Otherwise the proof is largely the same yielding the exact same result, i.e.

$$\angle \bar{H}(j\omega) = -\frac{M}{2}\omega T + \begin{cases} 0 \text{ or } \pi & \text{for symmetric case} \\ \pm\frac{\pi}{2} & \text{for anti-symmetric case} \end{cases}$$

and a group delay of $\frac{M}{2}T$ (this the position of the middle coefficient).

As an example, Figure 14.2.2 shows these condition for a length 11 FIR filter.

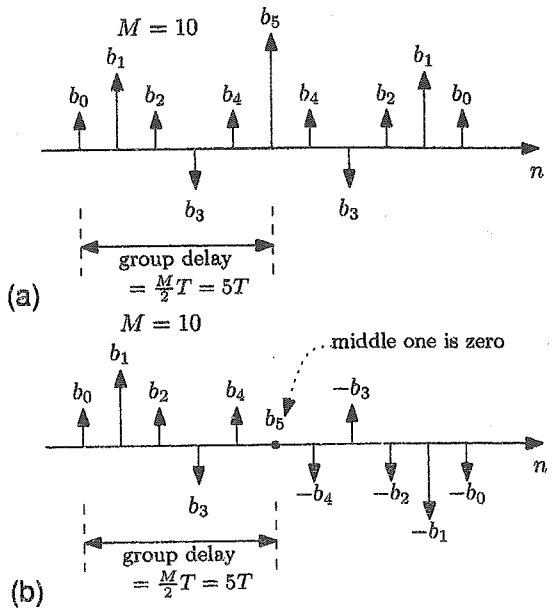


Figure 14.2.2: Example of an FIR filter with 11 real valued coefficients with (a) symmetry and (b) anti-symmetry. These will have a linear phase response with a group delay of 5 sampling intervals.

14.2.3 Conclusion

Real valued linear phase FIR filters can be constructed by ensuring that the coefficients have either:

1. Symmetry: $b_n = b_{M-n}$, or
2. Anti symmetry: $b_n = -b_{M-n}$

If there is an odd number of coefficients there is a middle coefficient, $b_{\frac{M}{2}}$ must be $= 0$.
In all cases the group delay is:

$$\tau = \frac{M}{2}T$$

We now claim, without proof, that for real valued FIR filters these represent a *necessary* condition for linear phase, i.e. they are not just sufficient.

For the interested reader, there is another formulation given in the Appendix B for complex valued coefficients where the same results are derived but the symmetries are conjugated.

14.3 z-domain view

Note that we've already established in Section 12.3.4 that any filter with real coefficients have their poles and zeros either on the real axis in the z-plane, or in conjugate pairs. FIR filters only have zeros but do still, of course, conform to this rule:

Real valued FIR filters have either real zeros and / or they occur in conjugate pairs. They also have M poles at the orgin.

Linear phase filters have an additional constraint on the position of the zeros, as we will now see.

Exercise:

In addition to the above criteria the zeros of a real valued linear phase FIR filters occur in invert pairs, i.e. for a linear phase filter, if ζ is a zero, then so too is ζ^{-1}.

Solution:

We now restate as follows:

> All real valued linear phase FIR filters have either real valued zeros, or they occur in conjugate pairs Additionally the inverse of any zero is also a zero.

Based on this theorem we can conclude that the zeros of a real linear phase FIR filter can only occur in the following ways:

- On their own at $z = -1$ or at $z = +1$ (as ± 1 are the only real values that are their own inverse).
- In conjugate pairs on the unit circle.
- In groups of 4, at for example at $r.e^{j\alpha}$ and its inverse $\frac{1}{r}.e^{-j\alpha}$ and both their conjugates $r.e^{-j\alpha}$ and $\frac{1}{r}.e^{j\alpha}$.

These are shown in Figure 14.3.1.

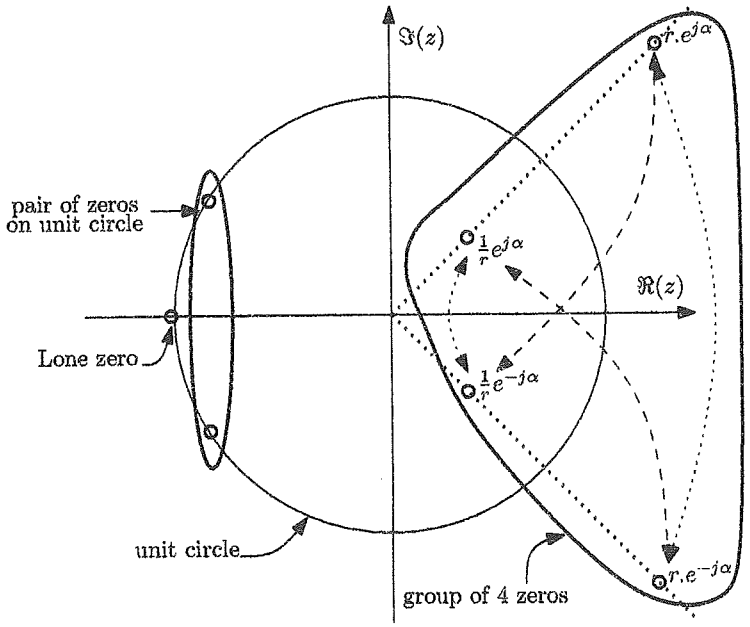


Figure 14.3.1: Example showing position of zeros for a real valued FIR filter with linear phase. All three possible types of zero are illustrated.

Chapter 15

FIR design methods

15.1 Introduction

How to calculate the coefficients $\{b_k\}$ so that the frequency response (amplitude and phase) is a good approximation of some desired response?

15.2 Fourier series method

Reminder:

A periodic function $\bar{f}(t)$ can be represented by a Fourier series:

$$\bar{f}(t) = \sum_{k=-\infty}^{+\infty} c_k e^{jk\frac{2\pi}{T}t}$$

where T is the period of $\bar{f}(t)$.
The Fourier coefficients c_k can be computed as:

$$c_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} \bar{f}(t)\, e^{-jk\frac{2\pi}{T}t} dt$$

Application to FIR design:

The frequency response, $\bar{H}(j\omega)$, of any digital filter is periodic and can also be represented by a Fourier series (note that normally we talk of a time domain signal being period and representing it as a Fourier series, whereas here we are swapping the roles of time and frequency). This is illustrated in Figure 15.2.1.

Note that the period (normally denoted T) is actually $\frac{2\pi}{T}$.

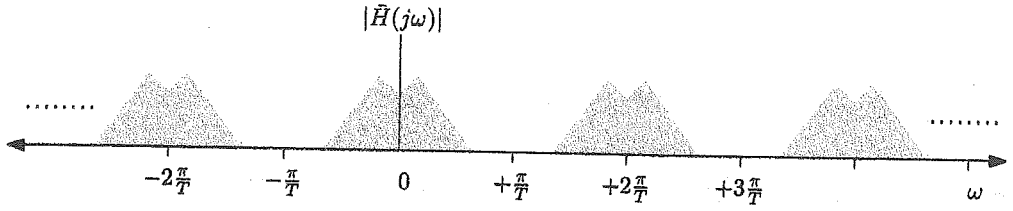


Figure 15.2.1: The frequency response of a digital filter is periodic. The "period" is $\frac{2\pi}{T}$.

So apply the Fourier series to $\bar{H}(j\omega)$ (with the roles of time and frequency reversed[1]) we get:

$$\begin{aligned}\bar{H}(j\omega) &= \sum_{k=-\infty}^{+\infty} c_k e^{jk\frac{2\pi}{\left(\frac{2\pi}{T}\right)}\omega} \\ &= \sum_{k=-\infty}^{+\infty} c_k e^{jk\omega T}\end{aligned} \tag{15.2.1}$$

where

$$\begin{aligned}c_k &= \frac{T}{2\pi}\int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \bar{H}(j\omega)\, e^{-jk\frac{2\pi}{\left(\frac{2\pi}{T}\right)}\omega} d\omega \\ &= \frac{T}{2\pi}\int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \bar{H}(j\omega)\, e^{-jkT\omega} d\omega\end{aligned}$$

But the frequency response of an (infinity) long FIR[2] with coefficients $\{b_k\}$ $-\infty < k < +\infty$, is:

$$\begin{aligned}\bar{H}_{FIR}(j\omega) &= \sum_{k=-\infty}^{+\infty} b_k e^{-jk\omega T} \\ &= \sum_{k=-\infty}^{+\infty} b_{-k} e^{jk\omega T}\end{aligned}$$

Comparing this to Equation 15.2.1 we can see that if:

$$b_k = c_{-k}$$

then the resulting FIR will have the same frequency response as the desired $\bar{H}(j\omega)$. But we have an equation for the Fourier coefficients $\{c_k\}$, thus:

[1] substitute $t \mapsto \omega$ and $T \mapsto \frac{2\pi}{T}$

[2] I'm aware of the contradiction in terms here, but lets just say a very ling FIR filter...

$$b_k = \frac{T}{2\pi}\int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \bar{H}(j\omega)\, e^{jkT\omega} d\omega \quad -\infty < k < +\infty \tag{15.2.2}$$

This is known as the *filter synthesis* formula.

We will now look at some examples, and see some of the limitations are this method.

15.2.1 LPF with linear phase

We require a filter with a response as shown in Figure 15.2.2, and described mathematically by for $|\omega| < \frac{1}{2}\omega_s$ (remember it's periodic outside that range):

$$\bar{H}(j\omega) = \begin{cases} e^{-j\lambda\omega T} & |\omega| < \omega_c \\ 0 & \text{elsewhere} \end{cases}$$

Where λ relates to the group delay and can be any real number ≥ 0.

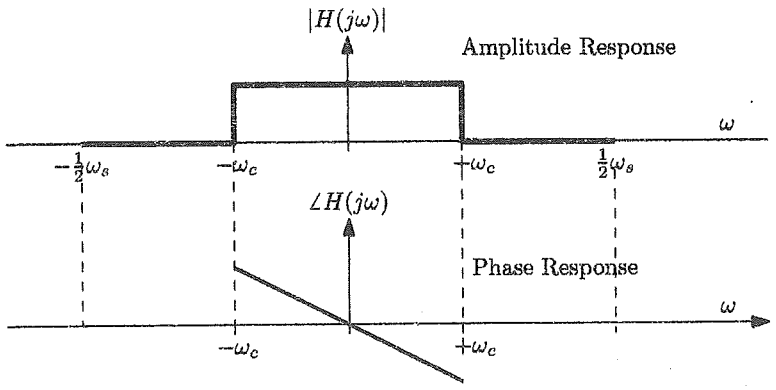


Figure 15.2.2: Ideal Low-Pass-Filter (LPF) with linear phase response and cut-off frequency of ω_c rads/sec.

Exercise:

Applying the filter synthesis Equation 15.2.2 to this $\bar{H}(j\omega)$ to get the result:

$$b_k = 2\left(\frac{f_c}{f_s}\right)\text{sinc}\left(2(k-\lambda)\frac{f_c}{f_s}\right)$$

where $\text{sinc}(x) \triangleq \frac{\sin \pi x}{\pi x}$.

Solution:

Example: Let $\lambda = 0$ (corresponds to a zero-phase filter), and $\frac{f_c}{f_s} = \frac{1}{8}$, we have:

$$b_k = \frac{1}{4}\text{sinc}\left(\frac{1}{4}k\right) \quad -\infty < k < +\infty$$

The middle 21 values (about $k = 0$) are shown in Figure 15.2.3.

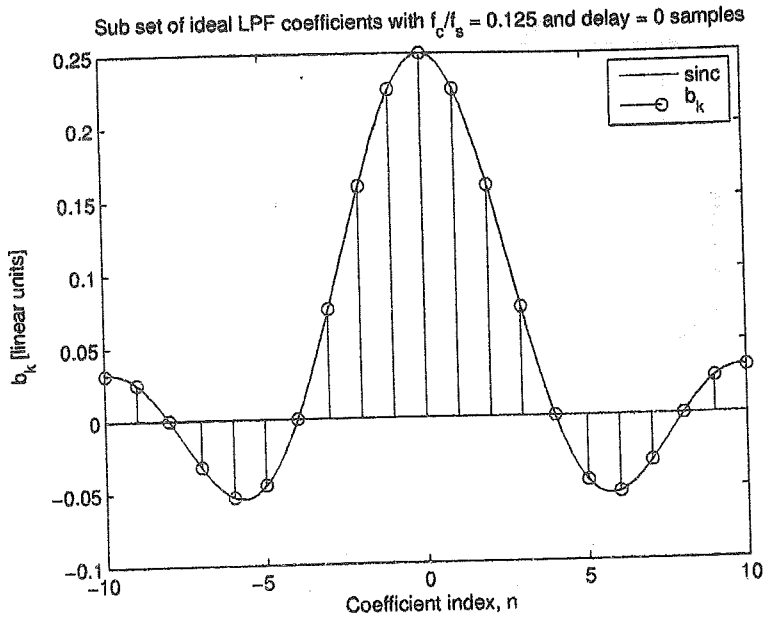


Figure 15.2.3: The 21 coefficients of an Ideal LPF with zero delay.

15.2.2 Finite length

One problem with the filter synthesis formula, Equation 15.2.2 is that it produces an infinite number of coefficients $\{b_k\}$ which clearly can't be implemented in a finite machine!

We could simply take a sub-set of them, e.g. the middle 21 coefficients as was shown in Figure 15.2.3. By doing this we would expect that the response of the filter will no longer match the exact specification and will instead be an approximation - this is illustrated in Figure 15.2.4.

