Vector Spaces

- A **vector space** (or **linear space**) S is a collection of objects called **vectors** $\boldsymbol{v}$. We would then say:

$$\boldsymbol{v} \in S$$

- We assume these vectors follow the laws of associativity and commutativity. These vectors are **closed under addition** and **multiplication:**

> **Closure Under Addition**: For vectors $\boldsymbol{v_1}, \boldsymbol{v_2} \in S$, we say that:
>
> $$\boldsymbol{v_1} + \boldsymbol{v_2} \in \boldsymbol{S}$$
>
> **Closure Under Scalar Multiplication**: For some vector $\boldsymbol{v} \in S$ and some scalar $\alpha \in \mathbb{R}$:
>
> $$\alpha \boldsymbol{v} \in S$$

- The **additive identity** states that the sum of a vector $\boldsymbol{v} \in S$ and the zero vector produces the same vector, $\boldsymbol{v}$. That is: $\boldsymbol{v} + \boldsymbol{0} = \boldsymbol{v}$

Inner Product Spaces

- An **inner product space** is an extension of the linear (vector) space.
- Includes a structure called the **inner product** that maps two vectors in S to some field of scalars F. The operation can be defined as:

$$\langle \cdot, \cdot \rangle : S \times S \to F$$

Suppose some $\boldsymbol{v}, \boldsymbol{w} \in S$. Then, their inner product produces a real scalar i.e. $\langle \boldsymbol{v}, \boldsymbol{w} \rangle \in \mathbb{R}$

- If two vectors are **real**, then their inner product is simply the **dot product**:

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{v} = \sum_{i=1}^{n} v_i w_i$$

- If the two vectors are defined on $\mathbb{C}$, then we take the Hermitian (conjugate transpose) of the second vector in the inner product:

$$\langle v, w \rangle = w^H v = \sum_{i=1}^{n} v_i \overline{w_i}$$

Properties of the Inner Product

1) Symmetry: $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \langle \boldsymbol{w}, \boldsymbol{v} \rangle$

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \sum_{i=1}^{n} v_i \, w_i = \sum_{i=1}^{n} w_i \, v_i = \langle \mathbf{w}, \mathbf{v} \rangle$$

2) Linearity (1): $\langle \alpha \boldsymbol{v}, \boldsymbol{w} \rangle = \alpha \langle \boldsymbol{w}, \boldsymbol{v} \rangle \implies \langle \boldsymbol{v}, \alpha \boldsymbol{w} \rangle = \alpha \langle \boldsymbol{w}, \boldsymbol{v} \rangle \quad (by\ symmetry)$

$$\langle \alpha \boldsymbol{v}, \boldsymbol{w} \rangle = \sum_{i=1}^{n} (\alpha v_i) \, w_i = \alpha \sum_{i=1}^{n} v_i \, w_i = \alpha \langle \mathbf{w}, \mathbf{v} \rangle$$

However, if $\alpha \in \mathbb{C}$, then we have:

$$\langle \boldsymbol{v}, \alpha \boldsymbol{w} \rangle = \sum_{i} v_i \overline{(\alpha w_i)} = \overline{\alpha} \sum_{i} v_i w_i = \overline{\alpha} \langle \boldsymbol{v}, \boldsymbol{w} \rangle$$

3) **Linearity (2):** $\langle \boldsymbol{v} + \boldsymbol{u}, \boldsymbol{w} \rangle = \langle \boldsymbol{v}, \boldsymbol{u} \rangle + \langle \boldsymbol{u}, \boldsymbol{w} \rangle \implies \langle \boldsymbol{v}, \boldsymbol{w} + \boldsymbol{u} \rangle = \langle \boldsymbol{v}, \boldsymbol{w} \rangle + \langle \boldsymbol{v}, \boldsymbol{u} \rangle$

$$\langle \boldsymbol{v} + \boldsymbol{u}, \boldsymbol{w} \rangle = \sum_{i=1}^{n} (v_i + u_i) w_i = \sum_{i=1}^{n} v_i w_i + \sum_{i=1}^{n} u_i w_i = \langle \boldsymbol{v}, \boldsymbol{w} \rangle + \langle \boldsymbol{u}, \mathbf{w} \rangle$$

4) **Cauchy-Schwarz Inequality**

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle \leq \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle \cdot \langle \boldsymbol{w}, \boldsymbol{w} \rangle} \quad OR \quad \langle \boldsymbol{v}, \boldsymbol{w} \rangle \leq \|\boldsymbol{v}\| \cdot \|\boldsymbol{w}\|$$

Other Important Properties

- The **norm** of a vector computes the **length** of a vector. That is:

$$\|\boldsymbol{v}\| = \langle \boldsymbol{v}, \boldsymbol{v} \rangle$$

- Since the inner product is effectively a dot-product between two vectors, the ***angle*** (θ) between two vectors can be defined as:

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = |\boldsymbol{v}||\boldsymbol{w}| \cdot cos\theta$$

- The **triangle inequality** states that:

$$\|\boldsymbol{v} + \boldsymbol{w}\| \leq \|\boldsymbol{v}\| + \|\boldsymbol{w}\|$$

- Some vector $\boldsymbol{v} \in S$ can be defined by an **orthonormal set** is a set of vectors $\{\boldsymbol{\psi}_i\} \in S$ that form a **basis** for S. That is, we can express $\boldsymbol{v}$ as a linear combination of $\boldsymbol{\psi}_i$ vectors using scalars $\boldsymbol{a_i}$ that assign a weight to each basis vector:

$$\boldsymbol{v} = \sum_{\boldsymbol{i}} a_i \psi_i$$

  For the orthonormal set to a be a basis for a vector space, all elements of the set must be **linearly independent** and must be orthogonal (perpendicular) to one another. That is:

$$\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle = 0 \quad for\ i \neq j$$
$$\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle = 1 \quad for\ i = j$$

- Each basis vector is essentially produced by performing a vector **projection** of $\boldsymbol{v}$ onto each basis vector. A way of expressed this projection mathematically would be:

$$proj_{\boldsymbol{\psi}_i} \boldsymbol{v} = \langle \boldsymbol{v}, \boldsymbol{\psi}_i \rangle \boldsymbol{\psi}_i$$

  The inner product $\langle \boldsymbol{v}, \boldsymbol{\psi}_i \rangle \in \mathbb{R}$ serves to scale the basis vector (that points in a certain direction) by a certain amount!

Signal Spaces

- A *signal* is a time-varying quantity that carries information (e.g. voltage, current, stock market fluctuations, oscillatory motion etc.)
- A signal can be treated as a mathematical object subject to the same conditions as linear spaces.

Discrete Signals (Sequences)

- A **discrete** signal is a finite sequence of values and is effectively a vector itself

$$\boldsymbol{x} = (x_0, x_1, x_2, x_3 \dots) \quad for\ \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$$

- A **square-summable sequence**, $\boldsymbol{x}$ is such that:

$$\sum_{i=1}^{\infty} |x_i|^2 < \infty$$

- A sequence can be defined by an orthonormal basis of singleton (unit impulse) sequences $\{\boldsymbol{\delta}_i\} \in \mathbb{R}^n$ where: $\boldsymbol{\delta}_i = (\dots\ 0\ 0\ 0\ 1\ 0\ 0\ 0 \dots)$

Continuous Signals (Functions)

- The only difference here is that these signals are differentiable everywhere.
- Define a set of functions $\boldsymbol{f} = f(x)$. $\boldsymbol{f}$ forms a vector space if define their closure under addition and scalar multiplication **pointwise:**

$$\begin{aligned} \boldsymbol{f} + \boldsymbol{g} = \boldsymbol{h} \quad &\Leftrightarrow \quad f(x) + g(x) = h(x) \quad \forall x \\ \alpha \boldsymbol{f} = h \quad &\Leftrightarrow \quad \alpha f(x) = h(x) \quad \forall x \end{aligned}$$

- Such functions, like discrete sequences need to be square-integrable
- The inner product between two continuous signals is defined using the **integral** instead

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \int f(x) g(x) dx$$

However, if either $\boldsymbol{f}$ and $\boldsymbol{g}$ are in $\mathbb{C}$, then, we need to take the Hermitian of one function:

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \int f(x) g^H(x) dx$$

Linear, Time-Invariant Systems and Operators

- A **system** performs some of mathematical mapping (or *filtering*) of a vector, $\boldsymbol{v} \in S$ to an output vector $\boldsymbol{v}' \in S'$ where S' is the vector space after the transformation

- If the system is **linear**, then usually $S = S'$

- If H defines the function that performs the transformation on some vector, then the properties of linearity and closure under addition and scalar multiplication apply:

$$H(\boldsymbol{v} + \alpha \boldsymbol{w}) = H(\boldsymbol{v}) + \alpha H(\boldsymbol{w}) \qquad \forall \boldsymbol{v}, \boldsymbol{w} \in S, \alpha \in \mathbb{R}$$

- The transformation is also technically a **linear transformation**. As such, H can be defined via a matrix and can be dot multiplied with the vector to be transformed

$$\boldsymbol{y} = H(\boldsymbol{x}) = \boldsymbol{H}^* \cdot \boldsymbol{x}$$

Where $*$ denotes the Hermitian (to avoid confusion)

Alternatively, we say that the i[th] element of $\boldsymbol{y}$ (y_i) is produced by dot multiplying the i[th] element of $\boldsymbol{x}$ (x_i) and the i[th] column of the system matrix ($\boldsymbol{h}_i$). That is:

$$\boldsymbol{y} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} v_i \cdot \boldsymbol{h}_i = \langle \boldsymbol{v}, \boldsymbol{h_i} \rangle$$

In the above, the filtered output is the linear combination of each system response (column vector) multiplied with an element of the input sequence (acting as a scalar).

Discrete LTI Systems

- **Input-Based Filtering**: Essentially computing the linear combination of the filter's impulse response scaled by each element in the input sequence and delayed by i samples. Implementation-wise, $\boldsymbol{x}$ must be padded to match the length of $\boldsymbol{y}$.

$$\boldsymbol{y} = H(\boldsymbol{x}) = \sum_i x[i] \cdot \boldsymbol{h}_i = \sum_k x[k] h[n-k] = (h \star x)[n]$$

- **Output-Based Filtering:** An element in the output vector is computed by computing the inner product between the input sequence and the time-reversed version of the filter's impulse response.

$$y[k] = \sum_i x[i] h[k-i] = \sum_i x[i] \tilde{h}\,[i-k] = \langle \boldsymbol{x}, \tilde{\boldsymbol{h}}_k \rangle$$

Verifying If A System Is LTI

Suppose a system H that performs a transformation on an input sequence, $\boldsymbol{x}$ and produces an output sequence $\boldsymbol{y}$. To determine if H is LTI, we perform the following tests:

1) **Test for Linearity**: For some scalar $\alpha \in \mathbb{R}$, and the input sequence, the output sequence must ALSO contain the scalar factor:

$$\boldsymbol{x}_0 = \alpha \boldsymbol{x} \quad \Longrightarrow \quad \boldsymbol{y}_0 = \alpha \boldsymbol{y}$$

2) **Test for Time-Invariance**: A time-shift imposed in the input sequence must induce an identical time-shift in the output sequence.

$$x_o = x[n - n_o] \quad \Longrightarrow \quad y_0 = y[n - n_0]$$

Example Questions

Determine if the following systems are LTI:

1) Where $x[n]$ is the input sequence, and:

$y[n] = x^2[n]$

2) $g = H(f)$ where f and g are continuous time signals and

$$g(t) = \frac{\partial f}{\partial t}(t)$$

3) $y = H(x)$ where y and x are discrete time signals and:

$$y[n] = \sum_{k=0}^{K} h[k]x[2n - k]$$

Solutions

1) **Test for linearity:** Suppose we have an input $x_o[n] = \alpha x[n]$ for some $\alpha \in \mathbb{R}$. We want to show that this will produce an output $y_o[n] = \alpha y[n]$.

$$y_o[n] = x_o^2[n] = (\alpha x[n])^2 = \alpha^2 x^2[n] = \alpha^2 y[n] \neq \alpha y[n]$$

Hence, the system is **non-linear** (which makes sense since the system squares the signal)

Test for time-invariance: Now suppose our input has been time-shifted. That is, for some $n_o \in \mathbb{R}$, we have $x_o[n] = x[n - n_o] \Longrightarrow y_o[n] = y[n - n_o]$.

$$y_o[n] = x_o^2[n] = (x[n - n_0])^2$$
$$y[n - n_0] = (x[n - n_0])^2$$

Hence, we can see that $y_o[n] = y[n - n_o]$ when a shift is applied to the input sequence. The system can be said to then be **time-invariant.**

2) **Test for linearity:** Suppose some input $f_o(t) = \alpha f(t)$ for some $\alpha \in \mathbb{R}$, then, we should expect the output to be in the form $g_o(t) = \alpha g(t)$

$$g_o(t) = \frac{\partial f_o(t)}{\partial t} = \frac{\partial}{\partial t}(\alpha f(t)) = \alpha \cdot \frac{\partial f}{\partial t} = \alpha g(t)$$

Therefore, the system is **linear**.