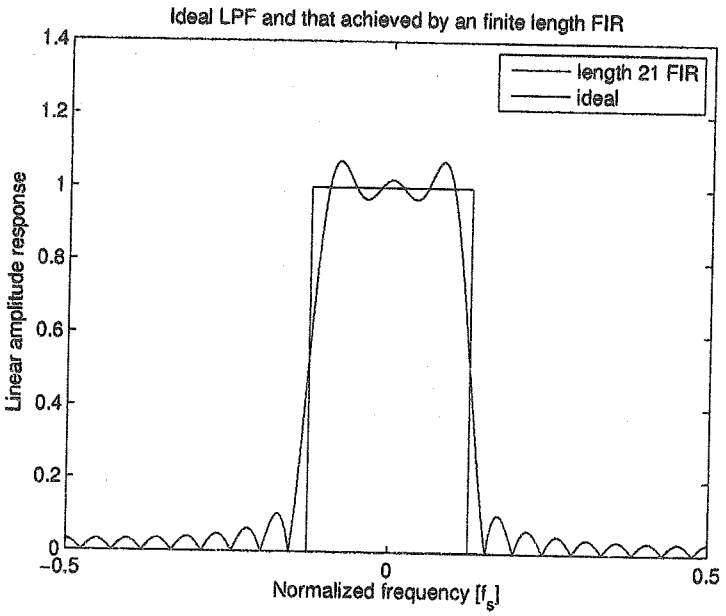


Figure 15.2.4: The response a finite length LPF designed using the Fourier series method and truncated using a rectangular window. The difference between the idea can be modeled by the convolution with a sinc function (the Fourier transform of a rectangular window).

The process of taking just a sub-set of the coefficients can be modeled mathematically as the multiplication of the (sampled time domain) filter coefficients by a rectangular window (of length 21 in this case).

Using the usual Fourier pairs, we can say that the impact on the frequency response is that of a convolution with the Fourier transform of the rectangular window, i.e. another sinc function.

This convolution has the effect of smearing the ideal response as shown in Figure 15.2.3.

Again as with many Fourier transforms, the wider the time domain rectangular window (i.e. the more samples we take) the more narrow the frequency domain sinc shape becomes - i.e. the convolution has less impact.

The subject of windowing requires further exploration and is considered in Section 15.3.

15.2.3 Causality

Another issue with the set of coefficients presented in Figure 15.2.3 is that they are non-causal, i.e. they don't start at $k = 0$, and so any implementation would require a time-machine!

However this is easily fixed by simply introducing a delay.

Recall the discussion on linear phase filters in Section 14.

The filter designed here has a linear phase response with group delay of

$$\begin{aligned} \tau &= -\frac{d}{d\omega}\angle \bar{H}\left(j\omega\right) \\ &= \lambda T \end{aligned}$$

This is why setting $\lambda = 0$ as in Figure 15.2.3 results in a set of coefficients that are centered on $k = 0$. Had we, for example, selected $\lambda = 10$ we would get a set of coefficients as per Figure 15.2.5 - which is causal.

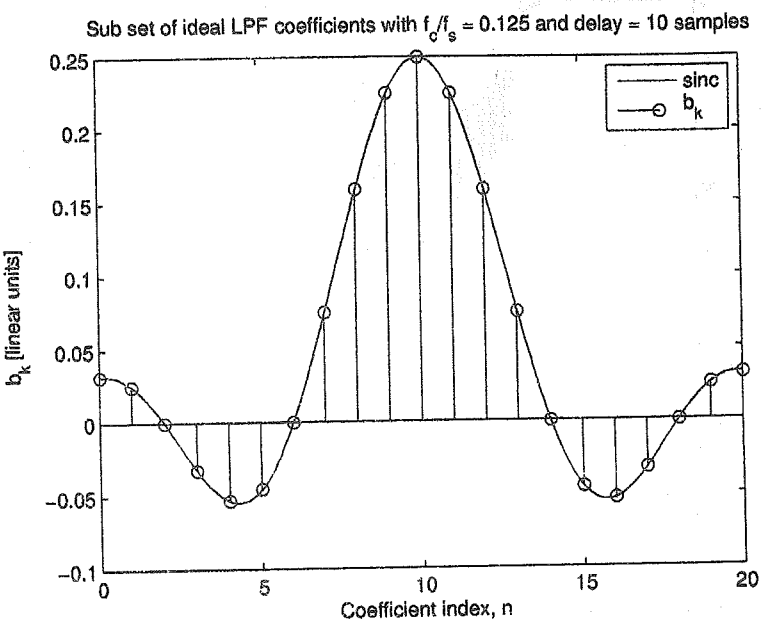


Figure 15.2.5: The same filter as Figure 15.2.3 but with a delay of $\lambda = 10$.

15.3 Windowing

To demonstrate the reason for windowing we'll begin by considering the example of Section 15.2.1 a 21 coefficient design was presented - we now show its amplitude response on a log scale in Figure 15.3.1 along with several other lengths (obtained according to the same procedure).

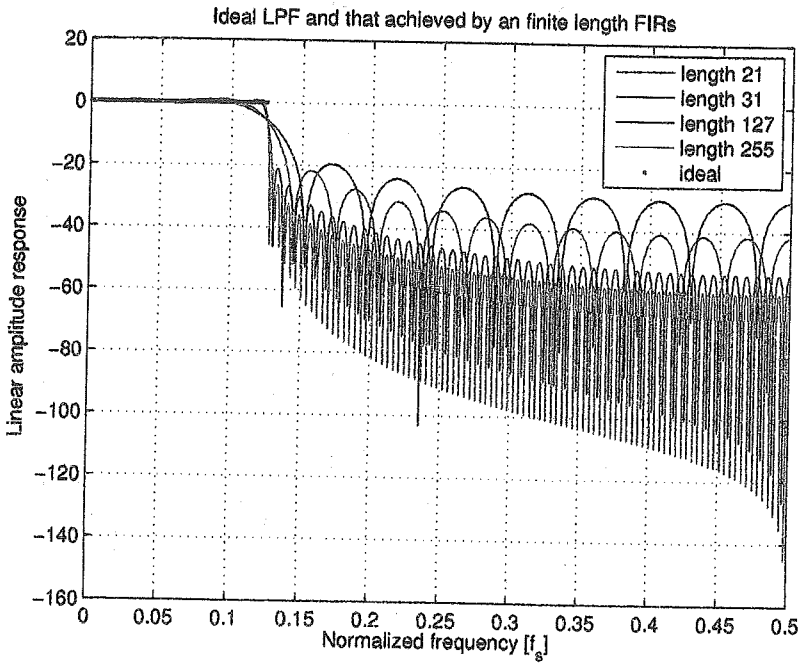


Figure 15.3.1: Various length FIR designed according to the synthesis equation 15.2.2 method.

We can see that that as the length increases the stopband attenuation increases slowly according to the following table:

FIR Length	attenuation at $\frac{1}{2}f_s$ [dB]	lowest attenuation after $0.15f_s$ [dB]
21	-30.5	-19.6
31	-41.7	-21.8
127	-53.9	-30.2
255	-60.4	-36.1

So we get nearly 20dB of stopband attenuation with a length 21 filter, but only an extra 10dB improvement when implement a 127 length filter - this slow increase in attenuation is a major problem and is one of the main reasons rectangular windowing is often not liked.

To see what happens within the passband we need to zoom in as per Figure 15.3.2. Here we see that the peak error within the passband remains large - especially close to the start of the transition band. This is known as the Gibbs effect - only if we take the full (infinite number) set of coefficients will be not have this error.

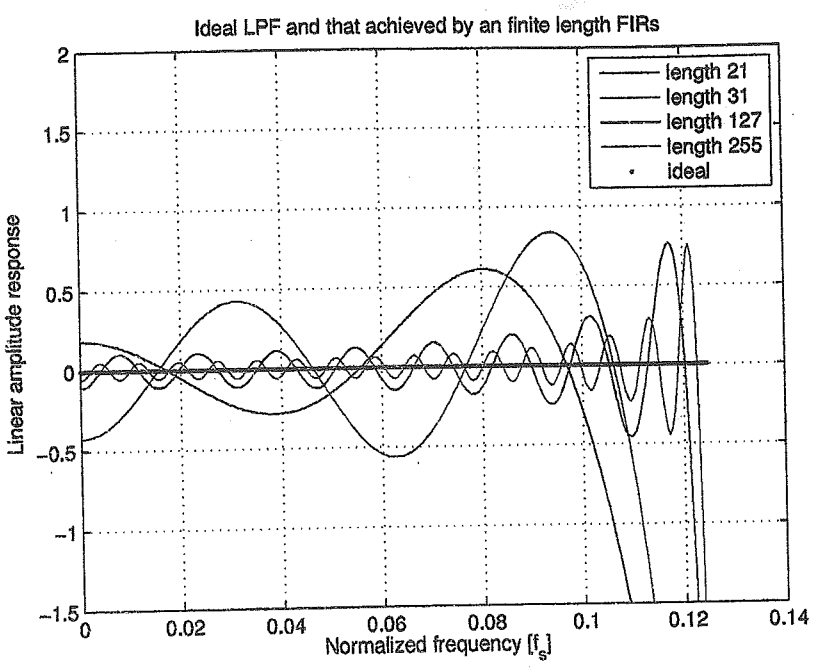


Figure 15.3.2: Zoomed in version of Figure 15.3.1.

15.3.1 Math model

We can model the taking of a finite number of samples from the filter design formula 15.2.2 as a multiplication be a rectangular time domain window. The effect of doing this in the frequency domain is a convolution by its Fourier transform. This is illustrated in Figure 15.3.3.

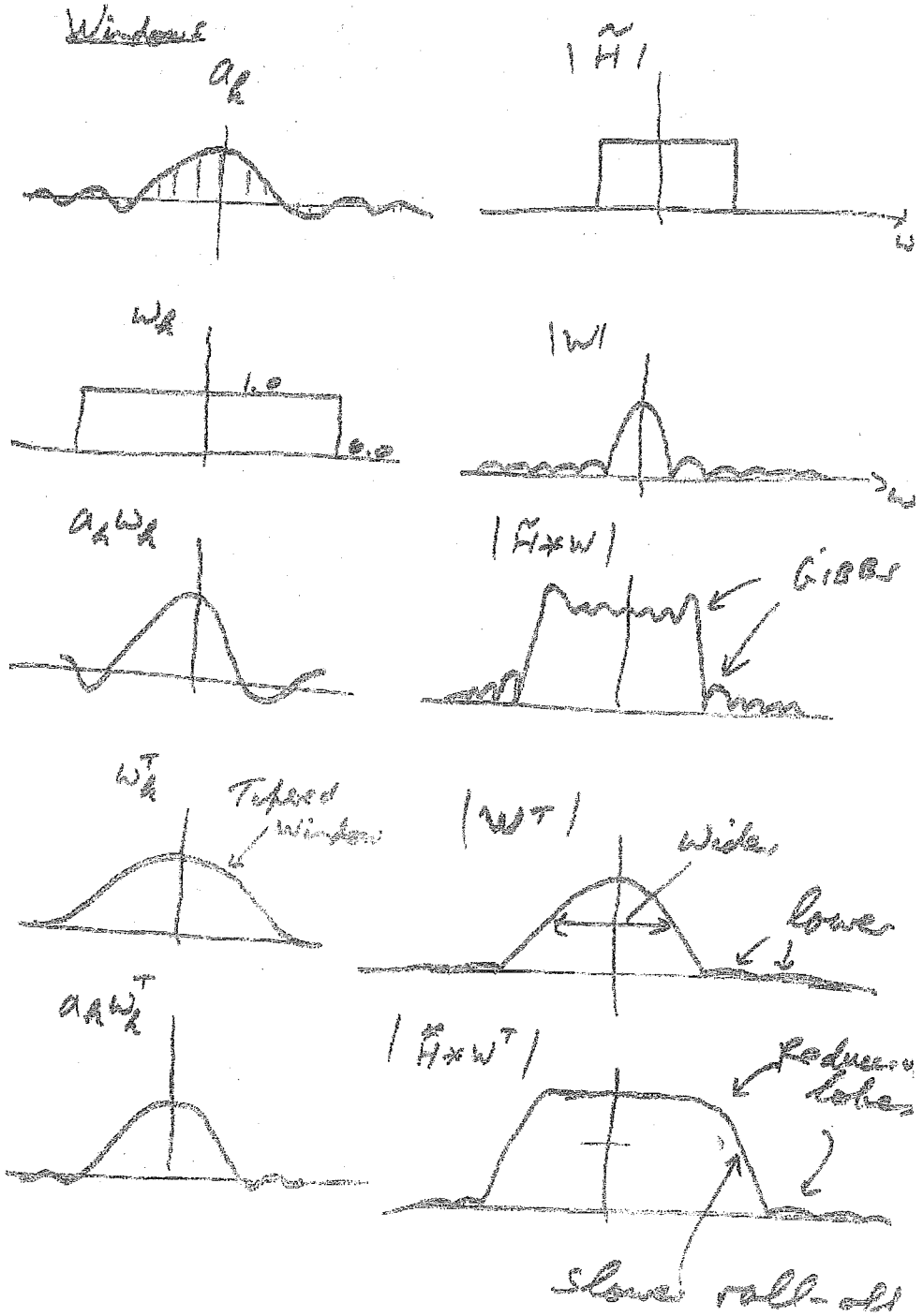


Figure 15.3.3: Several Fourier transform pairs showing the effect of windowing.

Also shown in Figure 15.3.3 is the use of a tapered window instead. This, in the frequency domain, has a wider main lobe, but smaller side lobes. Thus it spreads the resulting amplitude spectrum differently:

- The transition band becomes slower, but

- The side lobes in the stopband can be much lower.

15.3.2 Windowing procedure

When truncating the infinite set $\{b_k\}$ do the following:

- Center the window about the region of maximum energy (the middle coefficient)
- Take an equal amount of samples about this central point - this ensures coefficient symmetry, thus preserving linear phase.
- Multiply by the window function $\{w_k\}$, which is assumed to be zero beyond a finite number of coefficients, i.e. the filter coefficients become: $\{w_k b_k\}$ for $k = 0, \cdots, M$.

Common window functions are:

- Rectangular
- Hamming(*)
- Blackman
- Kaiser(*)

By way of an example we will now look at the popular Hamming and Kaiser windows.

15.3.2.1 Hamming

The Hamming window is essentially a raised cosine shape:

$$w_k = 0.54 - 0.56 \times \cos\left(2\pi \frac{k}{M}\right) \quad \text{for } k = k = 0, \cdots, M$$

Note, it's peak (value = 1.0) occurs in the middle position. An example is shown in Figure 15.3.4

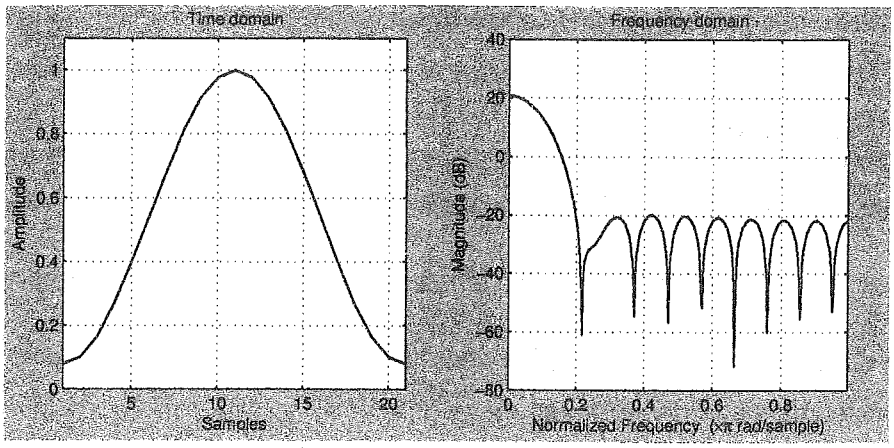


Figure 15.3.4: Length 21 Hamming window.

15.3.2.2 Kaiser

The Kaiser window is defined by:

$$w_k = \begin{cases} \dfrac{I_o\left[\beta\sqrt{1-\left|\frac{n}{N}\right|^2}\right]}{I_o(\beta)} & \text{for } -N \le k \le N \\ 0 & \text{elsewhere} \end{cases}$$

where β is a parameter of the window used to control the level of the sidelobes, and $I_o(x)$ is the modified Bessel function of the first kind with zero order and can be computed by:

$$I_o(x) = 1 + \sum_{m=1}^{\infty} \left(\frac{1}{m!} \left(\frac{1}{2} x \right)^m \right)^2$$

In practice the infinite sum is replaced with with a sufficient large sum to yield the required level of accuracy.

The range of β is any positive real number and is used to control the level of the sidelobes. A large value of β implies a small side lobe, and vice versa. This can be seen in Figure 15.3.5.

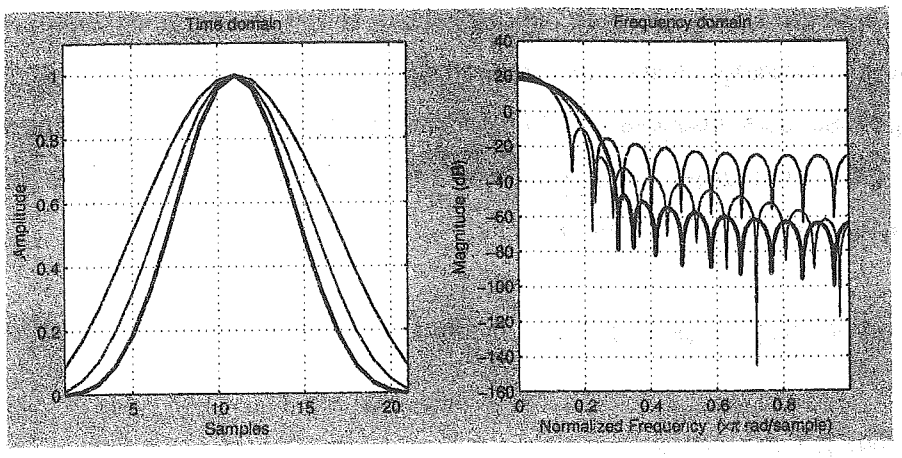


Figure 15.3.5: Length 21 Kaiser window for $\beta = 4,\ 2\pi,\ 9$ (blue, green and red respectively).

15.4 *minimax* filters

From the study of real valued polynomials we have the Chebyshev equi-oscillation theorem, which we'll now state without proof:

Theorem. *Let f be a continuous real function on the range $[a, b]$. Among all the polynomials of degree $\leq n$, the polynomial g that minimizes the maximum error $\zeta = \max \|f - g\|$ (on the range $[a, b]$) has $n + 2$ points $a \leq x_0 < x_1 < \cdots < x_{n+1} \leq b$ such that $f(x_i) - g(x_i) = \sigma(-1)^i\zeta$ where $\sigma = \pm 1$.*

In words this theorem states that if maximum error between a target real valued function, f, and some degree n polynomial, g on some range $[a, b]$ is ζ. Then the g that has the smallest error ζ is that polynomial that oscillates between $f \pm \zeta$, $n+2$ times on the range $[a, b]$. There are methods available to compute said polynomial. This is illustrated in Figure 15.4.1.

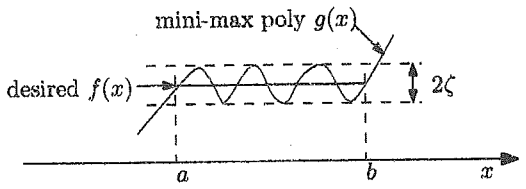


Figure 15.4.1: The Chebyshev equi-oscillation theorem says the polynomial that minimizes the maximum error oscillates around the target by exactly $\pm$ this maximum error.

If we consider the target function, f, to be the desired amplitude spectrum of a filter, then we can use these methods to design an order n polynomial g that minimizes the maximum error between the desired amplitude spectrum and g.

Furthermore we can easily compute the coefficients of an FIR filter to implement any amplitude spectrum matching a finite order polynomial.
Combining these facts produces algorithms for designing FIR filters that minimize the maximum error between a target specification and some FIR filter's response. The most famous of these being the Parks-McClellan algorithm.

Filters based on this principle are called *minimax* filters.

15.4.1 Weighted regions

Filters generally have at least 2 regions (in the frequency domain) that we place constraints on, i.e. a Low-Pass-Filter will have

- a *passband* where we wish to allow frequencies pass
- a stopband where we wish to stop frequency components from passing.

Acknowledging that the world is an imperfect place, typically we might say that the gain in the passband should be "close to unity", and that the gain in the stopband should be "close to zero".
In addition there is a transition band located between the pass and stopbands where we don't place any particular restriction on the gain of the filter.
The minimax FIR filter design procedure outlined previously can be extended to cover several bands, i.e. not just one contiguous range $[a, b]$ allowing us, for example, to specify that we'd like to minimize the maximum error occurring in both the pass and stop bands at the same time.
However, depending on the application, we usually "care" more about the error in one band or the other. i.e. we might be able to tolerate a larger error in the stopband compared to the passband.
For this reasons most filter design computer packages allow you to specify different error *weights* for each band of interest.
When the algorithm is computing the error between the desired function and the actual polynomial, these weights are used to scale the error in the different regions before the minimization is performed. Therefore the regions with the larger weighs will have the smaller error after the procedure completes, i.e. the weighted maximum errors in each of the regions should be the same across all regions.

15.4.2 Example LPF

Consider a low pass filter with cut-off frequency $f_c = \frac{1}{8} f_s$, and look at the results from the Matlab® *firpm*[3] function for various different weight configurations.

15.4.2.1 Equal weights

Setting the weights in the passband and stopband to be the same, e.g. $[1, 1]$, then we get the response, for a 21 coefficient filter, shown in Figure 15.4.2, and the zoomed in version is shown in Figure 15.4.3.

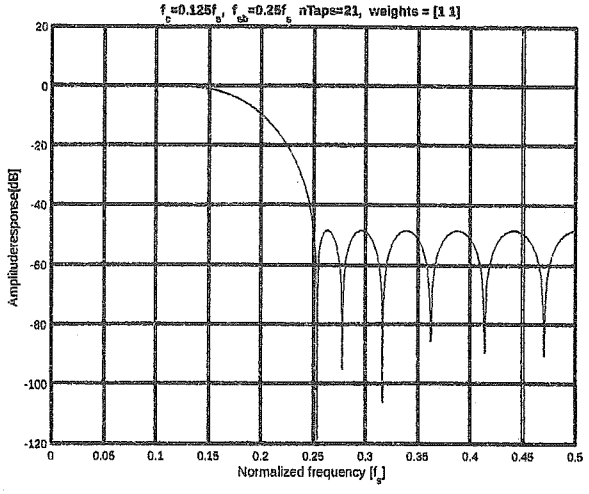


Figure 15.4.2: A LPF designed using the firpm function in Matlab® with equal weighting in the passband and stopband. The achieved stopband attenuation is approximately -41dB.

[3]firpm is the Parks-McClellan algorithm for designing FIR filters.

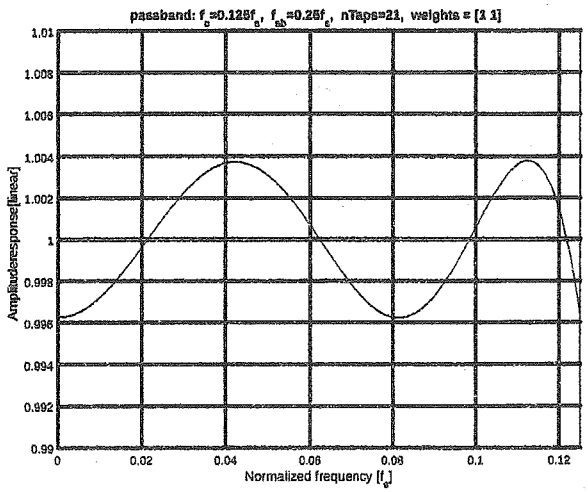


Figure 15.4.3: The passband response of Figure 15.4.2 on a linear scale. Notice the equal-ripple behavior.

15.4.2.2 Non-equal weights

Setting the weights in the passband and stopband to be $[1, 100]$ we are saying that maximum error in the stopband $\times 100$ should equal the maximum error in the passband', i.e. we allow must more error in the passband than the stopband.

Thus for the same length filter we'd expect the ripple in the passband to increase and the attenuation to the stopband to improve. Indeed this is what happens as can be seen in Figure 15.4.4,and 15.4.5.

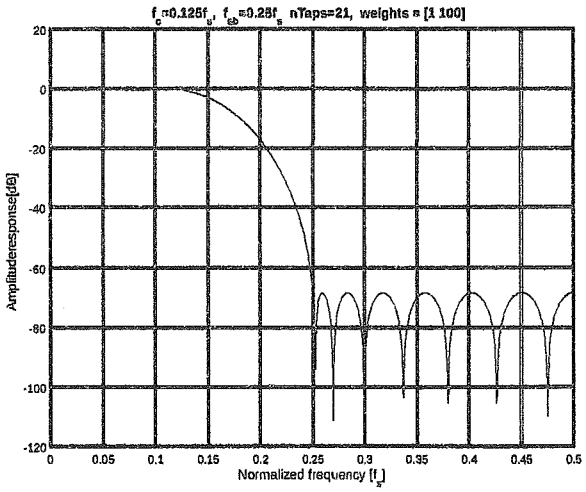


Figure 15.4.4: A LPF designed using the firpm function in Matlab® with non-equal weighting in the passband and stopband. The achieved stopband attenuation is approximately -62dB.

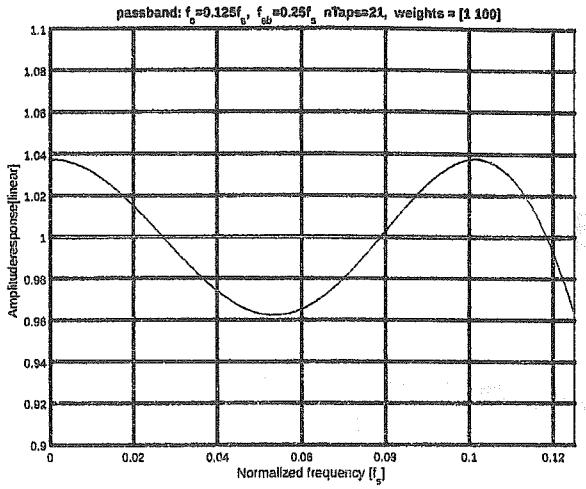


Figure 15.4.5: The passband response of Figure 15.4.4 on a linear scale. Notice how this has got worse compared to the equal weighting case.

15.4.3 More complex example

Consider a the Band Pass Filter (BPF) with the following specification:

	stopband 1	passband	stopband 2
frequency $[f_s]$	$0 \leftrightarrow 0.1$	$0.15 \leftrightarrow 0.3$	$0.35 \leftrightarrow 0.5$
Gain [linear]	0	1	0
Weights	50	1	5

Table 15.1: The specification of a BPF

The resulting amplitude spectrum for a 81 coefficient FIR filter is shown in Figure 15.4.6.

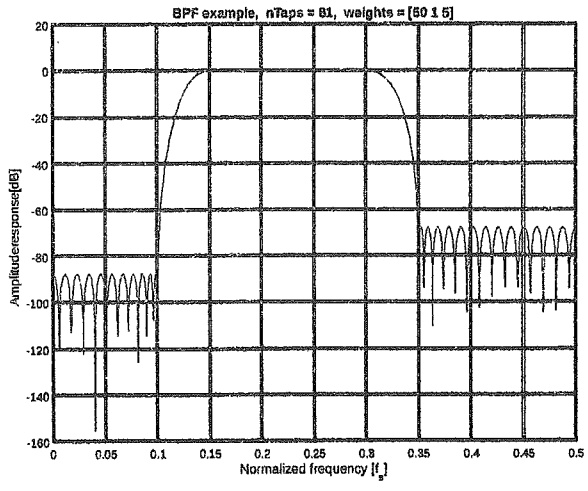


Figure 15.4.6: The response of the filter with specification given in Table 15.1.

Chapter 16

Infinite Impulse Response

16.1 Introduction

The output of a FIR filter is just the weighted sum of the current and (a finite number of) previous inputs. However the output in an IIR filter is also a fucntion of a finite number of previous outputs, the $\{a_l\}$ coefficients in equation 16.1.1 are non zerps.

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} - \sum_{l=1}^{L} a_l y_{n-l} \tag{16.1.1}$$

This is a Difference Equation and is the basis of digital filtering.

The values (sometimes known as coefficients) of $\{a_n\}$ and $\{b_n\}$ are constants[1].

$\{x_n\}$ is the input, and

$\{y_n\}$ is the output, i.e. the filtered signal.