Test for time-variance: Suppose some shift to the input sequence such that out input is now $f_o(t) = f(t - t_0)$. Then, we expect the output to exhibit an identical time-shift. That is, we have $g_o(t) = g(t - t_0)$

$$g_o(t) = \frac{\partial f_o(t)}{\partial t} = \frac{\partial f(t - t_0)}{\partial t} = \lim_{h \to 0} \frac{f(t - t_0 + h) - f(t - t_0)}{h}$$

Now examine the output when performing $g(t - t_0)$:

$$g(t - t_0) = \frac{\partial f(t - t_0)}{\partial t} = \lim_{h \to 0} \frac{f(t - t_0 + h) - f(t - t_0)}{h}$$

The same result appears, and we can see that $g_o(t) = g(t - t_0)$. Hence the system is also **time-invariant.**

3) **Test for linearity:** Suppose we have an input $x_o[n] = \alpha x[n]$ for some $\alpha \in \mathbb{R}$. We want to show that this will produce an output $y_o[n] = \alpha y[n]$.

$$y[n] = \sum_{k=0}^{K} h[k] x_0[2n - k] = \sum_{k=0}^{K} h[k] \cdot \alpha x[2n - k] = \alpha y[n] \quad (\therefore linear)$$

Test for time-invariance: Now suppose our input has been time-shifted. That is, for some $n_o \in \mathbb{R}$, we have $x_o[n] = x[n - n_o] \Longrightarrow y_o[n] = y[n - n_o]$.

$$y_0[n] = \sum_{k=0}^{K} h[k] x_0[2n - k] = y[n] = \sum_{k=0}^{K} h[k] x[2n - k - n_0]$$

$$y[n - n_0] = \sum_{k=0}^{K} h[k] x[2(n - n_0) - k] = \sum_{k=0}^{K} h[k] x[2n - k - 2n_0]$$

We can clearly see that $y_o[n] \neq y[n - n_0]$. Hence, the system is **NOT time-invariant**.

Sampling & Aliasing

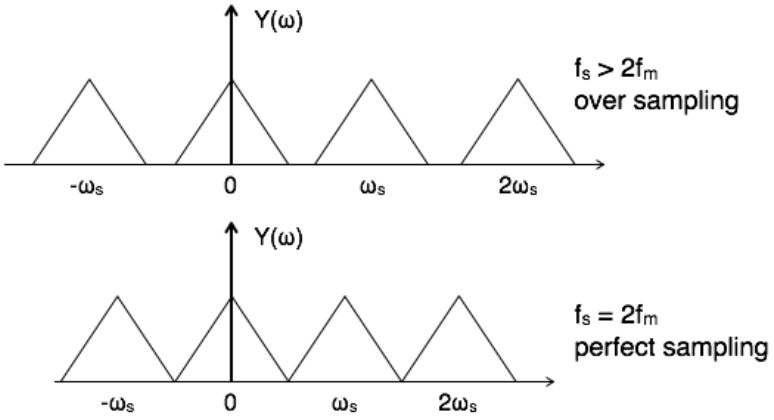
- Using **unit sampling** via the direct delta function $\boldsymbol{\delta}$ at integer multiples of a sampling period, $t = nT$ for all $n \in \mathbb{Z}$

$$x[n] = f(t)|_{t=nT} = f(nT)$$

Where $f(t)$ is a continuous signal with a bandlimited CTFT $\hat{f}(\omega) \neq 0$ for $|\omega| < \pi$ and $x[n]$ is the sampled/discrete version with DTFT $\hat{x}(\omega)$.

- The sampling frequency is $F_s = \frac{1}{T}$

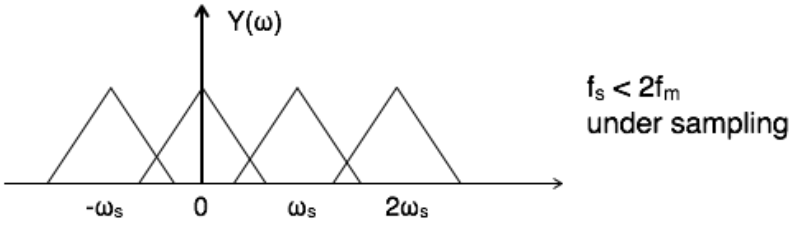
- In the frequency domain, sampling takes the original, bandlimited spectrum $\hat{f}(\omega)$ and replicates it at every integer multiple of F_s



- To ensure the replicated spectra do **not** overlap (aliasing), the sampling frequency must be greater than **twice the maximum** frequency in the CTFT $\hat{f}(\omega)$ which we denote as f_m.

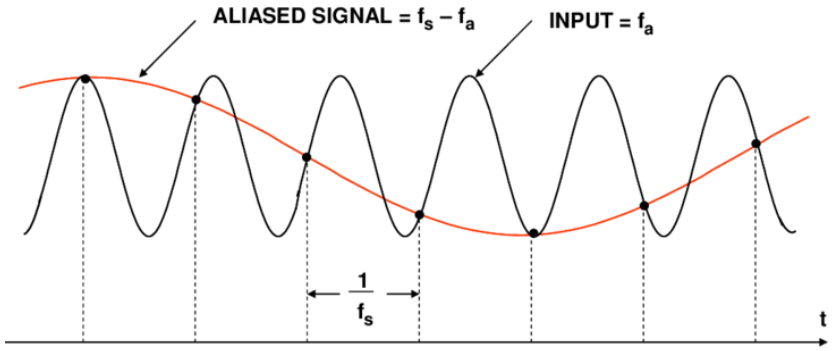
$$F_s \geq 2f_m \quad (Nyquist\ sampling\ theorem)$$

Frequency-domain aliasing will occur when the sampling rate does not meet this condition.



- Time-domain aliasing is the result of **under-sampling** in the time-domain (i.e insufficient zero-padding)

- Under-sampling results in the incorrect representation of the original signal. This is shown in the following picture.



NOTE: f_a IS SLIGHTLY LESS THAN f_s

Reconstruction (Interpolation)

- The continuous and discrete versions of a signal are connected via their CTFT and DTFT

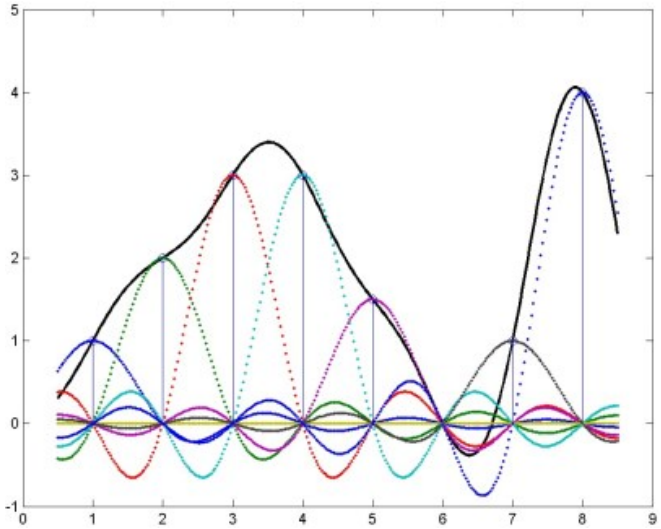
$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{x}(\omega)e^{jn\omega}d\omega = x[n] = f(t)|_{(t=nT)} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{jn\omega}d\omega$$

- The above relationship requires that $\hat{x}(\omega) = \hat{f}(\omega)$.

- The process of recovering $f(t)$ from $x[n]$ is called **interpolation**. This process computes the signal values in *between* the discrete samples via some continuous **interpolation kernel**. The interpolation kernel appears when we attempt to connect the DTFT and CTFT:

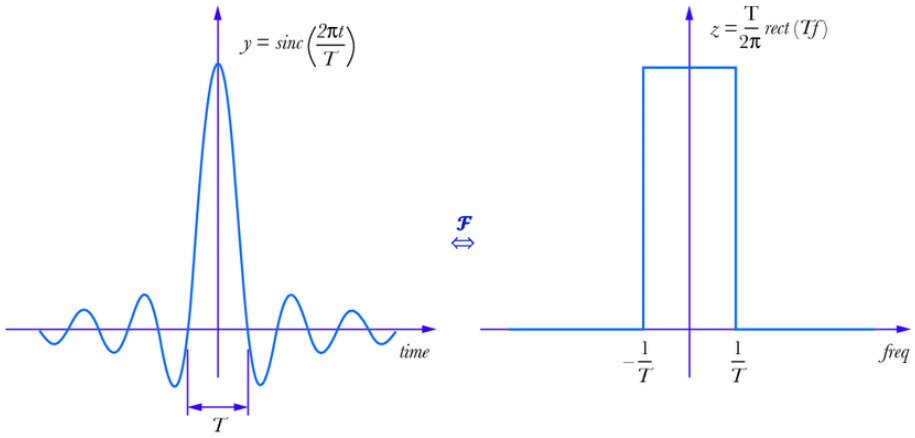
$$\begin{aligned}
f(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\omega)e^{j\omega t}d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{x}(\omega)e^{j\omega t}d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\sum_{n=-\infty}^{\infty}x[n]e^{-j\omega n}\right)e^{jn\omega}d\omega \\
&= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}x[n]\int_{-\infty}^{\infty}e^{j\omega t}\,e^{-j\omega n}\,d\omega \\
&= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}x[n]\int_{-\infty}^{\infty}e^{j(t-n)\omega}\,d\omega \\
&= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}x[n]sinc(t-n) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}x[n]q(t-n)
\end{aligned}$$

- The continuous signal is reconstructed using shifted *sinc* functions weighted by the amplitude of each sample in the discrete input sequence, $x[n]$. The sinc is in fact the interpolation kernel.



- It is **crucial** however, that the interpolation kernel (sinc or related) **is bandlimited**. That is the frequency response of the kernel $\hat{q}(\omega)$ must be zero for all $|\omega| \geq \pi$.

- For the sinc interpolation kernel specifically, its frequency response is a rectangular window.



If the kernel is NOT bandlimited in $\omega \in (-\pi, \pi)$, there is a chance that it will capture the replicated spectra of the discrete signal during the reconstruction process.

- The sum for the interpolation process can be re-written in vector notation:

$$\boldsymbol{f} = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} x[n]\boldsymbol{\psi_n}$$

Where $\{\boldsymbol{\psi}_n\} = \{sinc_n(t)\}$ is an orthonormal basis comprising of sinc functions that can be used to represent the function $\boldsymbol{f}$ as a linear combination of these sinc functions. i.e. $\boldsymbol{f}$ can be expressed as the orthonormal expansion of $x(t)$.

- Parseval's theorem holds for both the discrete and reconstructed signal. The relationship is:

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt$$

The FS, FT, DTFT and DFT

Fourier Series (Sines & Cosine Basis)

- Any periodic function $f(t)$ with some period T_0 (or frequency $f_0 = 1/T_0$) can be expressed via a sum of sines and cosines. The series can be expressed as:

$$f(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} [A_k \cos(\omega_0 kt) + B_k \sin(\omega_0 kt)]$$

  Where $\omega_0 = 2\pi f_0$. Higher values of k correspond to sines/cosines of higher frequencies. The $A_0/2$ is known as the DC term as it is NOT associated with an oscillatory component.

- The Fourier coefficients A_k and B_k act as weights for the $\cos(kt)$ and $\sin(kt)$ terms. These coefficients are expressed as:

$$A_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(\omega_0 kt)dt = \frac{1}{\| \cos(\omega_0 kt) \|} \cdot \langle f(t), \cos(\omega_0 kt)\rangle$$
$$B_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(\omega_0 t)\,dt = \frac{1}{\| \sin(\omega_0 kt) \|} \cdot \langle f(t), \sin(\omega_0 kt)\rangle$$

  Where the division of the inner products $\| \cos(kt) \|$ and $\| \sin(kt) \|$ normalises the coefficients. We can compute their norms by first finding the squared-norm which is the inner-product of each basis function on itself and then square-rooting the result:

$$\begin{aligned}\|\cos(\omega_0 kt)\|^2 &= \langle \cos(\omega_0 kt), \cos(\omega_0 kt)\rangle \\ &= \int_{-\pi}^{\pi} \cos^2(\omega_0 kt)\,dt \\ &= \frac{1}{2}\int_{-\pi}^{\pi} (1 + \cos(2\omega_0 kt))dt \\ &= \frac{1}{2}\left[t + \frac{\cos(2\omega_0 kt)}{2\omega_0 k}\right]_{t=-\pi}^{t=\pi} \\ &= \frac{1}{2}[(\pi + 0) - (\pi + 0)] \\ &= \pi = \| \sin(\omega_0 kt)\|^2\end{aligned}$$

  We conclude then that:

$$A_k = \frac{1}{\sqrt{\pi}} \cdot \langle f(t), \cos(\omega_0 kt)\rangle \qquad B_k = \frac{1}{\sqrt{\pi}} \cdot \langle f(t), \sin(\omega_0 kt)\rangle$$

- **Geometrically** the Fourier series is constructed by first considering $\sin(\omega_0 kt)$ and $\cos(\omega_0 kt)$ as orthogonal basis functions (like basis vectors) for the Fourier space.
- We then compute the Fourier coefficients by **projecting** $f(t)$ onto the basis functions
- This coefficient then acts as a weight for its relevant basis function ($\sin(\omega_0 kt)$ or $\cos(\omega_0 kt)$) at a particular frequency, k in the infinite sum.

Fourier Series (Complex Sinusoid Basis)

- Since sine and cosine waves are connected via Euler's formula in $\mathbb{C}$, we can express $f(t)$ in terms of an orthonormal basis of ***complex sinusoids*** in the following way:

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$$

Whereby Euler's Formula: $e^{jk\omega_0 t} = \cos(k\omega_0 t) + jsin(k\omega_0 t) := \psi_k$.

- The coefficients $c_k \in \mathbb{C}$ are produced by now taking the inner product of the function $f(t)$ with the complex sinusoid basis function/vector. That is:

$$c_k = \frac{1}{\|\psi_k\|} \cdot \langle f(t), \psi_k \rangle = \frac{1}{\|\psi_k\|} \cdot \int_{-\pi}^{\pi} f(t)\overline{\psi_k} dt$$

Where $\|\psi_k\|$ is the norm of the complex sine. We can compute what this is by taking the inner product of the complex sine with itself (just like the previous section). Because we are in $\mathbb{C}$, we need to use the general definition for the inner product:

$$\langle f(x), g(x) \rangle = \int f(x) \cdot \overline{g(x)}\, dx$$

We compute the squared-norm of the complex sine basis function to be:

$$\begin{aligned} \|\psi_k\|^2 &= \langle \psi_k, \psi_k \rangle \\ &= \int_{-\pi}^{\pi} \psi_k \cdot \overline{\psi_k} \cdot dt \\ &= \int_{-\pi}^{\pi} e^{jkt} e^{-jkt} dt \\ &= \int_{-\pi}^{\pi} dt \\ &= [t]_{t=-\pi}^{t=\pi} \\ &= 2\pi \end{aligned}$$

We can then say that the normalised complex coefficients can be computed as:

$$c_k = \frac{1}{\sqrt{2\pi}} \cdot \langle f(t), \psi_k \rangle$$

- Because of this, the family of orthonormal complex sinusoids $\{\psi_k\}$ comprises of components that can be expressed as:

$$\psi_k(t) = \frac{1}{\sqrt{2\pi}} \cdot e^{jk\omega_0 t}$$

Dirichlet Conditions

A periodic function can be expressed as a Fourier Series if it meets the following ***Dirichlet*** conditions

1) $x(t)$ must be absolute integrable over a single period i.e. $\int_0^{T_0}|x(t)|dt < \infty$
2) $x(t)$ has a finite number of extrema within each period.
3) $x(t)$ has at most a finite number of discontinuities within each period.

DTFT and Connection to Fourier Series

- The *Discrete-Time Fourier Transform* maps a discrete sequence, $x[n]$ to the continuous Fourier Space to produce the discrete sequences *frequency spectrum* $\hat{x}(\omega)$. The DTFT pair is given by:

$$\underbrace{\hat{x}(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}}_{DTFT} \quad \overleftrightarrow{\mathcal{F}/\mathcal{F}^{-1}} \quad \underbrace{x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{x}(\omega)e^{j\omega n}d\omega}_{Inverse\ DTFT}$$

- The connection between the DTFT and the Fourier Series can be seen by recalling the Fourier series for $2L$ periodic functions. That is $\omega = \frac{\pi k}{L}$. We can also say that $\Delta\omega = \frac{\pi}{L}$. Then:

$$x[n] = \sum_{k=-\infty}^{\infty} c_k e^{j\frac{k\pi}{L}t} = \sum_{k=-\infty}^{\infty} c_k e^{j\mathrm{k}\Delta\omega t}$$

We know that the complex coefficients are produced by the inner product of the function/sequence and the complex sinusoid basis:

$$c_k = \frac{1}{2L}\sum_{n=-L}^{L} x[n]e^{-jn\Delta\omega}$$

This means that the discrete Fourier Series for $x[n]$ can be written as:

$$x[n] = \sum_{k=-\infty}^{\infty}\left[\frac{\Delta\omega}{2\pi}\sum_{n=-\frac{\pi}{\Delta\omega}}^{\frac{\pi}{\Delta\omega}} x[n]e^{-jn\Delta\omega}\right]\cdot e^{j\mathrm{k}\Delta\omega t}$$

If we let $\Delta\omega \to 0$, the outer sum becomes Riemann integrable with differential $\Delta\omega \to d\omega$:

$$x[n] = \frac{1}{2\pi}\int_{-\infty}^{\infty}\underbrace{\left[\sum_{n=-\infty}^{\infty} x[n]e^{-jn\omega}\right]}_{\hat{x}(\omega) \to\ DTFT\ of\ x[n]}\cdot e^{j\omega t}\,d\omega$$

Where $\hat{x}(\omega)$ is the DTFT of $x[n]$ which is the projection of $x[n]$ onto the basis functions.

CTFT (Fourier Transform)

The CTFT maps a finite energy continuous signal, $f(t)$ to the frequency domain in the same exact way that is done in the DTFT scenario.

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \qquad \overleftrightarrow{\mathcal{F}/\mathcal{F}^{-1}} \qquad f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)e^{j\omega n}d\omega$$

Important Properties of the CTFT

Property	Operation
Conjugate Symmetry	$\hat{f}(\omega) = \overline{\hat{f}(-\omega)}$
Time-Shifting	$f_0(t) = f(t - t_0) \implies \hat{f}_0(\omega) = e^{-j\omega t_0}\,\hat{f}(\omega)$
Convolution	$(f \star g)(t) \implies \hat{f}(\omega) \cdot \hat{g}(\omega)$
General Modulation	$g(t) = f(t) \cdot m(t) \implies \hat{g}(\omega) = \frac{1}{2\pi}\left(\hat{f}(\omega) \star \hat{m}(\omega)\right)$
Parseval's Theorem	$\int_{-\infty}^{\infty} \lvert f(t)\rvert^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} \left\lvert\hat{f}(\omega)\right\rvert^2 d\omega$
Differentiation	$g(t) = \frac{d}{dt}h(t) \implies \hat{g}(\omega) = j\omega \cdot \hat{h}(\omega)$

Important Examples of CTFT To Remember (Memorise)

- **Rectangular Pulse (Time – Domain)**: The rectangular pulse is defined as:

$$\Pi(t) = \begin{cases} 1 & |t| < 0.5 \\ 0 & |t| \geq 0.5 \end{cases}$$

The FT of $\Pi(t)$ is the sinc function with zeros at every 2π. This is:

$$\widehat{\Pi}(\omega) = sinc\left(\frac{\omega}{2\pi}\right)$$

- **Rectangle Pulse (Fourier – Domain):** The rectangular pulse in the frequency-domain in the bandlimited interval $\omega \in (-\pi, \pi)$ is

$$\widehat{\Pi}(\omega) = \begin{cases} 1 & |\omega| < \pi \\ 0 & |\omega| \geq \pi \end{cases}$$

The inverse DTFT of this would be the time-domain sinc function. We can see the **duality** between the sinc and rectangular pulse in the time/frequency domain.

$$x(t) = \frac{\sin(\pi t)}{\pi t}$$

- **Triangular Pulse:** The triangular pulse in the time-domain is defined by the following:

$$\Lambda(t) = \begin{cases} 1 - |t| & if \quad |t| < 1 \\ 0 & if \quad |t| \geq 1 \end{cases}$$

The Fourier transform of the $\widehat{\Lambda}(\omega)$ is given by:

$$\widehat{\Lambda}(\omega) = sinc^2\left(\frac{\omega}{2\pi}\right)$$

Convolution & Polynomial Multiplication

- Convolution is an operation between two functions x and h that produces an output that expresses how the shape of one function is modified by the other.

- Convolution is an LTI operation and must be a causal operation.

- Mathematically, the convolution in continuous time can be expressed as:

$$(x \star h)(t) = \int_{-\infty}^{\infty} x(\tau) \underbrace{h(t-\tau)}_{\substack{time-rev. \\ and \\ shifted}} d\tau$$

  The initial time reversal ensures causality – effects of future outputs are caused by inputs in the past.

- In discrete time, this would be the same by using an infinite sum:

$$\boldsymbol{y} = (x \star h)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[k]\boldsymbol{h}_n = \langle \boldsymbol{x}, \boldsymbol{h}_n \rangle$$

  We may think of the above operation as producing a whole output vector, $\boldsymbol{y}$ as a linear combination of the system response $\boldsymbol{h}$ shifted by n and weighed by an element of the input sequence, $\boldsymbol{x}$.

- If we let $i = n - r$, then the convolution becomes:

$$\boldsymbol{y} = \sum_{r=-\infty}^{\infty} h[r]x[n-r] = \sum_{r=-\infty}^{\infty} h[r]\boldsymbol{x_n} = \langle \boldsymbol{h}, \boldsymbol{x}_n \rangle \qquad (\alpha)$$

  That is, the output is produced as a linear combination of the input sequence shifted by n scaled by each element of the system response, $\boldsymbol{h}$.