We now redraw the IIR implementation in Figure An obvious way to implement such a filter is shown in Figure 16.1.1.

[1] To simplify the math in Chapter 12 we also defined $a_0 = 1$.

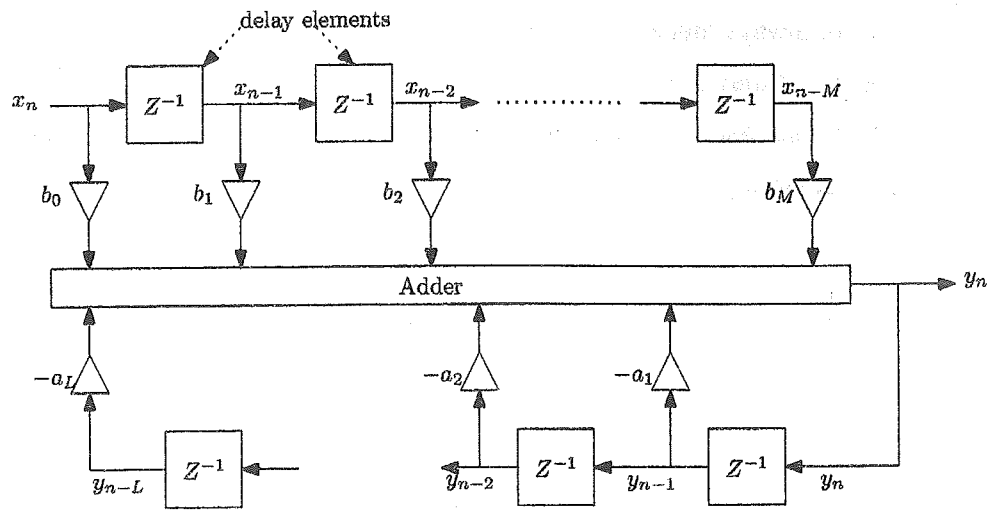


Figure 16.1.1: Direct form implementation of an IIR filter.

16.2 Recap: The transfer function

Summarizing previous results (from Chapter 12), we can write the transfer function in any of the following equivalent ways:

$$\begin{aligned}\tilde{H}(z) &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L}} \\ &= \frac{\tilde{B}(z)}{\tilde{A}(z)} \\ &= b_0 z^{L-M} \frac{\prod_{m=1}^{M}(z - z_m)}{\prod_{l=1}^{L}(z - p_l)}\end{aligned}$$

where $\tilde{A}(z)$ and $\tilde{B}(z)$ are the z-transforms of $\{a_n\}$ and $\{b_n\}$ respectively,
$\{z_m\}$ are the M zeros of $\tilde{H}(z)$, (the roots of $\tilde{B}(z)$),
$\{p_l\}$ are the L poles of $\tilde{H}(z)$, (the roots of $\tilde{A}(z)$), and
Additionally (for causal filters) there is either $L - M$ zeros (if $L > M$) or $M - L$ poles (if $M > L$) at the origin ($z = 0$).

The position of the poles and zeros of a filter have a huge impact on both the frequency response and stability and form the basis of much analysis and synthesis of digital filters. Often we like to visualize them on a pole-zero diagram like the one shown in Figure 16.2.1. As previously noted, for clarity we usually don't draw the poles or the zeros at the origin

because a) they are always there for causal filterts and just serve to confuse the issue, and b) it can be shown that they don't affect the magnitude response nor the stability (however they do effect the phase response by adding or subtracting a fixed group delay - see the shift property of the z-transform).

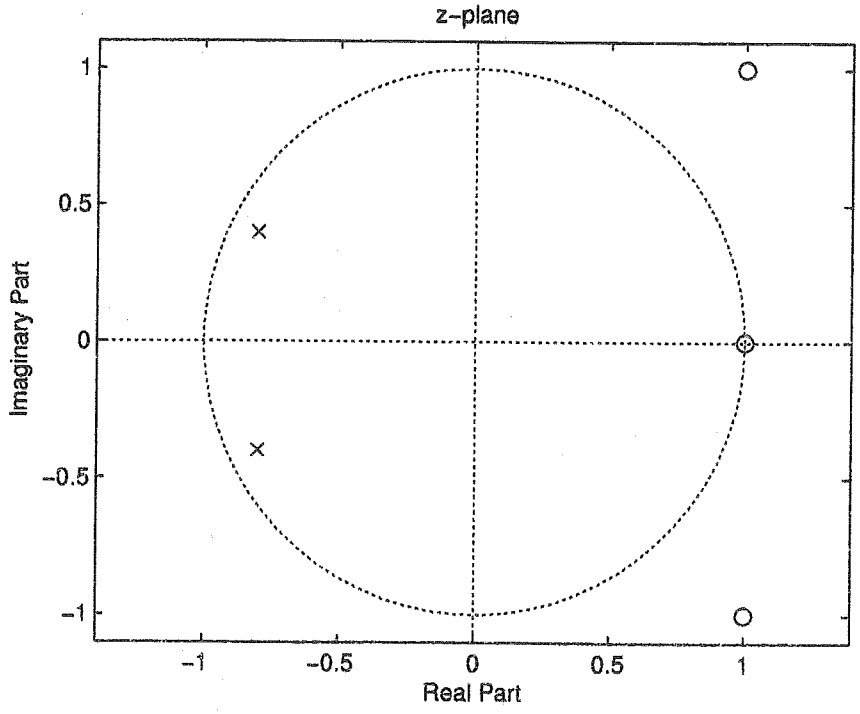


Figure 16.2.1: Example IIR pole-zero diagram. Here we have two poles (the x's) and three zeros (the o's). The zero at the origin is not shown.

Note, again ignoring the poles at the origin, FIR filters only have zeros and are sometimes called *all-zero* filters, and as the output does not depend on previous outputs they are also sometimes called *non-recursive filters*.

IIR filters are sometimes called *recursive filters*.

16.3 Example

Consider the difference equation:

$$y_n = x_n + 0.8y_{n-1}$$

We will now compute it's transfer function, pole-zero diagram, impulse response, unit step response, and its frequency response.

16.3.1 Transfer function

Take the z-transform of both sized of the difference equation:

$$\tilde{Y}(z) = \tilde{X}(z) + 0.8z^{-1}\tilde{Y}(z)$$

Here we've used the shift property of the z-transform, i.e. $\mathcal{Z}[\{y_{n-1}\}] = z^{-1}\mathcal{Z}[\{y_n\}] = z^{-1}\tilde{Y}(z)$.

Now rearranging terms we get:

$$\frac{\tilde{Y}(z)}{\tilde{X}(z)} \triangleq \tilde{H}(z) = \frac{1}{1-0.8z^{-1}}$$

16.3.2 Pole-zero diagram

Rewrite as a ratio of polynomials of z, not z^{-1}:

$$\tilde{H}(z) = (z)\frac{1}{(z-0.8)}$$

Zeros: There is one at $z = 0$.

Poles: There is one at $z = 0.8$

This is shown in Figure 16.3.1, again we note that often we don't draw the pole/zero at the origin.

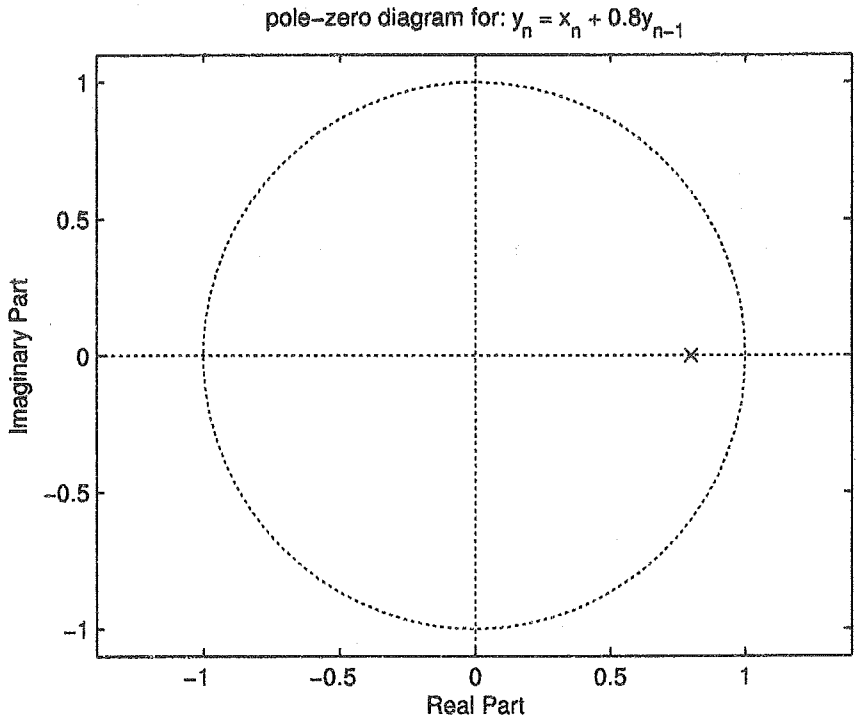


Figure 16.3.1: Pole-zero diagram for $y_n = x_n + 0.8y_{n-1}$ (poles and zero at the origin not shown).

16.3.3 Impulse response

What is the output, y_n, from the IIR filter to the following "impulse" input?

$$x_n = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{elsewhere} \end{cases}$$

Here we will assume $y_n = 0$ for $n < 0$, i.e. before the pulse arrives the all previous inputs and outputs are zero.

n	x_n	$x_n + 0.8y_{n-1}$	y_n
-1	0	0	0
0	1	$x_0 + 0.8y_{-1}$	1
1	0	$x_1 + 0.8y_0$	0.8
2	0	$x_2 + 0.8y_1$	$(0.8)^2$
3	0	$x_3 + 0.8y_2$	$(0.8)^3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	0	$x_n + 0.8y_{n-1}$	$(0.8)^n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	0	$\vdots$	0

This is shown in Figure 16.3.2.

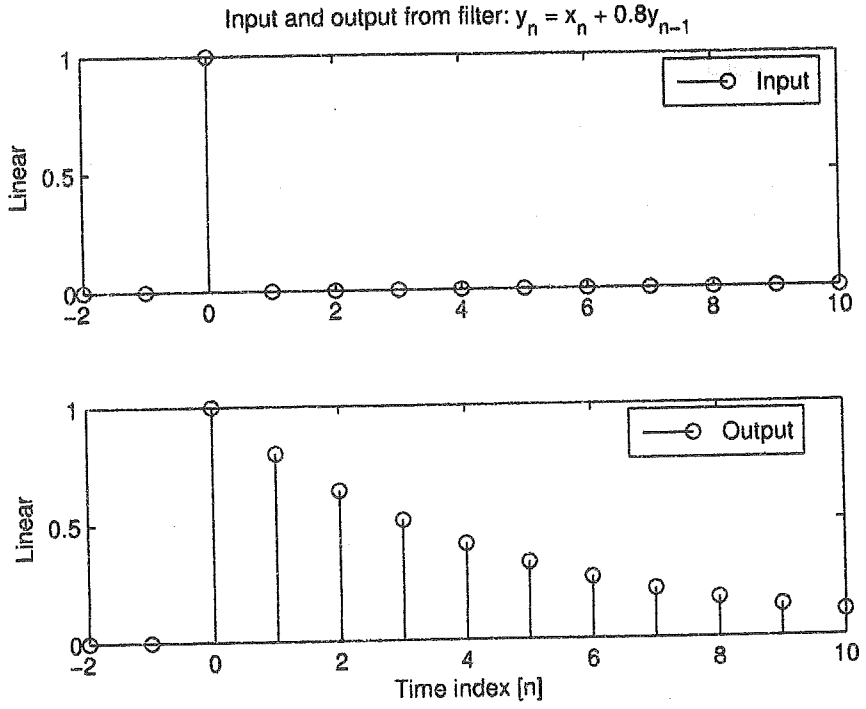


Figure 16.3.2: Impulse response of $y_n = x_n + 0.8y_{n-1}$

16.3.4 Step response

What is the output, y_n, from the IIR filter to the following "impulse" input?

$$x_n = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

Here, agin, we will assume $y_n = 0$ for $n < 0$, i.e. before the pulse arrives the all previous inputs and outputs are zero.

n	x_n	$x_n + 0.8y_{n-1}$	y_n
-1	0	0	0
0	1	$x_0 + 0.8y_{-1} = 1 + 0.8 \times 0$	1
1	1	$x_1 + 0.8y_0 = 1 + 0.8 \times 1$	1.8
2	1	$x_2 + 0.8y_1 = 1 + 0.8 + 0.8^2$	2.44
3	1	$x_3 + 0.8y_2 = 1 + 0.8 + 0.8^2 + 0.8^3$	2.952
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	1	$x_n + 0.8y_{n-1} = \sum_{k=0}^{n} 0.8^k$	$\frac{1-0.8^{n+1}}{1-0.8}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	1	$= \lim_{n\to\infty} \sum_{k=0}^{n} 0.8^k$	5

This is shown in Figure 16.3.3.