- Taking the Fourier transform of both sides of equation (α), we can show that convolution in the time-domain is multiplication in the frequency domain

$$\begin{aligned} \hat{\boldsymbol{y}}(\boldsymbol{\omega}) &= \mathcal{F}\left\{ \sum_{r=-\infty}^{\infty} h[r]x[n-r] \right\} \\ &= \sum_{r=-\infty}^{\infty} h[r] \cdot \mathcal{F}\{x[n-r]\} \\ &= \hat{x}(\omega) \cdot \underbrace{\sum_{r=-\infty}^{\infty} h[r]e^{-j\omega r}}_{\hat{h}(\omega)} = \hat{\boldsymbol{x}}(\boldsymbol{\omega}) \cdot \hat{\boldsymbol{h}}(\boldsymbol{\omega}) \end{aligned}$$

Convolution as Polynomial Multiplication

This is best explained by an example question. Consider the input sequence $x[n]$ and the impulse response of some LTI filter, $h[n]$ such that:

$$x[n] = \{\underline{0}, 1, 3, 1, 2\} \qquad h[n] = \{1, \underline{-1}, 1\}$$

(a) Compute $\boldsymbol{x} \star \boldsymbol{h}$
(b) If $y[n] = x[n-2]$, compute $\boldsymbol{y} \star \boldsymbol{h}$

Solution

(a) We position the impulse response sequence on top of the input sequence making sure to line up the elements in both sequences where $n = 0$.

$$\begin{array}{rccccccccc} & & 1 & \underline{-1} & 1 & & & & & \\ \times & & & \underline{0} & 1 & 3 & 1 & 2 & & \\ \hline & 0 & 0 & 0 & & & & & & \\ & & 1 & -1 & 1 & & & & & \\ & & & 3 & -3 & 3 & & & & \\ & & & & 1 & -1 & 1 & & & \\ & & & & & 2 & -2 & 2 & & \\ \hline & & 0 & \underline{1} & 2 & -1 & -4 & -1 & 2 & \end{array}$$

Hence, we have $y[n] = \{0, \underline{1}, 2, -1, -4, -1, 2\}$

(b) The consequence of the shift in the input sequence is the $n = 0$ reference element. For the input sequence, we now have:

$$x[n] = \{0, 1, \underline{3}, 1, 2\}$$

And since the system is LTI, a shift in the input sequence MUST induce an identical shift in the output sequence. Hence, we have:

$$y[n] = \{0, 1, 2, \underline{-1}, 1, 4, -1, 2\}$$

Output Sequence Length After Convolution

- If the length of $\boldsymbol{x}$ is N elements long and the length of $\boldsymbol{h}$ is M elements long then the length of the resulting vector produced by the convolution is:

$$len(\boldsymbol{y}) = N + M - 1$$

Circular Convolution

- The convolution performed prior is called **linear convolution** and is performed between two aperiodic sequences.

- **Circular (cyclic) convolution** computes the convolution of one aperiodic sequence $\boldsymbol{h}$ with another sequence $\boldsymbol{x}$ that is N-periodic (i.e. repeats itself after N elements). This is expressed as mathematically as:

$$\boldsymbol{y} = (\boldsymbol{h} \star \boldsymbol{x}_N)[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x_N[n-k] \sum_{k=-\infty}^{\infty} h[k] \cdot \left(\sum_{r=\infty}^{\infty} x[n-k-rN] \right)$$

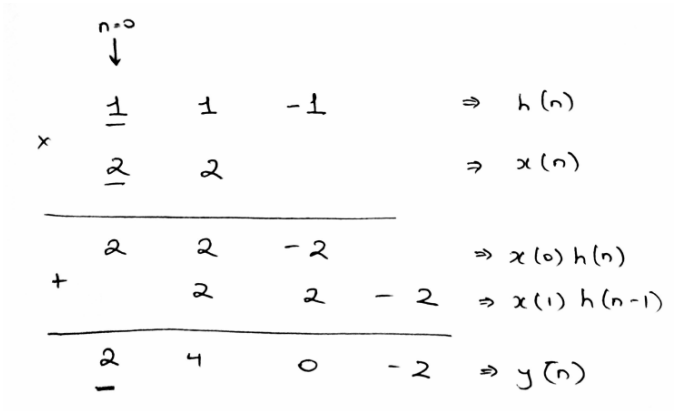
Here, $length(\boldsymbol{y}) = \max(N, M)$ where N is the length of sequence $\boldsymbol{x}$ and M is the length of sequence $\boldsymbol{y}$.

Cyclic Convolution Example

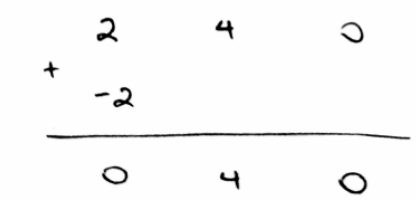
Consider two sequences $x[n]$ and $h[n]$ such that:

$$x[n] = \{\underline{2}, 2\} \qquad h[n] = \{\underline{1}, 1, -1\}$$

We would perform the *linear* convolution as expected:



However, the length of the output sequence needs to be equal to $\max(N, M)$ which in this case is 3, the length of $\boldsymbol{h}$. Hence, we take the first three elements of $\boldsymbol{y}$ and add to all subsequent elements that must be cycled back to $n = 0$. This is shown below as a better explanation.



Hence, we can say $y[n] = \{0, 4, 0\}$.

Z-Transform

- The Z-transform expressed a discrete sequence $x[n]$ as an infinite power series of z. The sum produces a polynomial in z^{-1} of degree N if $0 \leq n \leq N$. In general, though:

$$X(z) = \sum_{z=-\infty}^{\infty} x[n]z^{-n} \qquad (Bilateral\ z-transform)$$
$$= \sum_{z=0}^{\infty} x[n]z^{-n} \qquad (Unilater\ z-transform)$$

- For two discrete sequences $x[n]$ and $y[n]$ with Z-transforms $X(z)$ and $Y(z)$, we have:

$$X(z)Y(z) = (\boldsymbol{x} \star \boldsymbol{y})[n]$$

The coefficients from the multiplication can be found via linear convolution!

- A sample delay in a discrete LTI system is associated with multiplication z^{-r} where $r \in \mathbb{Z}$ is the sample delay. For instance:

$$y[n] = x[n] - \alpha y[n-1] \quad \overrightarrow{\mathcal{Z}\ trans.} \quad Y(z) = X(z) - \alpha Y(z)z^{-1}$$

- The Z-transform allows us to determine the transfer function in the $\mathcal{Z}$ domain. Using the above example, we have:

$$Y(z)[1 + \alpha z^{-1}] = X(z) \quad \Longrightarrow \quad H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \alpha z^{-1}}$$

- The transfer function obtained is generally the **closed-form** but can be re-converted to a **recursive form** by noting that such transfer functions originate for geometric progressions. Here, we see that:

$$S_{\infty} \triangleq \frac{a}{1-r} \equiv \frac{1}{1 - \alpha z^{-1}}$$

Hence, we can re-write this system transfer function as:

$$H(z) = \sum_{n=0}^{\infty} \alpha^n z^{-n}$$

We can deduce the impulse response of the system (discrete-time domain) to be:

$$h[n] = \alpha^n u[n]$$

Connection Between The Z-Transform & Fourier Transform

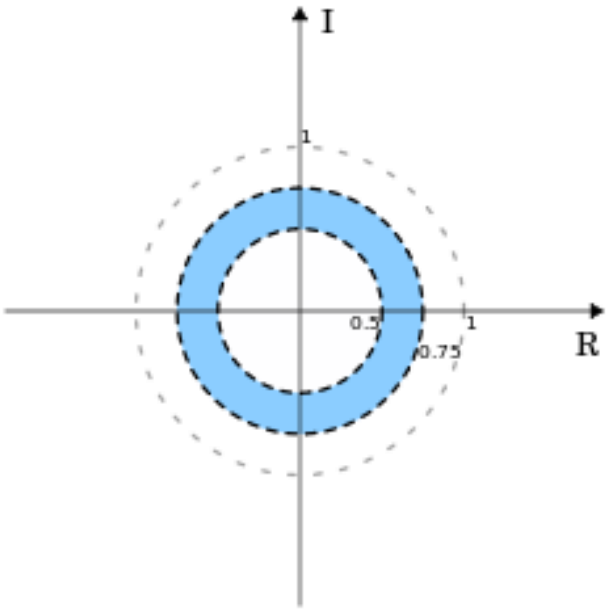
- If we let $z \in \mathbb{C}$, then a substitution of $z = re^{j\omega}$ into the general recursive form of the Z-transforms gives us the Fourier Transform of $x[n]$ with a scaling of r^n

$$X(re^{j\omega}) = \hat{x}_r(\omega) = \sum_{n=-\infty}^{\infty} x[n]r^n e^{j\omega n}$$

- If we want r to **not grow exponentially (diverge)**, we need to set $|r| \leq 1$. This ensures the power series is **convergent** instead. If we set $r = 1$, we get back the Fourier transform:

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

- The radius $r = r_0$ which makes the power series convergent is called the **RADIUS of convergence.**
- The *region* on $\mathbb{C}$ for which $|r| \leq 1$ is called the **REGION of convergence**.



Manipulating The Z-Transform

- If an LTI filter with impulse response, $h[n]$, is applied to a discrete sequence, $x[n]$ via convolution to produce an output response, $y[n]$, then their Z-transforms are related as a rational function:

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

- The above representation is called the **pole-zero representation** of the filter, ***h***
  - The ***zeros*** are the roots of $Y(z)$ that make $H(z) = 0$
  - The ***poles*** are the roots of $X(z)$ that act as discontinuities and make $H(z) \to \infty$

- The LTI filter is stable (does not diverge) if the ***poles are within the unit circle*** i.e. $|r| \leq 1$

- A filter is BIBO (Bounded Input – Bounded Output) stable is the output of a filter is bounded for every bounded input. That is:

$$|x[n]| \leq B_x, \quad \forall n \quad \Longrightarrow \quad |y[n]| \leq B_y, \quad \forall n$$

  We can prove this in the following way:

$$\begin{aligned} |y[n]| &= \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right| \\ &\leq \sum_{k=-\infty}^{\infty} |h[k]x[n-k]| \\ &= \sum_{k=-\infty}^{\infty} |h[k]| \cdot |x[n-k]| \\ &\leq \sum_{k=-\infty}^{\infty} |h[k]| \cdot B_x \\ &= B_y \end{aligned}$$

- Alternatively, the filter is BIBO stable if:

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty$$

- The stability triangle for a second-order system is a useful way of determining ***by inspection*** whether the system is stable AND whether it has real poles OR complex conjugate poles.