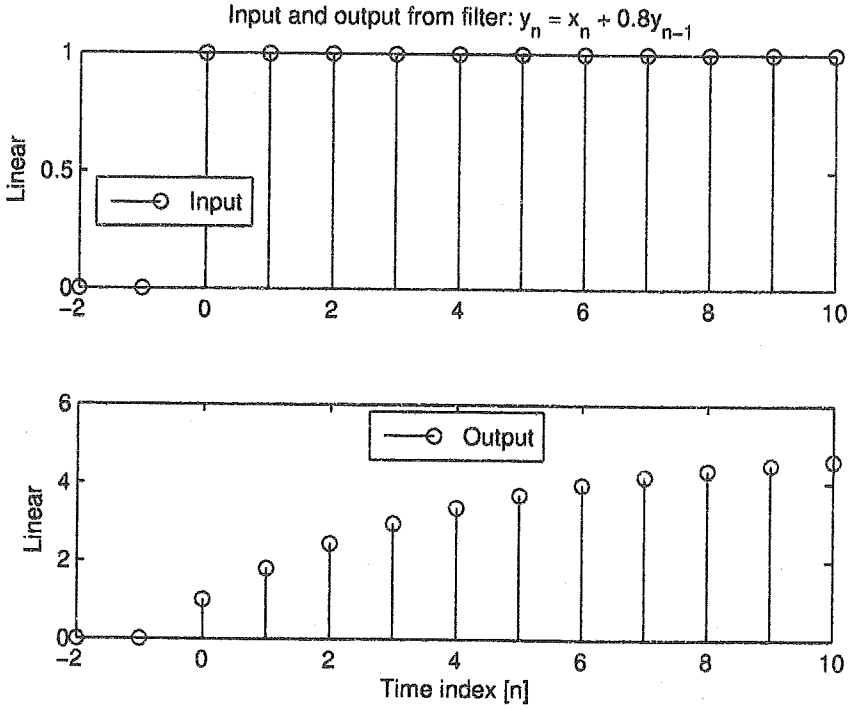


Figure 16.3.3: Step response of $y_n = x_n + 0.8y_{n-1}$

16.3.5 Frequency response

Exercise:

For $y_n = x_n + 0.8y_{n-1}$, show that:

$$\left|\bar{H}\left(j\omega\right)\right| = \frac{1}{\sqrt{1.64 - 1.6\cos\omega T}} \quad \text{and} \quad \angle\bar{H}\left(j\omega\right) = -\tan^{-1}\left(\frac{0.8\sin\omega T}{1 - 0.8\cos\omega T}\right)$$

as shown in Figure 16.3.4.

Solution:

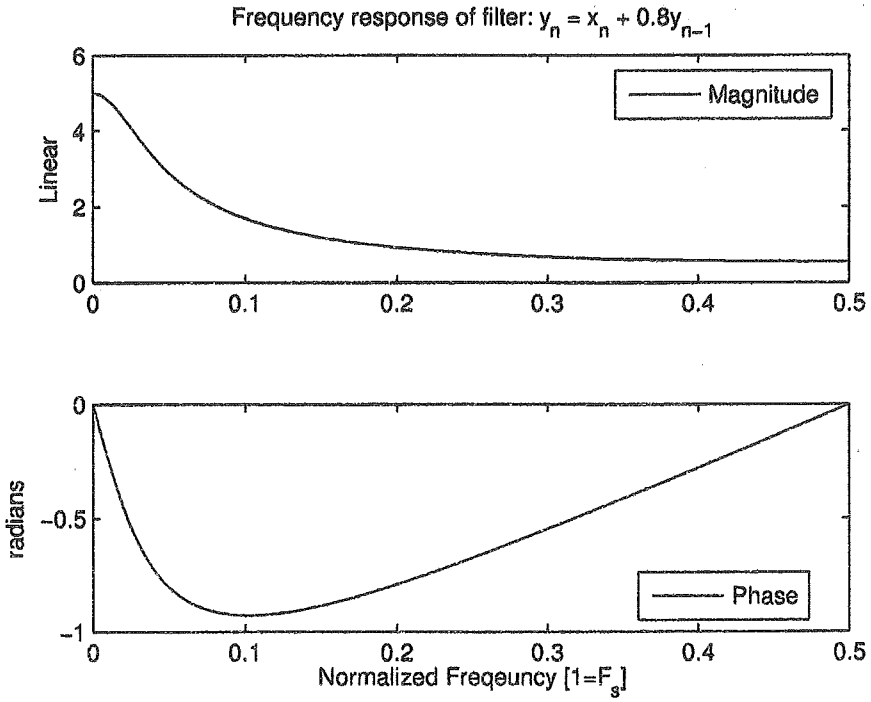


Figure 16.3.4: Frequency response of $y_n = x_n + 0.8y_{n-1}$

16.3.6 Generalization

Exercise:

Generalize the above expressions for $\left|\bar{H}\left(j\omega\right)\right|$ and $\angle\bar{H}\left(j\omega\right)$ for the case where $y_n = x_n + \alpha y_{n-1}$.

Also compute the value of α that results in a power gain of -3dB (compared to DC) at some frequency ω_0.

Solution:

16.4 Significance of pole and zeros

The position of the poles and zeros, to the trained eye, can be used to quickly guess the magnitude and phase response of a filter, as we will now explain.

16.4.1 Amplitude response

The amplitude (or magnitude) response can be written as a function of the poles and zeros by evaluating $\left|\tilde{H}(z)\right|$ along the units circle, as follow:

$$\begin{aligned}
\tilde{H}(z) &= b_0 z^{L-M} \frac{\prod_{m=1}^{M}(z-z_m)}{\prod_{l=1}^{L}(z-p_l)} \\
\Rightarrow \left|\tilde{H}(j\omega)\right| &= \left|\tilde{H}\left(z=e^{j\omega T}\right)\right| \\
&= |b_0|\left|\left(e^{j\omega T}\right)^{L-M}\right| \frac{\prod_{m=1}^{M}\left|e^{j\omega T}-z_m\right|}{\prod_{l=1}^{L}\left|e^{j\omega T}-p_l\right|}
\end{aligned}$$

Note that the magnitude of $\left(e^{j\omega T}\right)^{L-M}$ is 1 so we have:

$$\left|\bar{H}\left(j\omega\right)\right| = |b_0| \frac{\prod_{m=1}^{M} \left|e^{j\omega T} - z_m\right|}{\prod_{l=1}^{L} \left|e^{j\omega T} - p_l\right|}$$

i.e. in terms of poles and zeros, the ones at $z = 0$ do not contribute to the magnitude response - this is one of the main reasons they are often ignored during filter design and analysis (but you do need to be careful and know what you are doing).

Often we ignore the constant $|b_0|$ term as it is a constant and effects all frequencies equally:

$$\left|\bar{H}\left(j\omega\right)\right| \propto \frac{\prod_{m=1}^{M} \left|e^{j\omega T} - z_m\right|}{\prod_{l=1}^{L} \left|e^{j\omega T} - p_l\right|}$$

Graphically each of the numerator terms $\left|e^{j\omega T} - z_m\right|$ is just the distance from a point on the unit circle and the m^{th} zero. Likewise the denominator terms are the distance from the poles.

So to know what the amplitude response is at some frequency $\omega = \omega_o$ do the following:

- Compute the corresponding point on the unit circle ($e^{j\omega T}$)
- Compute the distances to the zeros then compute their product (ignoring zeros at the origin)
- Compute the distances to the poles then compute their product (ignoring zeros at the origin)
- The amplitude response is proportional to the quotient of the these two products.

Sounds difficult!!, but its not for a small number of pole and or zeros.
One thing that really helps is knowing that point $z = 1 + 0j$ on the z-plane corresponds to DC, and $z = -1 + 0j$ corresponds to $\frac{1}{2}f_s$, and moving in a anti-clockwise motion between these points gives all the frequencies in between - see example below.

16.4.2 Phase response

The phase response can be written as a function of the poles and zeros as follow:

$$\begin{aligned} \tilde{H}(z) &= b_0 z^{L-M} \frac{\prod_{m=1}^{M}(z-z_m)}{\prod_{l=1}^{L}(z-p_l)} \\ \Rightarrow \angle\tilde{H}(z) &= \angle b_0 + \angle z^{L-M} + \sum_{m=0}^{M} \angle(z-z_m) - \sum_{l=0}^{L} \angle(z-p_l) \end{aligned}$$

Assuming real valued filter, then b_0 is real, and therefore $\angle b_0 = 0$.

Evaluating along the unit circle to get the amplitude response, we have:

$$\angle\tilde{H}(j\omega) = \omega(L-M)T + \sum_{m=0}^{M} \angle\left(e^{j\omega T} - z_m\right) - \sum_{l=0}^{L} \angle\left(e^{j\omega T} - p_l\right)$$

where each of the $\angle\left(e^{j\omega T} - z_m\right)$ terms are the angle between a point on the unit circle and the m^{th} zero. Likewise the $\angle\left(e^{j\omega T} - p_l\right)$ terms are the angles from the poles.

So to know what the phase response is at some frequency $\omega = \omega_o$ do the following:

- Compute the corresponding point on the unit circle ($e^{j\omega T}$)
- Compute the angles to the zeros then compute their sum
- Compute the angles to the poles then compute their sum
- The phase response is the difference between these two sums plus a line term $\omega(L-M)T$

As discussed in in Chapter 14, the addition of $\omega(L-M)T$, a linear function of ω simply introduces a fixed group delay to the filter.

16.4.3 Example

Consider the zero-pole arrangement shown in Figure 16.4.1.

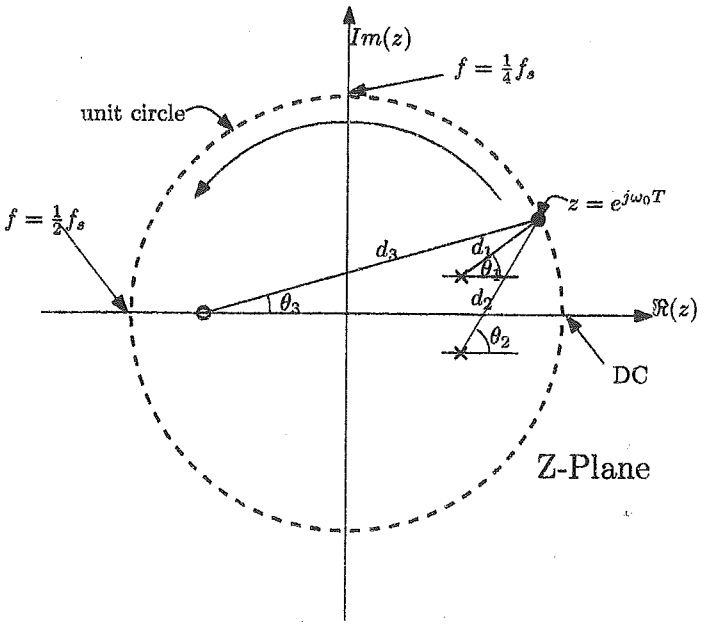


Figure 16.4.1: Computation of magnitude and phase response graphically from the pole-zero diagram.

Based on our previously derived equations we know that the frequency response at $\omega = \omega_0$, is:

$$\begin{aligned} \left|\bar{H}\left(j\omega_0\right)\right| &\propto \frac{d_3}{d_1 d_2}, \text{ and} \\ \angle\bar{H}\left(j\omega_0\right) &= \theta_3 - \theta_1 - \theta_2 + \text{linear function of } \omega \end{aligned}$$

This is a low pass filter - how can I tell?

Chapter 17

Stability of IIR filters

17.1 Introduction

What do we mean by stability?
Generally in digital filters we are concerned with BIBO (Bounded Input, Bounded Output) stability which, as its names suggests, says that provided the input to a system is bounded, i.e. $\leq C$ for some finite C, then so too is its output.

17.2 Stability criteria

17.2.1 Time domain criteria

Theorem. *Let $\{h_j\}$ for $-\infty < j < +\infty$ be the impulse response of a digital filter the a necessary & sufficient conditions for stability is that:*

$$\sum_{j=-\infty}^{+\infty} |h_j| \; < \; \infty$$

Proof. First we'll prove the necessary part.
To do this we take the worst case signal that could be applied:

$$x_{-j} \; = \; C \times \text{sign}\,(h_j)$$

Then, by convolution, at some time $n = 0$ we get the following filter output:

$$\begin{aligned} y_0 &= \sum_{j=-\infty}^{+\infty} h_j x_{-j} \\ &= C \sum_{j=-\infty}^{+\infty} h_j \text{sign}\,(h_j) \\ &= C \left(\sum_{j=-\infty}^{+\infty} |h_j| \right) \end{aligned}$$

This is bounded if $\sum_{j=-\infty}^{+\infty} |h_j| < \infty$ thus, it is certainly a necessary condition.

Now we'll show that this is also a sufficient condition.

Suppose the input is such that $|x_n| \leq C$, i.e. a bounded input.

Then the output at any time n is:

$$\begin{aligned} y_n &= \sum_{j=-\infty}^{+\infty} h_j x_{n-j} \\ \Rightarrow |y_n| &\leq \sum_{j=-\infty}^{+\infty} |h_j|\,|x_{n-j}| \\ &\leq C \left(\sum_{j=-\infty}^{+\infty} |h_j| \right) \end{aligned}$$

Hence the output is always finite if $\sum_{j=-\infty}^{+\infty} |h_j|$ is finite. Thus the condition:

$$\sum_{j=-\infty}^{+\infty} |h_j| < \infty$$

is sufficient to guarantee BIBO.

But we already established that it is also necessary - this completes the proof. □

17.2.1.1 Example

Consider the filter described by the difference equation:

$$y_n = x_n + \alpha y_{n-1}$$

This is exactly as per the example Section 16.3 but instead of a pole at $z = 0.8$ we now have one at $z = \alpha$. That example can be generalized to yield the following impulse response:

$$h_j = \begin{cases} 0 & j < 0 \\ \alpha^j & j \geq 0 \end{cases}$$

Our BIBO condition can now be rewritten as:

$$\begin{aligned} \sum_{j=-\infty}^{+\infty} |h_j| &< \infty \\ \Rightarrow \sum_{j=0}^{+\infty} |\alpha^j| &< \infty \end{aligned}$$

But $\sum_{j=0}^{+\infty} |\alpha^j|$ converges (and if thus finite) for $|\alpha| < 1$.

It diverges for $|\alpha| \geq 1$.

Therefore we can say that the system is stable if the magnitude of the pole α is < 1, i.e. inside the unit circle on the z-plane. This is illustrated in Figure 17.2.1.