- Consider the following second-order system in the $\mathcal{Z}$ domain in terms of z

$$H(z) = \frac{z^2}{z^2 + bz + c}$$

The poles of $H(z)$ are found via the quadratic equation for the denominator polynomial

$$p_1, p_2 = \frac{-b \pm \sqrt{b^2 - 4c}}{2} \quad (\beta)$$

The ***discriminant*** determines whether the poles are real or complex. The two cases are:

1) **Real Poles:** $\Delta = b^2 - 4c \geq 0 \implies |b| \geq 2\sqrt{c}$
2) **Complex Poles:** $\Delta = b^2 - 4c \leq 0 \implies |b| < 2\sqrt{c}$

- The geometric relationship between b and c via the discriminant is a parabolic relationship if we plot c against b. The coordinates (b, c) give us a geometric interpretation to the condition for real and complex poles.

$$c = \frac{b^2}{4}$$

- The 'triangle' is associated with the condition for real poles. In relation to the ROC, the system is completely stable if the magnitude of both conjugate poles is **strictly less than 1**. That is:

$$|p_1| = |p_2| < 1$$

Substituting equation (β), we have:

$$\left| \frac{-b}{2} \pm \frac{\sqrt{b^2 - 4c}}{2} \right| < 1$$

By the *triangle inequality* we know $|a + b| < |a| + |b|$. Applying to our inequation, we have:

$$\frac{|b|}{2} + \frac{\sqrt{b^2 - 4c}}{2} < 1$$

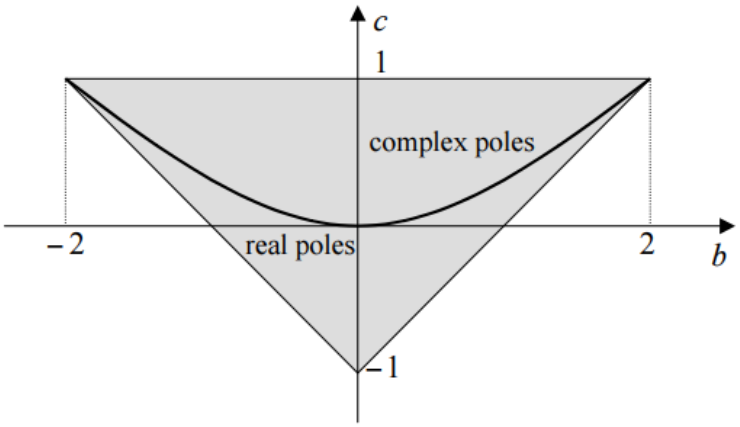
$$\sqrt{b^2 - 4c} < 2 - |b|$$

Squaring both sides and simplifying, we attain the critical result:

$$c < |b| - 1$$

The graph of $c = |b| - 1$ is an upside triangle on the number plane. The intersection of this graph with the parabola produces a **stability triangle** with vertices at $(0, -1)$, $(2,1)$ and $(-2,1)$

Both the parabola and the resultant triangle as a result of the intersection of the two graphs are shown in the diagram on the next page.



- This diagram conveniently shows that:
  - Any point (b, c) BELOW the parabola $(c < b^2/4)$ $\Longrightarrow$ **Distinct Real Poles**
  - Any point (b, c) ON the parabola $(c = b^2/4)$ $\Longrightarrow$ **Identical Real Poles**
  - Any point (b, c) ON the parabola $(c > b^2/4)$ $\Longrightarrow$ **Complex Conj. Poles**

Filter Properties

Linear Phase & Group Delay

- A casual filter, $\boldsymbol{h}$ will inevitably induce a delay on some input sequence $\boldsymbol{x}$
- However, not all filters induce non-uniform delay to all samples (delay is not the for all signals). For the filters, however, that do induce a constant delay such that:

$$x_\sigma(t) = x(t-\sigma)$$

The Fourier transform of the signal related the delayed signal to the original:

$$\hat{x}_\sigma(\omega) = e^{-j\omega\sigma} \cdot \hat{x}(\omega)$$

In this case, the filter has a transfer function $\hat{h}(\omega) = e^{-j\sigma\omega}$ if its role is simply to induce a constant delay.

- The filter is said to have a **linear phase** response since $\theta(\omega) = \angle\hat{h}(\omega)$ is a linear function of ω with a gradient of σ.

$$\theta(\omega) = \angle\hat{h}(\omega) = -\sigma\omega$$

- The **group delay ($\boldsymbol{t_g}$)** is the time-delay induced on all amplitude values on an input signal into the filter. t_g is the negative derivative of the phase. That is:

$$t_g = -\frac{d\theta(\omega)}{d\omega}$$

Condition for A Realisable (Implementable) Filter

- Any filter has the transfer function in the frequency domain as:

$$\hat{h}(\omega) = \left|\hat{h}(\omega)\right|e^{-j\sigma\omega} \quad \Longrightarrow \quad \left|\hat{h}(\omega)\right| = \frac{\hat{h}(\omega)}{e^{-j\sigma\omega}} = \hat{h}(\omega)e^{j\sigma\omega}$$

Since $\left|\hat{h}(\omega)\right|$ is even, we can say that such linear filters also satisfy negative frequencies via:

$$\hat{h}(-\omega) = \left|\hat{h}(-\omega)\right|e^{+j\sigma\omega} = \left|\hat{h}(\omega)\right|e^{j\sigma\omega} = \hat{h}(\omega)e^{j2\sigma\omega}$$

If we let $z = e^{j\omega}$, we can re-interpret the above relation in the $\mathcal{Z}$ domain as:

$$H(z^{-1}) = H(z) \cdot z^{2\sigma}$$

If we let $N = 2\sigma$ where N is the length of the filter, then we have: $\boldsymbol{H\left(z^{-1}\right) = H(z) \cdot z^{N}}$

- The main consequence is that the group delay MUST be half of the filter's length for such a filter to be implementable as a physical circuit.
- Such linear phase filters MUST be symmetric in that sense
- Such filters MUST NOT have any non-trivial poles (i.e. poles must be only those at the origin).
- Any non-trivial zeros/poles of a linear phase filter MUST appear in reciprocal pairs.

Minimum/Maximum Phase Filters & All-Pass Filters

- An **all-pass** filter is one whose magnitude response is unity

$$\left|\hat{h}(\omega)\right| = 1 \quad \forall\omega$$

- All poles, p_k of $H(z) = \hat{h}(\omega)|_{z=e^{j\omega}}$ must be paired with a corresponding reciprocal zero, $z_k = \frac{1}{p_k}$. To see why, consider the following lines which relate the $\hat{h}(\omega)$ to $H(z)$.

$$\left|\hat{h}(\omega)\right|^2 = 1$$

$$\hat{h}(\omega) \cdot \hat{h}^*(\omega) = 1$$

We also know that $\hat{h}^*(\omega) = \hat{h}(-\omega)$. Then:

$$\hat{h}(\omega) \cdot \hat{h}(-\omega) = 1$$

Let $z = e^{j\omega}$. Then:

$$H(z)H(z^{-1}) = 1$$

$$H(z) = \frac{1}{H(z^{-1})} = \prod_k \frac{\left(z - \frac{1}{p_k}\right)}{(z - p_k)}$$

- A **minimum-phase** filter is one whose zeroes lies INSIDE the unit circle.
    - Of all filters with the same magnitude response, the minimum phase version will have the LOWEST group delay
    - Minimum-phase filters can be reciprocated (inverted) and STILL be stable since zeros that were once inside the unit circle are now poles that are STILL in the unit circle (stable)

- Any filter with a pole-zero represented transfer function can be decomposed written in terms of a *minimum-phase* and an *all-pass* in the following way:

$$H(z) = H_{MIN}(z) \cdot H_{AP}(z)$$

Where we want ALL zeros in the minimum phase filter to be WITHIN the unit circle. This is best shown with an example scenario

Example Scenario
A filter has the following transfer function:

$$H(z) = \frac{(z-0.5)(z-1.3)}{(z-0.4)(z-0.6)}$$

The filter has a zero $z = 1.3 > 1$ which lies OUTSIDE the unit circle in $\mathbb{C}$. We can replace it with the reciprocated factor $\left(z - \frac{1}{1.3}\right)$ but in order to preserve the original filter, we need to divide by the factor. This gives:

$$H(z) = \frac{(z-0.5)(z-1.3)\left(z-\frac{1}{1.3}\right)}{(z-0.4)(z-0.6)\left(z-\frac{1}{1.3}\right)}$$

We can separate the red portions of $H(z)$ and let it be a standalone rational function. However, it follows the form of an all-pass filter! This is shown below.

$$H(z) = \underbrace{\frac{(z-0.5)\left(z-\frac{1}{1.3}\right)}{(z-0.4)(z-0.6)}}_{H_{MIN}(z)} \cdot \underbrace{\frac{(z-1.3)}{\left(z-\frac{1}{1.3}\right)}}_{H_{AP}(z)}$$

Filters

- A general causal LTI Filter can be written as a difference equation in the following way:

$$y[n] = \underbrace{\sum_{k=0}^{\infty} a_k x[n-k]}_{\substack{Feedforward \\ Terms}} + \underbrace{\sum_{k=0}^{\infty} b_k y[n-k]}_{\substack{Feedback \\ Terms}}$$

- An IIR (*Infinite Impulse Response*) filter is one where $h[n]$ is an infinite sequence of filter taps. Typically, IIR filters are generated by introducing non-trivial poles (which introduces feedback coefficients into the difference equation)
  - For this happen $a_k \neq 0$ and $b_k \neq 0$

- An FIR filter is one where $b_k = 0$ for all k (i.e. no feedback terms). Then, we have:

$$y[n] = \sum_{k=0}^{\infty} a_k x[n-k]$$

  The Z-transform of an FIR filter produces an **all-zero** filter with trivial poles at the origin. The number of poles at the origin is **equal** to the number of zeros in the FIR filter's transfer function.

- The **DC gain**, G_{DC} is the gain of the filter when $\omega = 0$. In the $\mathcal{Z}$ domain, this would be when $z = e^{j\times 0} = 1$. In other words:

$$G_{DC} = H(z = 1)$$

- The **Nyquist gain**, G_{Nyq} is the gain at the filter at $\omega = \pi$ (associated with the Nyquist frequency for some sampled signal). This would correspond to $z = -1$. That is:

$$G_{Nyq} = H(z = -1)$$

FIR and IIR Filter Design

FIR Filter Design Methods

Windowing Method

- Suppose we have a filter $h_d[n]$ with an impulse response with infinite support (making it IIR). Let $h[n]$ be our *realisable* designed impulse response that generally is an FIR filter (which is practically realisable).

- Introduce a window function/sequence $w[n]$ with sample support equal to the desired filter length. Then, we have:

$$h[n] = h_d[n] \cdot w[n]$$

Which in the Fourier domain is the circular convolution of $\hat{h}_d(\omega)$ and $\hat{w}(\omega)$ (up to a constant $1/2\pi$) which ensures the desired filter's frequency response remains within $\omega \in (-\pi, \pi)$.

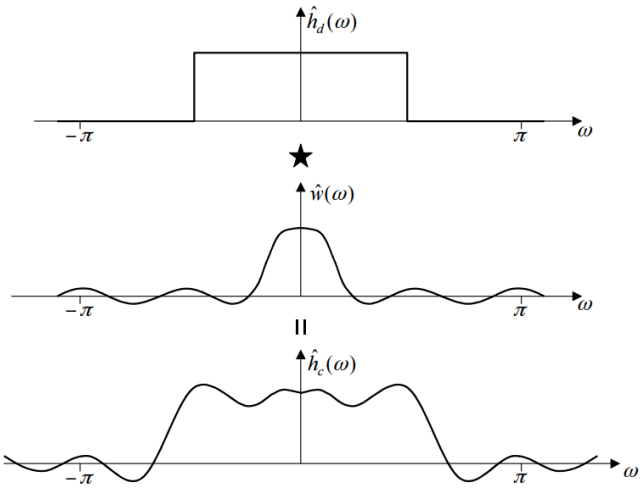
$$\hat{h}_d(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} h_d(\theta) \cdot w\big((\omega - \theta)\ mod\ 2\pi\big)d\theta$$

Where $\omega = \theta$ is some frequency shift induced as a result of the convolution.

- The figure below shows the effect of the rectangular window defined in the time-domain

$$w[n] = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & otherwise \end{cases}$$

Whose frequency domain representation is the non-causal sinc function with infinite support. The effect of the window on the produced spectrum is a *smearing* effect with the cutoffs of the filter decreases in steepness with rippling and sidelobes as a result of the window.



Least-Squares Method

- This method aims to minimise the squared error between the desired filter response $\hat{h}_d(\omega)$ and the actual filter response $\hat{h}(\omega)$ via some weighting function $|\hat{\rho}(\omega)|^2$. That is:

$$\varepsilon(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{\rho}(\omega)|^2 \cdot \left|\hat{h}_d(\omega) - \hat{h}(\omega)\right|^2 \, d\omega$$

  Where κ is defined in the time-domain as the weighted error between the impulse response of the actual and desired impulse response:

$$\kappa[n] = \rho[n] \star (h[n] - h_d[n])$$

  Then, the sum of the squared errors that needs to be minimised is (in the time-domain)

$$\varepsilon[n] = \sum_{n=0}^{N-1} |\kappa[n]|^2$$

- The desired filter's impulse response MUST be:

$$h_d[n] = \{a_L, a_{L+1}, a_{L+2}, \dots, a_U\}$$

$$\hat{h}_d(\omega) = \sum_{n=L}^{n=U} a_n \, e^{-j\omega n}$$

  This makes the objective function become:

$$\varepsilon(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{\rho}(\omega)|^2 \cdot \left|\sum_{n=L}^{n=U} a_n \, e^{-j\omega n} - \hat{h}(\omega)\right|^2 \, d\omega$$

  Noting that in $\mathbb{C}$ we have the result $|z|^2 = zz^*$. Then, we have:

$$\varepsilon(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{\rho}(\omega)|^2 \cdot \left(\sum_{n=L}^{n=U} a_n \, e^{-j\omega n} - \hat{h}(\omega)\right)\left(\sum_{n=L}^{n=U} a_n \, e^{+j\omega n} - \hat{h}^*(\omega)\right) d\omega$$

  Taking the partial derivative of ε with respect to $a_p \in \boldsymbol{a}$, and simplifying, we have:

$$\frac{\partial \varepsilon}{\partial a_p} = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{p}(\omega)|^2 \left\{\left[\hat{h}(\omega)e^{j\omega p} - \hat{h}^*(\omega)e^{-j\omega p}\right] - \left[\sum_{n=L}^{U} a_n\left(e^{j\omega(n-p)} + e^{-j\omega(n-p)}\right)\right]\right\} d\omega$$

Setting the derivative equal to zero and rearranging, we can see that:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\left(\hat{h}(\omega)e^{j\omega p}-\hat{h}^*(\omega)e^{-j\omega p}\right)d\omega = \sum_{n=L}^{U}a_n\cdot\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{p}(\omega)|^2\left(e^{j\omega(n-p)}+e^{-j\omega(n-p)}\right)d\omega$$

Splitting the integrals on the LHS (noting that $\Im$ denotes the *imaginary part*):

$$\begin{aligned}L &= \frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega-\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}^*(\omega)e^{-j\omega p}d\omega\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega-\frac{1}{2\pi}\left(\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega\right)^*\\ &= 2\times\Im\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega\right]\end{aligned}$$

For both integrals, set $\omega=-\alpha\rightarrow d\omega=-d\alpha$. Hence, we have:

$$\begin{aligned}L &= 2\times\Im\left[\frac{-1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega\right]\\ &= 2\times\Im\left[\frac{-1}{2\pi}\int_{\pi}^{-\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega\right]\\ &= 2\times\Im\underbrace{\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)e^{j\omega p}d\omega\right]}_{d[p]\in\mathbb{R}}\end{aligned}$$

The integral is the inverse DTFT of some real sequence $d[n]\in\mathbb{R}$ whose DTFT can be defined as $\hat{d}(\omega)=|\hat{\rho}(\omega)|^2\cdot\hat{h}(\omega)$. And since a real number has no imaginary parts, then:

$$L = 2\times d[p]$$

A similar calculation can be performed for the right-hand side such that the integral involved is the inverse DTFT of some real sequence $r[n]\in\mathbb{R}$ such that its DTFT is $\hat{r}(\omega)=|\hat{\rho}(\omega)|^2$

$$R = 2\sum_{n=L}^{U}a_n r[n-p]$$

Hence, we have the final relation:

$$\sum_{n=L}^{U}a_n r[n-p] = d[p]$$

This equation can be re-expressed in the form of a matrix equation. To see how this is the case, we will construct a small system of equations for a few values of p and then generalise to get an idea of how the equation will pop out.

Equation 1 ($p = L$):

$$a_L r[L - L] + a_{L+1} r[(L+1) - L] + \cdots + a_U r[U - L] = d[L]$$
$$\Downarrow$$
$$a_L r[0] + a_{L+1} r[1] + \cdots + a_U r[U - L] = d[L]$$

Equation 2 ($p = L + 1$):

$$a_L r[L - (L+1)] + a_{L+1} r[(L+1) - (L+1)] + \cdots + a_U r[U - (L+1)] = d[L+1]$$
$$\Downarrow$$
$$a_L r[-1] + a_{L+1} r[0] + \cdots + a_U r[U - (L+1)] = d[L+1]$$

Equation 3 ($p = L + 2$):

$$a_L r[-2] + a_{L+1} r[-1] + \cdots + a_U r[U - (L+2)] = d[L+2]$$

$$\vdots$$

Last Equation ($p = U$)

$$a_L r[L - U] + a_{L+1} r[(L+1) - U] + \cdots + a_U r[U - U] = d[U]$$

The equations can then be rearranged such that a matrix $\boldsymbol{R}$ consisting of elements of $r[n]$ in a $(U - L + 1) \times (U - L + 1)$ matrix is dot multiplied with a vector of the FIR filter coefficients $\boldsymbol{a} = [a_L, a_{L+1}, \dots, a_U]^T$ to produce a vector $\boldsymbol{d} = \left[d[L], d[L+1], \dots, d[U]\right]^T$. That is:

$$\underbrace{\begin{bmatrix} r[0] & r[1] & \dots & r[U-L] \\ r[-1] & r[0] & \dots & r[U-L-1] \\ \vdots & \vdots & \ddots & \vdots \\ r[L-U] & r[L-U+1] & \dots & r[0] \end{bmatrix}}_{R} \cdot \underbrace{\begin{bmatrix} a_L \\ a_{L+1} \\ \vdots \\ a_U \end{bmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{bmatrix} d[L] \\ d[L+1] \\ \vdots \\ d[U] \end{bmatrix}}_{\boldsymbol{d}}$$

That is, the equation can be written succinctly as:

$$\boldsymbol{Ra} = \boldsymbol{d} \quad \Longrightarrow \quad \boldsymbol{a} = \boldsymbol{R}^{-1}\boldsymbol{d}$$

Where the matrix $\boldsymbol{R}$ is symmetric such that $\boldsymbol{R}^T = \boldsymbol{R}$ and exhibits a Toeplitz structure (meaning all diagonal elements are equal to each other) that allow the inversion of $\boldsymbol{R}$ to be simple and robust!

Frequency Sampling

- Suppose we want a desired frequency $\hat{h}_d(\omega)$ (the DTFT of $h_d[n]$ not necessarily finite) bandlimited in $\omega \in (-\pi, \pi)$. Let our resultant designed FIR filter be $h[n]$ with sample support $n = 0 \dots M-1$ such that is of length M.

- Frequency sampling requires computing the DFT of $\hat{h}_d(\omega)$ at evenly spaced discrete frequency bins within the bandlimited range of $(-\pi, \pi)$. If the DFT of the desired frequency response is $H_d[k]$, then:

- If we sample at M frequencies of $\hat{h}_d(\omega)$, then the k^{th} frequency is given by $\omega_k = \frac{2\pi k}{M}$. To ensure we remain within the bandlimited range of the DTFT, we consider the modulus of the frequency:

$$\omega_k = \frac{2\pi k}{M} \; mod \; 2\pi$$

The DFT of the $\hat{h}_d(\omega)$ can be denoted as $H[k]$ such that:

$$H[k] = \hat{h}_d(\omega)|_{\omega=\frac{2\pi k}{M} mod \; 2\pi}$$

Then, the inverse DFT of $H[k]$ will give us $h[n]$. However, we can see that one of the critical disadvantages of this technique is the risk of sampling at too few frequency bins. We lose spectral information in the final FIR filter design that may not accurately reflect the DTFT of the desired filter response (which it most likely would not at any rate).

IIR Filter Design

The methods for IIR filter design that are described in these notes are:

- Impulse-Invariant Method
- Bi-Linear Transformation
- Least-Squares for IIR

Impulse – Invariant Method

- This method creates a digital approximation of an analog filter with frequency response $H_a(s)$ (where $s = \sigma + j\Omega$) whose impulse response is $h_a(t)$. Note that Ω is the analog frequency.

- This begins by assuming the analog filter can be decomposed via partial fraction decomposition into a series of N single pole transfer functions:

$$H_a(s) = \sum_{i=1}^{N} \frac{K_i}{s - p_i}$$

Whose inverse Laplace transform back into the analog time-domain would be:

$$h_a(t) = \mathcal{L}^{-1}\{H_a(s)\} = \sum_{i=1}^{N} K_i e^{p_i t}$$

- The analog impulse response is then sampled at some rate $F_s = \frac{1}{T}$ such that the digital version $h[n]$ of sample support $n = 0, 1, 2, \dots N-1$ is given as:

$$h[n] \triangleq h_a(nT) = \sum_{i=1}^{N} K_i e^{p_i nT}$$

The Z-Transform is then computed for the digital impulse response to obtain $H(z)$ such that:

$$H(z) = \mathcal{Z}\{h[n]\} = \sum_{p=1}^{N} \frac{K_i z}{z - e^{p_i}} = \sum_{p=1}^{N} \frac{K_i}{1 - \left(\frac{e^{p_i}}{z}\right)}$$

Let ω be the digital frequency after the mapping. Hence, the mapping from the digital to the analog domain is defined as:

$$z \Leftrightarrow e^{s_i T} = e^{(\sigma + j\Omega)T} = e^{\sigma T} e^{j\Omega T} \quad (Where\ z = re^{j\omega})$$

- If the analog filter is stable and causal, then so will be the digital filter after the conversion. This is reliant on the location of the poles in the s domain:

  - Recall that if the pole is on the **left-half plane (LHP)** in the s domain, then the pole will be within the unit circle in the z domain. i.e. $Re\{s_i\} \leq 0 \Longrightarrow r = |z_i| \leq 1$
  - Furthermore, $\omega = \Omega T$ or $\mathrm{T} = \frac{\omega}{\Omega}$ which leads to time-aliasing. That is, the digital signal is sampled at a rate much slower than the analog signal was originally.