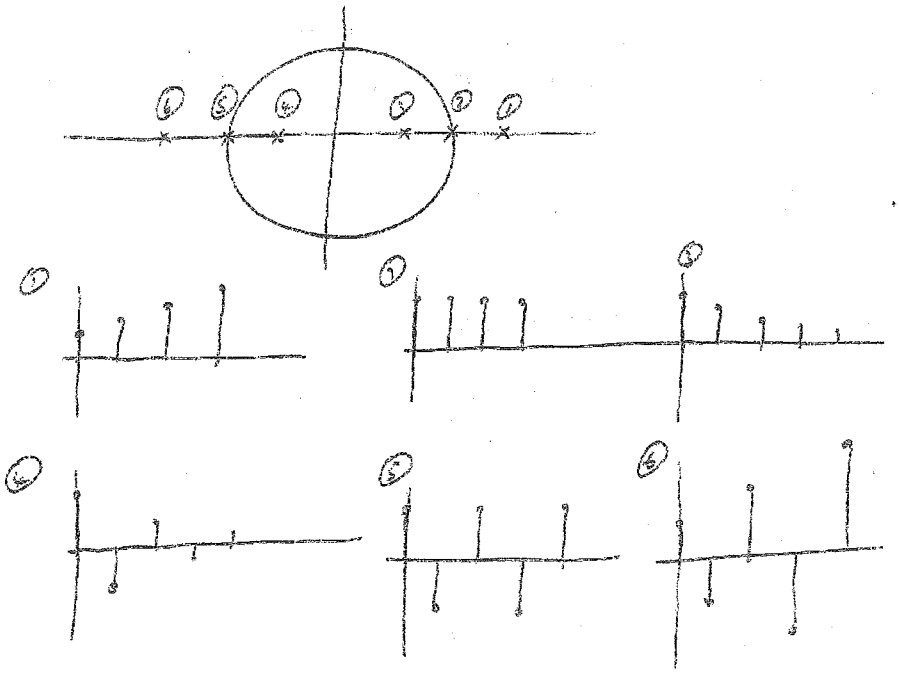


Figure 17.2.1: The impulse response $\{h_j\}$ for various pole position:
1) positive and outside unit circle ⇒ unstable.
2) positive and on the unit circle ⇒ unstable
3) positive and inside unit circle ⇒ stable
4) negative and inside unit circle ⇒ stable
5) negative and on unit circle ⇒ unstable
6) negative and outside unit circle ⇒ unstable

This result, whilst derived using the time domain criteria, hints at a more general fundamental criteria involving the position of the poles.

17.2.2 Z-domain domain criteria

Theorem. *For a stable digital filter it is a sufficient condition that all poles lie within the unit circle.*

Proof.

Guessing that the stability depends on the pole (and not the zeros) we write the transfer function in a hybrid fashion showing the numerator polynomial and the poles:

$$\begin{aligned}\tilde{H}(z) &= z^L \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{\prod_{l=1}^{L}(z - p_l)} \\ &= \tilde{G}(z) \sum_{m=0}^{M} b_m z^{-m}\end{aligned}$$

where

$$\tilde{G}(z) \triangleq \frac{z^L}{\prod_{l=1}^{L}(z-p_l)}$$

So the overall transfer function can be split into the sum of several (scaled & delayed) transfer functions, $\tilde{G}(z)$ i.e. the parallel combination of many filters. If they are all stable, then of course, the overall sum is also stable (the sum of M finite number is finite).

As the delay and scaling factor in each of these parallel filters does not impact on their stability, we just need to verify that $\tilde{G}(z)$ itself is stable to guarantee that the overall transfer function, $\tilde{H}(z)$, is stable. Therefore we will now focus on $\tilde{G}(z)$:

Any quotient with a factorisable denominator can be written as the sum of partial fractions, e.g.

$$\frac{41}{42} = \frac{41}{2\times 3\times 7} = \frac{1}{2}+\frac{1}{3}+\frac{2}{7}$$

We can do this for $\tilde{G}(z)$, luckily we don't care about the precise values of the numerators so we'll just call them R_i, e.g.:

$$\begin{aligned}\tilde{G}(z) &= \frac{z^L}{\prod_{l=0}^{L}(z-p_l)} = \sum_{l=0}^{L} R_l\left(\frac{z}{z-p_l}\right)\\ &= \sum_{l=0}^{L} R_l\left(\frac{1}{1-p_l z^{-1}}\right)\end{aligned}$$

But each of these bracketed terms is just a one-pole-filter (with a pole at $z = p_l$) which we already established in Example 17.2.1.1 that each of these is stable iff their pole lies within the unit circle. Furthermore if they are all stable (all p_l lie within the unit circle) then $\tilde{G}(z)$, being a weighed sum of them, is itself stable, which in turns implies $\tilde{H}(z)$ is also stable. Therefore it is a sufficient condition for stability that all poles lie within the unit circle, completing the proof. □

Note we have not proven that this is a necessary condition for stability.

Chapter 18

IIR filter design

18.1 Introduction

We seek the values of $\{b_k\}$ and $\{a_l\}$ such that the difference equation

$$y_n = \sum_{k=0}^{M} b_k x_{n-k} - \sum_{l=1}^{L} a_l y_{n-l}$$

yields some desired frequency domain response.

If phase response is important to use, then typically we'd use an FIR filter which, for example, can easily have a linear phase response, or we'd design an IIR filter and cascade it by a phase correcting filter (e.g. an all pass filter).
Either way, when designing IIR filters we often are just concerned with the magnitude response, $\left|\tilde{H}\left(j\omega\right)\right|$.

18.2 Recap

A filter's magnitude response at any given frequency, ω_0, can be computed by measuring the distance from the corresponding point on the unit circle, $e^{j\omega_0 T}$, to each of the poles and zeros and then computing

$$\frac{\prod_{m=1}^{M} \left|e^{j\omega T} - z_m\right|}{\prod_{l=1}^{L} \left|e^{j\omega T} - p_l\right|}$$

Graphically this is shown in Figure 18.2.1. Here the magnitude at $\omega = \omega_0$, is:

$$\left|\bar{H}\left(j\omega_0\right)\right| \propto \frac{d_3}{d_1 d_2}$$

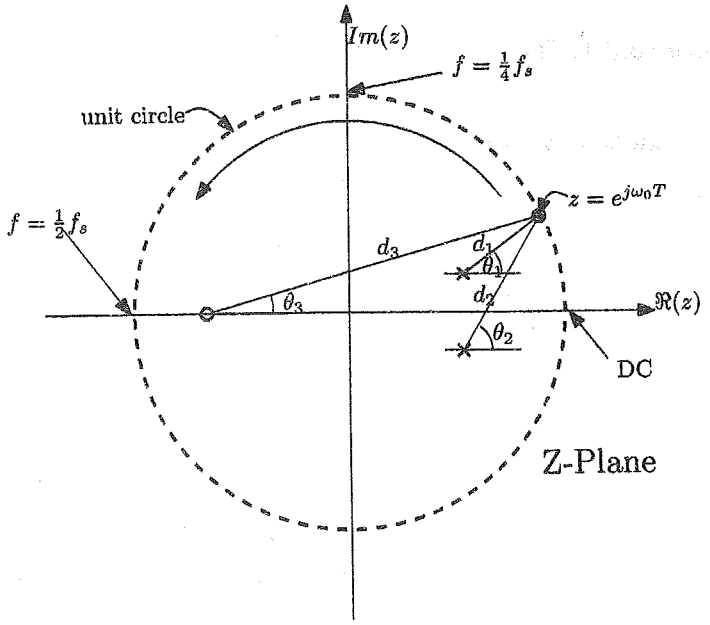


Figure 18.2.1: Computation of magnitude and phase response graphically from the pole-zero diagram.

We will now use this to design some simple (but effective) IIR filters.

18.3 Pole zero placement technique

There are many IIR filter design methods, however here we will only consider the pole zero placement technique.
Specifically we will look at the Notch filter and the resonator.

18.3.1 The notch filter

We use a notch filter to 'notch' out some narrow band of frequencies, but otherwise it should 'pass' all others. This is useful if the signal has being corrupted by some tone-like interfere (e.g. mains power supply noise at 50Hz). The ideal magnitude response of a notch filter is shown in Figure 18.3.1.

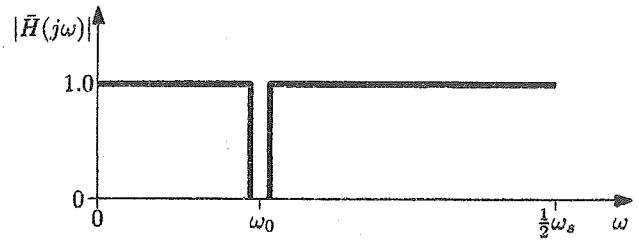


Figure 18.3.1: The ideal magnitude response of a notch filter.

Suppose we want a real filter with a notch at $\frac{1}{6}^{th}$ the half sampling frequency, i.e. $f_0 = \frac{1}{6}\frac{f_s}{2}$, and therefore $\omega_0 T = \frac{1}{6}\pi$.

So we require $\left|\tilde{H}(z)\right|$ to be zero when $z = e^{\jmath\frac{1}{6}\pi}$.

Step 1

One approach would be to have an FIR filter with a zero at $z = e^{\jmath\frac{1}{6}\pi}$. This would guarantee that the magnitude response is exactly zero at the required frequency.
Note, that but because we want a real filter there will also be on at the conjugate position, i.e. at $z = e^{-\jmath\frac{1}{6}\pi}$ (and the 2 poles at the origin). Thus $\tilde{H}(z)$ would be

$$\begin{aligned}\tilde{H}(z) &= b_0 z^{-2}\left(z - e^{\jmath\frac{1}{6}\pi}\right)\left(z - e^{-\jmath\frac{1}{6}\pi}\right) \\ &= b_0\left(z^{-2} - \sqrt{3}z^{-1} + 1\right)\end{aligned}$$

The value of b_0 is chosen could be chosen to give a specific gain at some frequency or we could just let $b_0 = 1$ for now and scale the coefficients later.

i.e. the $\{b_k\}$ coefficients are $\{1, -\sqrt{3}, 1\}$.

Using Matlab we can compute and draw the corresponding frequency response.
This, along with the pole-zero diagram is shown in Figure 18.3.2.

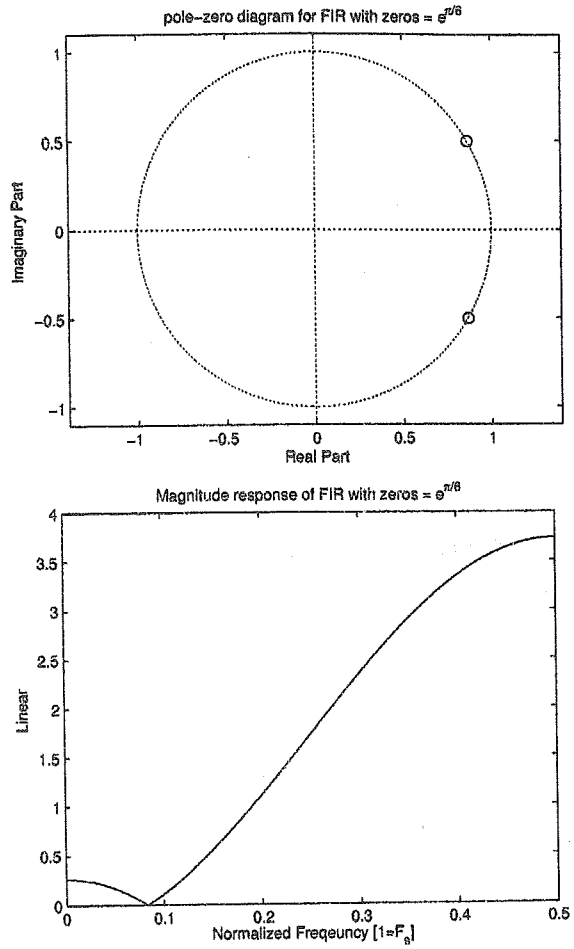


Figure 18.3.2: The pole-zero diagram (poles at origin not shown) and magnitude response of a FIR notch filter with only zeros.

Notice how Figure 18.3.2 looks pretty much nothing like the ideal notch filter shown in Figure 18.3.1 - the next step is magic, and will fix this...

Step 2

The trick is to add some poles very close to the zeros, specifically make the poles lie on the same radial line from the origin as the two zeros, i.e. if the zeros are at $z = e^{\pm j\omega_0 T}$, then place the poles at $z = R.e^{\pm j\omega_0 T}$ as per Figure 18.3.3.

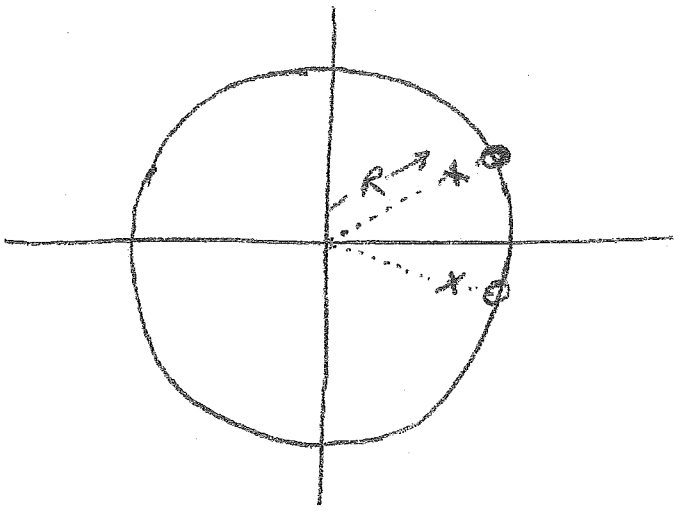


Figure 18.3.3: Place two conjuage poles near the zeros just inside the unit circle and having the same angle as the zeros.

The transfer function is:

$$\begin{aligned}\tilde{H}(z) &= \frac{\left(z - e^{j\omega_0 T}\right)\left(z - e^{-j\omega_0 T}\right)}{\left(z - R.e^{j\omega_0 T}\right)\left(z - R.e^{-j\omega_0 T}\right)} \\ &= \frac{1 - 2\cos\left(\omega_0 T\right) z^{-1} + z^{-2}}{1 - 2R\cos\left(\omega_0 T\right) z^{-1} + R^2 z^{-2}}\end{aligned}$$

The proof is an exercise.

The magnitude of this, along with the pole-zero diagram is shown in Figure 18.3.4.

Now that looks a lot better.

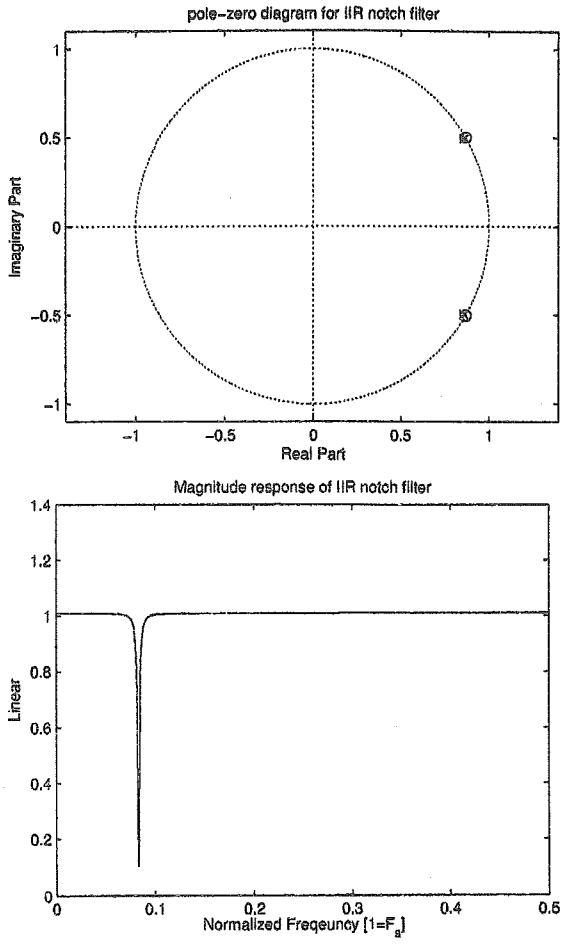


Figure 18.3.4: The pole-zero diagram and magnitude response of an IIR notch filter with zeros and nearby poles.

18.3.2 The digital resonator

Resonator definition (from wikipedia):

> "A resonator is a device or system that exhibits resonance or resonant behavior, that is, it naturally oscillates at some frequencies, called its resonant frequencies, with greater amplitude than at others."

Essentially this is almost like the opposite of a notch filter, i.e. instead of having a zero at some frequency we'd like to have a large amplitude response at some frequency.
We could place a pole on the unit circle at $z = e^{j\omega_0 T}$, corresponding to the desired resonant frequency ω_0, but this would be unstable, so we must instead place said pole just inside the unit circle at $z = R.e^{j\omega_0 T}$, for $|R| < 1$.
In order to make it real we must also have a 2^{nd} conjugate pole at $z = R.e^{-j\omega_0 T}$.

The transfer function is given by:

$$\tilde{H}(z) = \frac{z^2}{(z - R.e^{j\omega_0 T})(z - R.e^{-j\omega_0 T})}$$

This can be rewritten as (exercise):

$$\tilde{H}(z) = \frac{1}{1 - 2R\cos(\omega_0 T)\, z^{-1} + R^2 z^{-2}}$$

The magnitude of this and the zero-pole plot is shown in Figure 18.3.5.

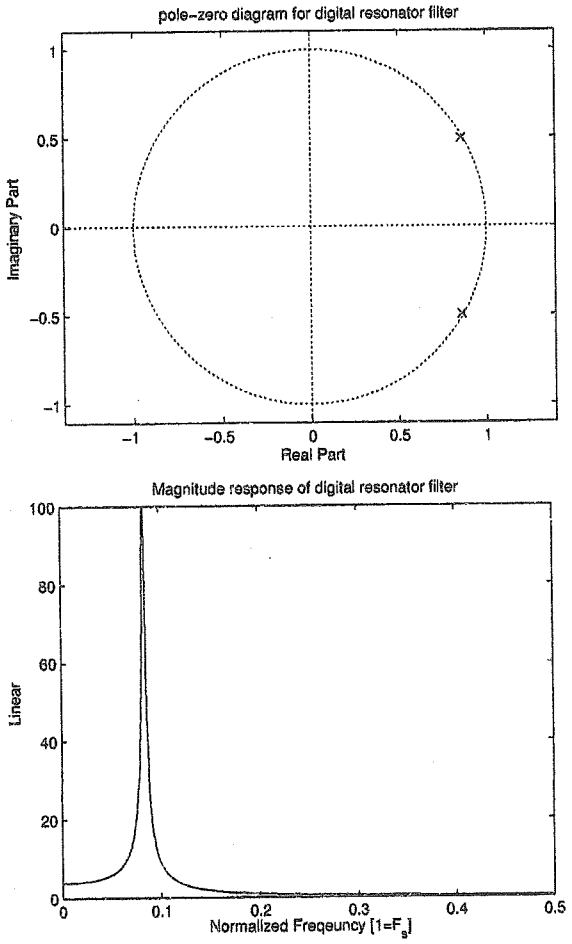


Figure 18.3.5: The pole-zero diagram and magnitude response of a 2^{nd} order digital resonator.

18.3.2.1 Impulse response

As the poles of this filter are very close to the unit circle, we could imagine that it is close to being unstable, lets look at the unit impulse response and see it's behavior to even the

simplest of stimulus.

Start with the pole-zero description, and decompose into the sum of 2 impulse response using the method of partial fractions:

$$\begin{aligned}\tilde{H}(z) &= \frac{z^2}{(z-z_0)(z-z_1)} \\ &= z^2\left(\frac{A}{z-z_0}+\frac{B}{z-z_1}\right)\end{aligned}$$

But what are A and B?

Exercise: Show that A and B are:

$$A = \frac{1}{j2R.\sin\omega_0 T}, \text{ and } B = -A$$

and therefore:

$$\begin{aligned}\tilde{H}(z) &= z^2 A\left(\frac{1}{z-z_0}-\frac{1}{z-z_1}\right) \\ &= \frac{z^2}{j2R.\sin\omega_0 T}\left(\frac{1}{z-R.e^{j\omega_0 T}}-\frac{1}{z-R.e^{-j\omega_0 T}}\right) \\ &= \frac{z}{j2R.\sin\omega_0 T}\left(\frac{1}{1-R.e^{j\omega_0 T}z^{-1}}-\frac{1}{1-R.e^{-j\omega_0 T}z^{-1}}\right) \\ &= \frac{z}{j2R.\sin\omega_0 T}\left(\sum_{n=0}^{\infty}\left(R.e^{j\omega_0 T}z^{-1}\right)^n-\sum_{n=0}^{\infty}\left(R.e^{-j\omega_0 T}z^{-1}\right)^n\right)\end{aligned}$$

Note the two $n = 0$ terms cancel, and so don't contribute to the final result. Thus both summation limits can safely be changed to $n = 1 \to \infty$. We can then substitute $k = n - 1$ to bring the limits back to $0 \to \infty$ and at the same time bring the z term in from outside the brackets making each summation "look" more like a z-transform:

$$\begin{aligned}\tilde{H}(z) &= \frac{z}{j2R.\sin\omega_0 T}\left(\sum_{k=0}^{\infty}\left(R.e^{j\omega_0 T}z^{-1}\right)^{k+1}-\sum_{k=0}^{\infty}\left(R.e^{-j\omega_0 T}z^{-1}\right)^{k+1}\right) \\ &= \frac{1}{j2R.\sin\omega_0 T}\left(\sum_{k=0}^{\infty}R^{k+1}e^{j(k+1)\omega_0 T}z^{-k}-\sum_{k=0}^{\infty}R^{k+1}e^{-j(k+1)\omega_0 T}z^{-k}\right)\end{aligned}$$

This is simply the z-transform of the impulse response $\{h_k\}$ given by:

$$\begin{aligned} h_k &= \frac{1}{j2R.\sin\omega_0 T}\left(R^{k+1}e^{j(k+1)\omega_0 T} - R^{k+1}e^{-j(k+1)\omega_0 T}\right) \quad k > 0 \\ &= \frac{R^{k+1}}{j2R.\sin\omega_0 T} 2j\sin\left((k+1)\omega_0 T\right) \\ &= \frac{R^k}{\sin\omega_0 T}\sin\left((k+1)\omega_0 T\right) \end{aligned}$$

The first term gets smaller (decays) with k (because $R < 1$), which essentially modulates the sinusoid part, this is illustrated in Figure 18.3.6.

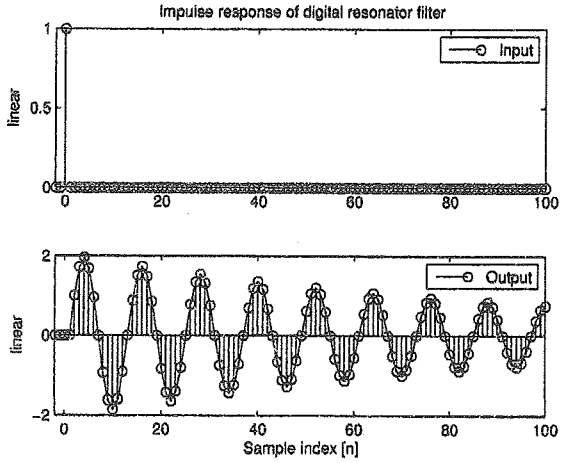


Figure 18.3.6: The impulse response of a 2^{nd} order digital resonator; note how its almost an oscillator (i.e. close to unstable)

Chapter 19

All pass filters

19.1 Introduction

As its name suggest an all-pass filter is one that passes all frequencies equally (seems strange, right?).
We will study them as they provide the basis for a very important pole swapping procedure that converts unstable IIR filters into stable ones whilst maintaining their amplitude response.

19.2 Zero-pole conjugate reciprocal pair

Theorem. *A conjugate reciprocal pole-zero pair does not affect the magnitude response of a filter apart from a constant scaling factor.*

Proof. Consider an IIR filter containing a pole at $p_1 = r.e^{j\theta}$ and a zero at the conjugate reciprocal position $z_1 = \left(\frac{1}{p_1}\right)^* = \frac{1}{r}.e^{j\theta}$, we can factorize them out as follows:

$$\begin{aligned} \tilde{H}(z) &= \frac{\tilde{N}(z)}{\tilde{D}(z)} = \frac{\tilde{N}'(z)}{\tilde{D}'(z)}\tilde{H}_1(z) \\ \text{where } \tilde{H}_1(z) &\triangleq \frac{\left(z - \frac{1}{p_1^*}\right)}{(z - p_1)} \\ &= \frac{z - \frac{1}{r}.e^{j\theta}}{z - r.e^{j\theta}} = \frac{1}{r}\frac{r.z - e^{j\theta}}{z - r.e^{j\theta}} \end{aligned}$$

This is shown in Figure 19.2.1:

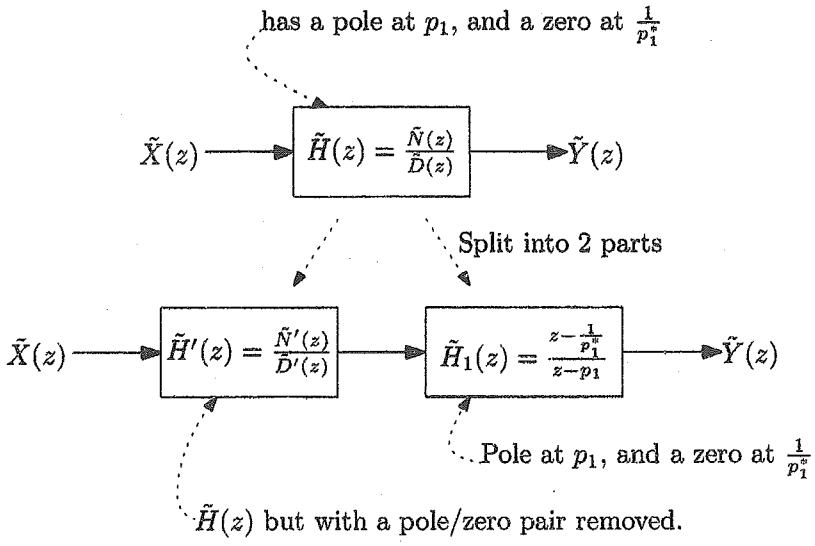


Figure 19.2.1: Illustration of how a filter having a pole / zero reciprocal can be split into two parts, one just containing said pair, and the other part having the pair removed.

The claim is that $\tilde{H}_1(z)$ has no impact on the overall *magnitude response* apart from a constant scaling factor that does not depend on ω.
The magnitude response is given by:

$$\left|\tilde{H}(j\omega)\right| = \left|\frac{\tilde{N}'(j\omega)}{\tilde{D}'(j\omega)}\right| \left|\tilde{H}_1(j\omega)\right|$$

If the term $\left|\tilde{H}_1(j\omega)\right|$ is a constant (i.e. independent of ω) then the claim must be true, so let's just prove that:

$$\begin{aligned}\left|\tilde{H}_1(j\omega)\right|^2 &= \left|\tilde{H}_1\left(e^{j\omega T}\right)\right|^2 \\ &= \cdots \text{several lines of math} \cdots \\ &= \frac{1}{|r|^2}\end{aligned}$$

i.e. a constant, not dependent on frequency ω, completing the proof □

Due to this property, a filter containing only poles and zeros that occur in conjugate reciprocal pairs is known as an all-pass filter, i.e. all frequencies are passed with equal magnitude gain (but they do experience different phase shifts).

19.3 Why is this useful?

The usefulness of the above becomes apparent after the following theorem.

Theorem. *A pole (or zero) of any filter can be replaced by a corresponding conjugate reciprocal pole (or zero) without effecting the magnitude response.*

Proof. The proof of this is actually pretty easy. Imagine a filter has a pole at p_1, then we can factorize it out as follows:

$$\tilde{H}(z) = \frac{\tilde{N}(z)}{\tilde{D}'(z)} \times \frac{1}{(z - p_1)}$$

But we already established that we can add a reciprocal pole-zero pair without affecting the amplitude response so lets add a zero at p_1 and a pole at $\left(\frac{1}{p_1}\right)^*$ to generate a new filter, $\tilde{H}_{new}(z)$, which will have the same magnitude response as the original filter $\tilde{H}(z)$:

$$\begin{aligned}
\tilde{H}_{new}(z) &= \tilde{H}(z) \frac{z - p_1}{z - \left(\frac{1}{p_1}\right)^*} \\
&= \frac{\tilde{N}(z)}{\tilde{D}'(z)} \times \frac{1}{(z - p_1)} \times \frac{z - p_1}{z - \left(\frac{1}{p_1}\right)^*} \\
&= \frac{\tilde{N}(z)}{\tilde{D}'(z)} \times \frac{1}{z - \left(\frac{1}{p_1}\right)^*}
\end{aligned}$$

But this is exactly the original filter with the pole at p_1 replaced by a pole at $\left(\frac{1}{p_1}\right)^*$
It is trivial to show the same is true for swapping zeros (but the pole version is more useful as we will now see). This completes the proof. □

19.3.1 Stabilizing IIR filters

Some IIR design techniques might produce a result containing a pole outside the unit circle which we know to be unstable. However if we are not concerned about the phase response then we can employ the above theorem to move all poles from outside the unit circle to be within it without effecting the magnitude response.
Note if a pole at $p_1 = r.e^{j\theta}$ is outside the unit circle, i.e. $r > 1$, then moving it to the conjugate reciprocal position $\left(\frac{1}{p_1}\right)^* = \frac{1}{r}e^{j\theta}$ is inside the unit circle.

Cascade with an all pass filter

The above argument can be thought of as taking an unstable filter, $\tilde{H}(z)$, and cascading it with a filter that has zeros where the unstable poles where (thereby canceling the unstable poles), and adding new poles at conjugate reciprocal position (thereby restoring the magnitude response).

The combined filter will have an unchanged magnitude response.

In practice we would never implement a separate unstable filter and follow it by an all-pass filter as this configuration may (and most likely will) experience finite precision numerical issues, instead we would always just implement the combined filter directly as show in Figure 19.3.1.

None-the-less modeling the system as the concatenation of two filters can be a useful mathematical.

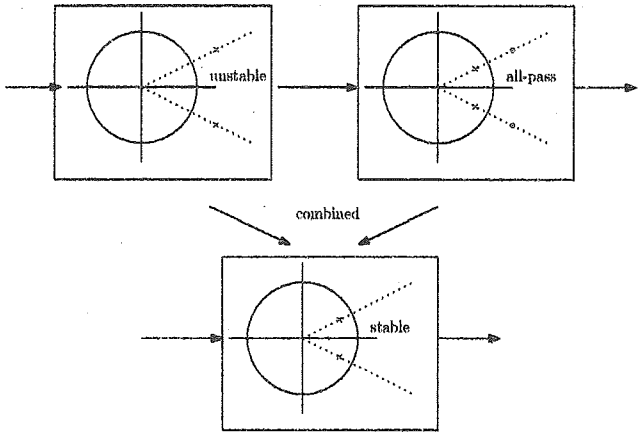


Figure 19.3.1: Illustration of how an unstable IIR filter can be converted to a stable one without changing its magnitude response by the cascading it with an all-pass filter.

19.4 Real reversed order coefficients

We already established that an all pass filter can be constructed by positioning a pole and a zero at reciprocal conjugate positions, but there is also a "time domain" view of this, i.e. there is symmetry in the filter's coefficients that can be used instead which can sometimes be easier to use.

Consider a transfer function having its numerator and denominator real valued coefficients

reversed, i.e.:

$$\tilde{H}(z) = K\frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}}{a_N + a_{N-1} z^{-1} + a_{N-2} z^{-2} + \cdots + a_0 z^{-N}}$$

Assume there is a zero at $z = z_1$, i.e.:

$$a_0 + a_1 z_1^{-1} + a_2 z_1^{-2} + \cdots + a_N z_1^{-N} = 0$$

Lets evaluate the denominator, $\tilde{D}(z)$, at the conjugate reciprocal position, i.e. at $z = \frac{1}{z_1^*}$,

$$\begin{aligned}
\tilde{D}\left(\frac{1}{z_1}\right) &= a_N + a_{N-1}\left(\frac{1}{z_1^*}\right)^{-1} + a_{N-2}\left(\frac{1}{z_1^*}\right)^{-2} + \cdots + a_0\left(\frac{1}{z_1^*}\right)^{-N} \\
&= \left[a_N + a_{N-1}\left(\frac{1}{z_1}\right)^{-1} + a_{N-2}\left(\frac{1}{z_1}\right)^{-2} + \cdots + a_0\left(\frac{1}{z_1}\right)^{-N}\right]^* \\
&= \left[z_1^N\left(a_N z_1^{-N} + a_{N-1} z_1^{-(N-1)} + a_{N-2} z_1^{-(N-2)} + \cdots + a_0\right)\right]^*
\end{aligned}$$

But the inner bracketed term is just the numerator evaluated at z_1 which is known to be zero and therefore, $\tilde{D}\left(\frac{1}{z_1}\right) = 0$.

Thus $\tilde{H}(z)$ has a pole at $z = \frac{1}{z_1}$.

What we've shown here is that a transfer function with numerator and denominator coefficients reversed contains only reciprocal pole-zero pairs and is consequently always an all-pass filter.

Appendix A

Theorems

Presented in this appendix are some useful theorems used throughout these notes.

Theorem 1. *The following equation holds:*

$$\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} = \begin{cases} N & \text{when } k \text{ is a multiple of } N \\ 0 & \text{elsewhere} \end{cases}$$

Proof. Firstly note that when $k = lN$, i.e. a multiple of N, we have:

$$\begin{aligned} \sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}}\Big|_{k=lN} &= \sum_{n=0}^{N-1} e^{+j\frac{2\pi nlN}{N}} \\ &= \sum_{n=0}^{N-1} e^{+j2\pi nl} \\ &= \sum_{n=0}^{N-1} \cos(2\pi nl) + j\sin(2\pi nl) \\ &= \sum_{n=0}^{N-1} 1 \\ &= N \end{aligned}$$

Now considering all other cases, i.e. when k is not a multiple of N. We start by noting that the summation is a geometric series of the form:

$$\sum_{n=0}^{N-1} q^n = \frac{q^N - 1}{q - 1}$$

so we have:

$$\begin{aligned}
\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} &= \sum_{n=0}^{N-1} \left(e^{+j\frac{2\pi k}{N}}\right)^n \\
&= \frac{\left(e^{+j\frac{2\pi k}{N}}\right)^N - 1}{e^{+j\frac{2\pi(k-m)}{N}} - 1} \\
&= \frac{e^{+j2\pi k} - 1}{e^{+j\frac{2\pi k}{N}} - 1} \\
&= \frac{\cos(2\pi k) + j\sin(2\pi k) - 1}{\cos\left(\frac{2\pi k}{N}\right) + j\sin\left(\frac{2\pi k}{N}\right) - 1} \\
&= \frac{0}{\cos\left(\frac{2\pi k}{N}\right) + j\sin\left(\frac{2\pi k}{N}\right) - 1}
\end{aligned}$$

We note that when k is not a multiple of N then the denominator is never zero, and so we avoid the divide-by-zero problem, and we have:

$$\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} = 0 \quad \text{when } k \text{ is a multiple of } N$$

But we already know the result for the case where $k = 0$, so putting them together we have:

$$\sum_{n=0}^{N-1} e^{+j\frac{2\pi nk}{N}} = \begin{cases} N & \text{when } k \text{ is a multiple of } N \\ 0 & \text{elsewhere} \end{cases}$$

as required. □

Theorem 2. *The variance of a zero uniform RV on* $\pm\frac{1}{2}\Delta$ *is* $\frac{\Delta^2}{12}$

Proof. First compute the non-central second moment:

$$\begin{aligned} E\left[X^2\right] &= \int_{-\infty}^{+\infty} x^2 f\left(x\right) dx \\ &= \int_{-\Delta/2}^{+\Delta/2} \frac{x^2}{\Delta} dx = \frac{1}{\Delta}\left.\frac{x^3}{3}\right|_{-\Delta/2}^{+\Delta/2} \\ &= \frac{1}{3\Delta}\left[\left(\Delta/2\right)^3 - \left(-\Delta/2\right)^3\right] \\ &= \frac{\Delta^2}{12} \end{aligned}$$

The variance is given by:

$$\begin{aligned} Var\left[X\right] &= E\left[X^2\right] - \left(E\left[X\right]\right)^2 \\ &= \frac{\Delta^2}{12} \end{aligned}$$

because the mean of a symmetric RV is zero. □

Appendix B

Complex linear phase filters.

In Chapter 14 some conditions on the real valued FIR coefficients $\{b_k\}$ to ensure linear phase operation were derived.
Question: Is there a more general framework for complex $\{b_k\}$?
Answer: Yes there is!

We claim that $b_k = e^{-j2\theta} b^*_{M-k}$ will work for odd and even FIR filters and all of the real valued cases in Chapter 14 can be considered as a special cases of this more encompassing rule.

B.1 An even number of coefficients

Assume M is odd, i.e. an even number of coefficients, we can start with expression 14.2.1 for $\angle \sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}$ and

$$= \angle \left(\sum_{k=0}^{\frac{M-1}{2}} \left(b_k e^{-j\left(k-\frac{M}{2}\right)\omega T} + b_{M-k} e^{j\left(k-\frac{M}{2}\right)\omega T} \right) \right)$$

Let $b_k = e^{-j2\theta} b^*_{M-k}$, and take $e^{-j\theta}$ out of the brackets:

$$\begin{aligned}
&= \angle \left(\sum_{k=0}^{\frac{M-1}{2}} \left(e^{-j2\theta} b^*_{M-k} e^{-j\left(k-\frac{M}{2}\right)\omega T} + b_{M-k} e^{j\left(k-\frac{M}{2}\right)\omega T} \right) \right) \\
&= \angle \left(e^{-j\theta} \sum_{k=0}^{\frac{M-1}{2}} \left(e^{-j\theta} b^*_{M-k} e^{-j\left(k-\frac{M}{2}\right)\omega T} + e^{j\theta} b_{M-k} e^{j\left(k-\frac{M}{2}\right)\omega T} \right) \right) \\
&= \angle \left(e^{-j\theta} \sum_{k=0}^{\frac{M-1}{2}} \Re \left(e^{-j\theta} b^*_{M-k} e^{-j\left(k-\frac{M}{2}\right)\omega T} \right) \right)
\end{aligned}$$

As the summation is real valued, the angle is simply the angle of $e^{-j\theta}$, i.e. $-\theta$, therefore we have:

$$\angle \bar{H}(j\omega) = -\frac{M}{2}\omega T - \theta$$

And the group delay is $\tau = \frac{M}{2}T$ as before.

B.1.1 An odd number of coefficients

Assume odd number of coefficients and following the same procedure as before we end up with some very similar expressions but this time the middle term is present. i.e. the expression foe $\angle \sum_{k=0}^{M} b_k e^{-j\left(k-\frac{M}{2}\right)\omega T}$ becomes:

$$= \angle\left(e^{-j\theta}\left(b_{\frac{M}{2}}e^{j\theta} + \sum_{k=0}^{\frac{M}{2}-1}\Re\left(e^{-j\theta}b^*_{M-k}e^{-j\left(k-\frac{M}{2}\right)\omega T}\right)\right)\right)$$

If we place the restriction that $\angle b_{\frac{M}{2}} = -\theta$ then everything (except the $e^{-j\theta}$ on the outside) becomes real valued, meaning that the angle is $-\theta$ and so we get:

$$\angle \bar{H}(j\omega) = -\frac{M}{2}\omega T - \theta$$

as before.

One further note is to say that the restriction that $\angle b_{\frac{M}{2}} = -\theta$ is already embodied in the statement that $b_k = e^{-j2\theta}b^*_{M-k}$ as can be seem by letting $k = \frac{M}{2}$ and as such does not represent any additional restriction.

Figure B.1.1 is an illustration showing the 4 examples we had earlier but now presented with complex coefficients. Note the real and imaginary restriction of the middle coefficient for symmetric and anti-symmetric modes respectively.

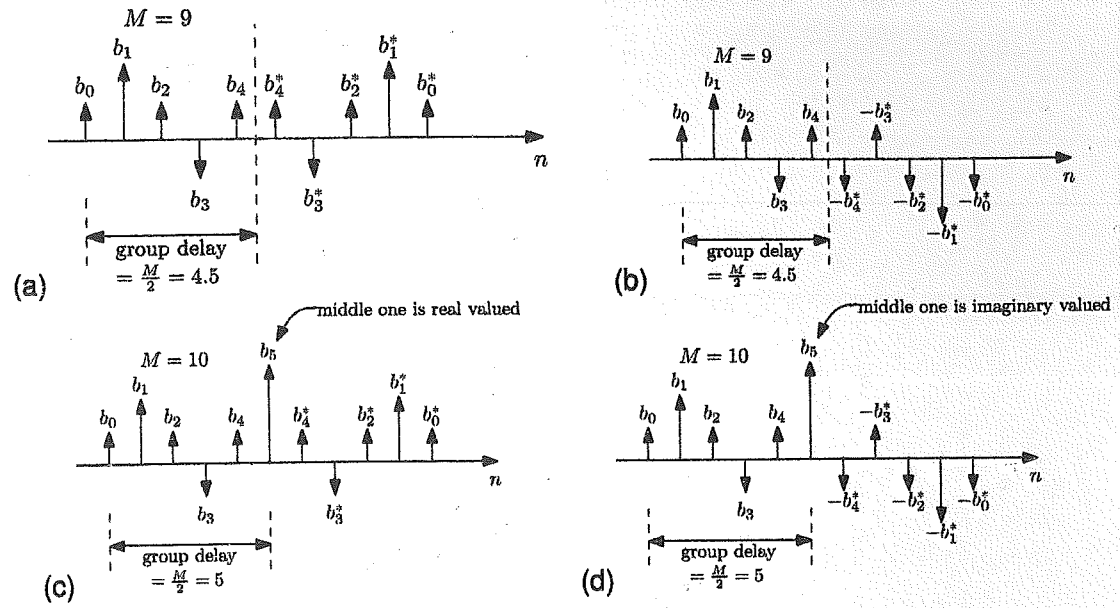


Figure B.1.1: Example complex valued linear phase FIR filters:
(a) even & conjugate symmetry: $b_k = b^*_{M-k}$
(b) even & conjugate anti-symmetry: : $b_k = -b^*_{M-k}$
(c) odd & symmetry: : $b_k = b^*_{M-k}$
(b) odd & conjugate anti-symmetry: : $b_k = -b^*_{M-k}$

数字信号处理

印　　刷：北京工业大学后勤文印部

印刷时间：2018年2月

印刷数量：89本

工 本 费：23.50元