Bi-Linear Transformation

-

Least-Squares for IIR Filter Design

Filter Structures

Direct Form

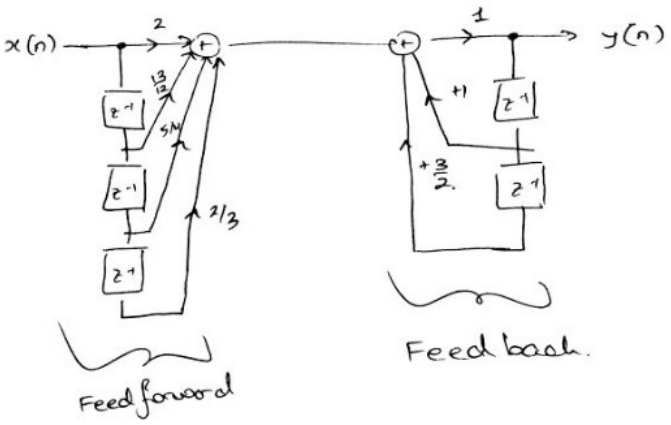
***Example Question**: Implement the following filter in direct form if the filter's difference equation is:*

$$y[n] = 2x[n] + \frac{13}{12}x[n-1] + \frac{5}{4}x[n-2] + \frac{2}{3}x[n-3] + y[n-1] + \frac{3}{2}y[n-2]$$

We take the Z-transform on both sides to obtain the following transfer function:

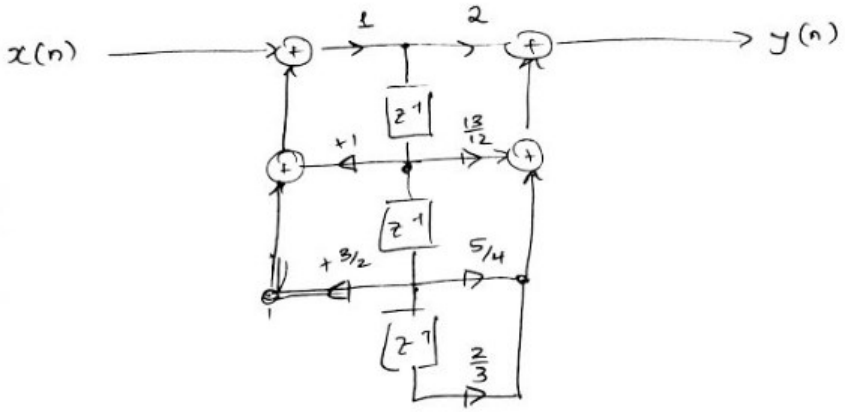
$$H(z) = \frac{2 + \frac{13}{12}z^{-1} + \frac{5}{4}z^{-2} + \frac{2}{3}z^{-3}}{1 - z^{-1} - \frac{3}{2}z^{-2}}$$

The following diagram shows the numerator and denominator as separate sections separated by summers (accumulators). Each delayed term is associated with a gain coefficient following the sign in the numerator polynomial. For the denominator section, the coefficients are associated with feedback (backward) gains which have opposite signs to those in the denominator polynomial.



Canonical Form

Using the previous example, this is a transposed version of the direct-form representation. Instead, will reduce the number of shift registers required.



Parallel

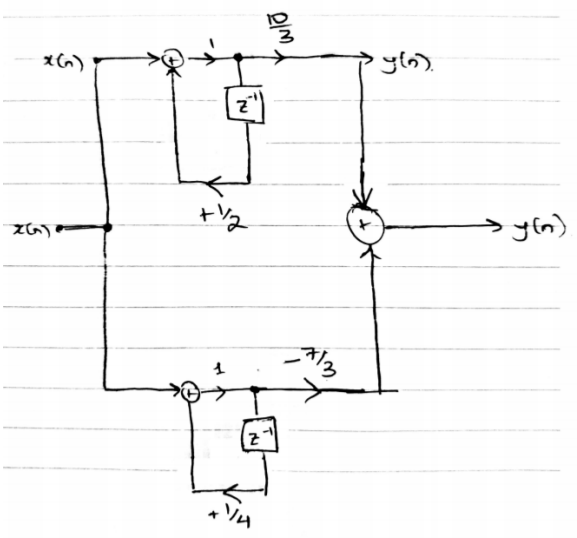
Here we try to re-express a transfer function as a SUM of two lower-order filters (generally 1st/2nd order sections) whose outputs are summed at the end of the filter structure

$$H(z) = \frac{1 + \frac{1}{3}z^{-1}}{1 - \frac{3}{4}z^{-1} + \frac{1}{8}z^{-2}}$$

It can be shown that the denominator can be factorised and expressed such that:

$$H(z) = \frac{1 + \frac{1}{3}z^{-1}}{\left(1 - \frac{1}{2}z^{-1}\right)\left(1 - \frac{1}{4}z^{-1}\right)} = \frac{\frac{10}{3}}{1 - \frac{1}{2}z^{-1}} + \frac{\left(-\frac{7}{3}\right)}{1 - \frac{1}{4}z^{-1}}$$

The resultant diagram for the filter structure can be drawn like that shown below



Cascade Form

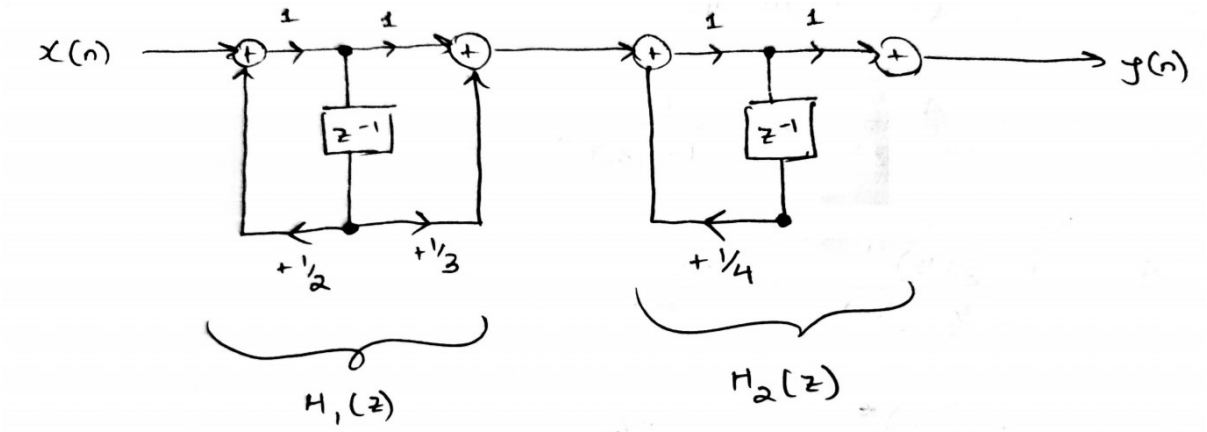
The numerator and denominator polynomials are first factorised into smaller first/second order polynomials. We then re-express the rational function as the product of two smaller order transfer functions in the form:

$$H(z) = H_1(z) \times H_2(z)$$

Where diagrammatically, the filters are joined by accumulators in between. For our example we could express our transfer function as:

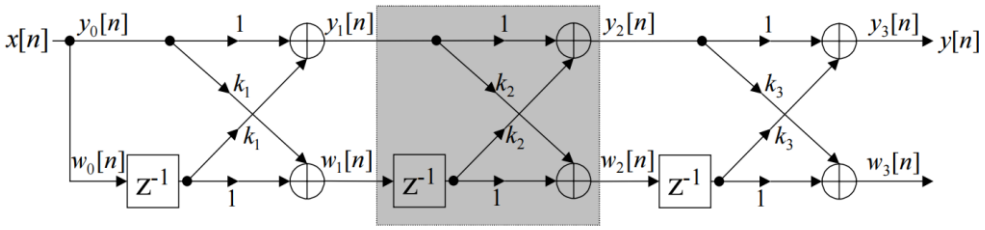
$$H(z) = \frac{1 + \frac{1}{3}z^{-1}}{\left(1 - \frac{1}{2}z^{-1}\right)\left(1 - \frac{1}{4}z^{-1}\right)} = \left(\frac{1 + \frac{1}{3}z^{-1}}{1 - \frac{1}{2}z^{-1}}\right) \times \left(\frac{1}{1 - \frac{1}{4}z^{-1}}\right) = H_1(z) \times H_2(z)$$

The corresponding filter structure can be drawn for $H_1(z)$ and $H_2(z)$ with the output of $H_1(z)$ cascaded to the input of $H_2(z)$



Lattice FIR Filters

FIR filters can be designed using *lattice* filters which apply a repeated first order structure whose network topology is fixed. Hence, the only work that needs to be done is computing the coefficients of each single order filter. The figure below shows the structure of a 3rd order FIR lattice filter.



Steps for Computing FIR Lattice Coefficients

The following steps are performed to compute the coefficients for the m^{th} first order lattice block:

1) Take the Z-transform of the difference equation (if provided one)

2) Find $H(z) = k_0 A_m(z)$ where k_m is the m^{th} coefficient associated with the z^{-m} term in $A_m(z)$

3) Find $B_m(z) = \underbrace{z^{-m} A_m(z^{-1})}_{\substack{Write\ in\ reverse \\ order\ to\ A_m(z)}}$

4) Find $A_{m-1}(z)$ where:

$$A_{m-1}(z) = \frac{A_m(z) - k_m B_m(z)}{1 - k_m^2}$$

5) Repeat steps 1 – 4 until $k_1, k_2, k_3 \ldots k_m$ are found.

Example Question

Implement the following FIR filter using a lattice structure:

$$y[n] = 2x[n] + \frac{13}{12}x[n-1] + \frac{5}{4}x[n-2] + \frac{2}{3}x[n-3]$$

1) The transfer function can be computed to be:

$$H(z) = 2 + \frac{13}{12}z^{-1} + \frac{5}{4}z^{-2} + \frac{2}{3}z^{-3}$$

2) Take out a factor of 2 to make the constant unity:

$$H(z) = \underbrace{2}_{k_0} \cdot \left(1 + \frac{13}{24}z^{-1} + \frac{5}{8}z^{-2} + \underbrace{\frac{1}{3}}_{k_3}z^{-3}\right) \equiv k_0 \cdot A_3(z)$$

We can see that $\boldsymbol{k_0 = 2}$ even though this is not used in the filter. From the polynomial $A_3(z)$, we can see that $\boldsymbol{k_3 = \frac{1}{3}}$ (coefficient of the z^{-3} term).

3) The polynomial $B_3(z)$ is:

$$B_3(z) = \frac{1}{3} + \frac{5}{8}z^{-1} + \frac{13}{24}z^{-2} + z^{-3}$$

4) We can now calculate the polynomial $A_2(z)$ to find k_2 such that:

$$A_2(z) = \frac{2\left(1 + \frac{13}{24}z^{-1} + \frac{5}{8}z^{-2} + \frac{1}{3}z^{-3}\right) - \frac{1}{3}\left(\frac{1}{3} + \frac{5}{8}z^{-1} + \frac{13}{24}z^{-2} + z^{-3}\right)}{1 - \left(\frac{1}{3}\right)^2}$$

$$= 1 + \frac{3}{8}z^{-1} + \frac{1}{2}z^{-2} \quad \Rightarrow \quad \boldsymbol{k_2 = \frac{1}{2}}$$

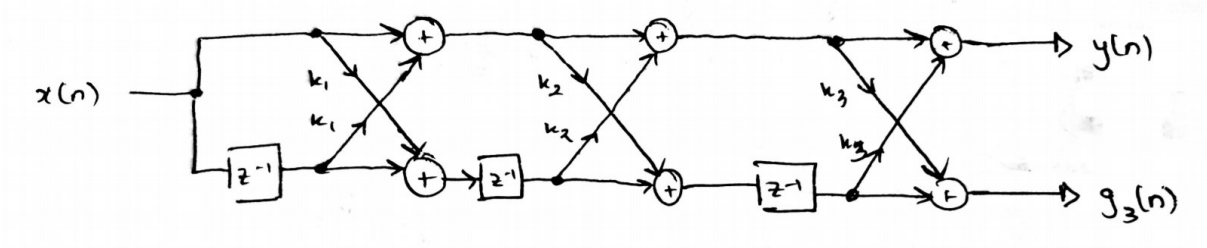
This implies that:

$$B_3(z) = \frac{1}{2} + \frac{3}{8}z^{-1} + \frac{1}{2}z^{-2}$$

5) Hence, we can now calculate for k_1

$$A_1(z) = \frac{A_2(z) - k_2 B_2(z)}{1 - k_2^2} = 1 + \frac{1}{4}z^{-1} \quad \Rightarrow \quad \boldsymbol{k_1 = \frac{1}{4}}$$

The filter can now be drawn in this fashion:



All-Pole IIR Lattice Filter

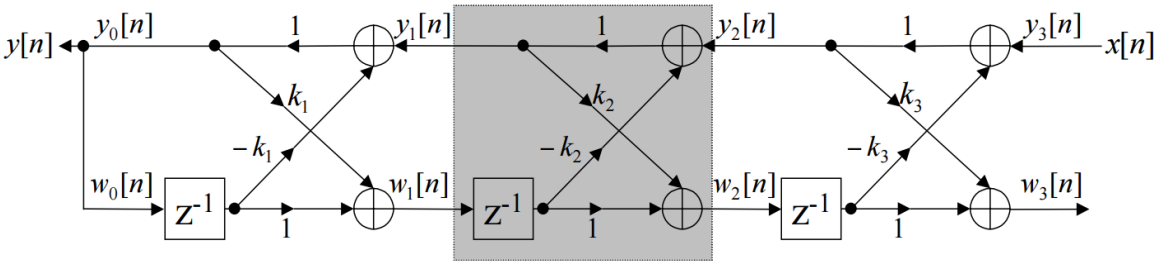
If our filter is defined as an all-pole system, in the form:

$$H_{IIR}(z) = \frac{1}{B(z)}$$

Then the lattice coefficients can be found for the IIR version first:

$$H_{FIR}(z) = B(z)$$

And the following modified lattice filter diagram can be drawn to cater for the feedback coefficients in the following way:

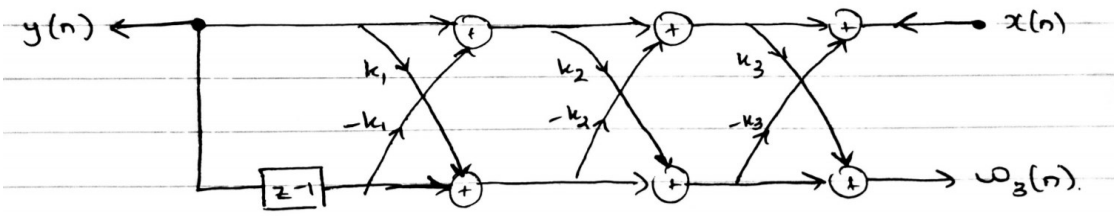


Where the structure looks identical except $y[n]$ is taken as the output at the beginning (left-hand side) of the structure while the input is taken as the input at the end of the structure (right-hand side). The feedback coefficients are catered for in the feedback direction with a negative sign associated with the related coefficients.

For our example, perhaps our transfer function might now be:

$$H(z) = \frac{1}{2 + \frac{13}{12}z^{-1} + \frac{5}{4}z^{-2} + \frac{2}{3}z^{-3}}$$

Then, our resultant all-pole lattice filter diagram would be drawn as:



Fixed-Point Arithmetic

Discrete Fourier Transform (DFT)

- The DTFT produces a *continuous spectrum*, $\hat{x}(\omega)$ when the FT is performed in a discrete-time sequence, $x[n]$ within the bandlimited domain $\omega \in (-\pi, \pi)$

- However, the $\hat{x}(\omega)$ has an infinite sample support (i.e. continuous) and this is effectively impossible to implement in a practical setting (e.g. computationally).

- The ***discrete Fourier Transform (DFT)*** is performed by **sampling the DTFT spectrum** to produce a discrete frequency spectrum, $X[k]$.

- The DFT is defined mathematically initially as for a discrete sequence $x[n]$ with finite support in the range $0 \leq n \leq N-1$ as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi k}{N}n} \qquad \boldsymbol{OR} \qquad X[k] = \sum_{n=0}^{N-1} x[n] \cdot w^{-nk}$$

That is, the sample support of $X[k]$ **also** matches that of the discrete time-domain sequence. We may think of the mapping visually like so:

$$\{x_0, x_1, x_2, \dots, x_{N-1}\} \xrightarrow{Discrete\ Fourier\ Transform} \{X_0, X_1, X_2, \dots, X_{N-1}\}$$

- The mapping is achieved by projecting each sample of the input sequence onto the k^{th} discrete Fourier vector, $\boldsymbol{w}_k$ where:

$$\begin{aligned} \boldsymbol{w}_k &= \left[1, e^{j\frac{2\pi k}{N}}, e^{j\frac{2\pi k \cdot 2}{N}}, e^{j\frac{2\pi k \cdot 3}{N}}, e^{j\frac{2\pi k \cdot 4}{N}} \dots, e^{j\frac{2\pi k \cdot (N-1)}{N}}\right]^T \\ &= \left[1, w^k, w^{2k}, w^{3k}, w^{4k} \dots, w^{k(N-1)}\right]^T \end{aligned}$$

Where $w = e^{j\cdot\frac{2\pi}{N}}$ and each element in $\boldsymbol{w}_k$ takes the form w^{nk}. Hence, the N-point DFT output vector is produced by multiplying N-point discrete sequence with the N-point DFT matrix that is shown below:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & \dots & w^{N-1} \\ 1 & w^2 & w^4 & \ddots & w^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & \dots & w^{(N-1)(N-1)} \end{bmatrix}}_{N-point\ DFT\ Matrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

- The DFT can also be written as an **inner product**:

$$X[k] = \langle \boldsymbol{x}, \boldsymbol{w}_k \rangle$$

- The inverse DFT is expressed as:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot w^{nk} \qquad \boldsymbol{OR} \qquad \boldsymbol{x} = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot \boldsymbol{w}_k$$

- If we want to work with an orthonormal basis, we can work the basis vectors:

$$\boldsymbol{w}_k' = \frac{1}{\|\boldsymbol{w}_k\|}\boldsymbol{w}_k = \frac{1}{\sqrt{N}}\boldsymbol{w}_k$$

We can also normalise the DFT length by dividing by the normal of $\boldsymbol{X}$ which is $\sqrt{N}$:

$$X'[k] = \frac{1}{\sqrt{N}}X[k] = \langle \boldsymbol{x}, \boldsymbol{w}_k{}' \rangle$$

Hence, we can also say that:

$$\boldsymbol{x} = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot \boldsymbol{w}_k = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}}X[k] \cdot \frac{1}{\sqrt{N}}\boldsymbol{w}_k = \sum_{k=0}^{N-1} X'[k] \cdot \boldsymbol{w}_k{}' = \sum_{k=0}^{N-1} \langle \boldsymbol{x}, \boldsymbol{w}_k{}' \rangle \cdot \boldsymbol{w}_k{}'$$

Connection Between the DFT and DTFT

- The DTFT is defined for sequences with infinite support whereas the DFT is defined for sequences with finite support

- Both are equivalent if we impose the condition that:

$$\omega = \frac{2\pi k}{N}$$

Where now ω is a discrete frequency dependent on the finite length of the signal, N and the integer parameter, k.

- It can be clearly seen that:

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j \cdot \frac{2\pi k}{N} n} \\ &= \left(\sum_{n=-\infty}^{\infty} x[n] e^{-jn\omega} \right)_{\omega = \frac{2\pi k}{N}} \\ &= \hat{x}(\omega)|_{\omega = \frac{2\pi k}{N}} \end{aligned}$$

However, this does not restrict the value of ω to be within the ideal bandlimited domain $\omega \in (-\pi, \pi)$. Hence, we need to compute its ***modulo*** to stay in this interval.

$$x \bmod 2\pi \triangleq x - 2\pi \left\lfloor \frac{x}{2\pi} + \frac{1}{2} \right\rfloor$$

Hence, the equivalence between the DTFT and DFT is found when we let:

$$\omega = \left(\frac{2\pi k}{N}\right) mod\ 2\pi$$

That is:

$$X[k] = \hat{x}(\omega)|_{\omega = \frac{2\pi k}{N} mod\ 2\pi}$$

DFT Requires Circular Convolution in The Time-Domain

- The DTFT operation implies that ***linear convolution*** in the time-domain is multiplication in the frequency domain. That is, the following relationship is true.

$$y[n] = (\boldsymbol{x} \star \boldsymbol{h})[\boldsymbol{n}] \quad \overset{DTFT}{\Longrightarrow} \quad \hat{y}(\omega) = \hat{x}(\omega)\, \hat{h}(\omega)$$

  The question is, does this hold true for the DFT? That is:

$$y[n] = (\boldsymbol{x} \star \boldsymbol{h})[\boldsymbol{n}] \quad \overset{DFT}{\Longrightarrow} \quad Y[k] = X[k]\, H[k]$$

- To understand this, we consider the initial conditions for all sequences involved in convolution operation. We suppose that:
  - $x[n]$ has finite sample support from $0\ to\ N-1$ where $N \in \mathbb{Z}$ is its length
  - $h[n]$ has finite sample support from $0\ to\ M-1$ where $M \in \mathbb{Z}$ is its length
  - $y[n]$ has finite sample support from $0\ to\ P-1$ where $P \in \mathbb{Z}$ is its length
  - **Suppose that $\boldsymbol{N > M}$.**

  Where P would be the length of the output sequence after the convolution. Using the traditional linear convolution, we would have:

$$P = N + M - 1$$

  However, the operation $Y[k] = X[k]\, H[k]$, requires that the length of X and H **are the same** (since its element-by-element multiplication)

  There are two cases for which we can account for this. We could either:

  - **Case 1:** Increase the length of $h[n]$ to the extended sequence $h_e[n]$ with a new sample support of 0 to $N-1$ to match the length of $x[n]$
  - **Case 2:** Increase the length of both $x[n]$ to $x_e[n]$ and $y[n]$ to $y_e[n]$ such that the sample support for their extended sequences are 0 to $P-1$ to match the length of $y[n]$

  In each case, the sequence is achieved by **zero-padding** which is adding **zeros to the end of the sequences**.

Case 1: Padding to $h[n]$ to match $x[n]$

We now pad $h[n] \rightarrow h_e[n]$ such that the support of $h_e[n]$ is now 0 to $N-1$. As such, the sample support for $y[n]$ is also from 0 to $N-1$. We want to prove the whether the following equivalence holds:

$$y[n] = (\boldsymbol{x} \star \boldsymbol{h})[\boldsymbol{n}] \quad \overset{DFT}{\Longrightarrow} \quad Y[k] = X[k]\,H[k]$$

Start by considering the *inverse DFT* of $Y[k]$:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} Y[k] e^{j\cdot\frac{2\pi k}{N}n}$$

But $Y[k] = X[k]H[k]$. Substituting, we have:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} (X[k]H[k]) e^{j\cdot\frac{2\pi k}{N}n}$$

Since $X[k]$ and $H[k]$ both are invertible, we can re-write the above equation in terms of their inverse DFT's:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{l=0}^{N-1} x[l]e^{-j\cdot\frac{2\pi k}{N}l}\right)\left(\sum_{m=0}^{N-1} h[m]e^{-j\cdot\frac{2\pi k}{N}m}\right)e^{j\cdot\frac{2\pi k}{N}n}$$

We now want to combine all the exponential terms together. This gives:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{l=0}^{N-1} x[l]\sum_{m=0}^{N-1} h[m]\right]\cdot e^{j\cdot\frac{2\pi k}{N}(n-m-l)}$$

Commutativity allows the order of the multiplication to change to:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} e^{j\cdot\frac{2\pi k}{N}(n-m-l)}\cdot\left[\sum_{l=0}^{N-1} x[l]\sum_{m=0}^{N-1} h[m]\right]$$

Now we let $a = e^{j\cdot\frac{2\pi}{N}(n-m-l)}$. Hence, we have:

$$y[n] = \frac{1}{N}\cdot\left(\sum_{k=0}^{N-1} a^k\right)\cdot\left(\sum_{l=0}^{N-1} x[l]\sum_{m=0}^{N-1} h[m]\right)$$

The sum $\sum_k a^k$ can take one of two values depending on the value of a. If $a \neq 1$, then the sum forms a geometric progression in $\mathbb{C}$. However, if $a = 1$, then we no longer have a geometric progression. The cases are summarised as:

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & if \quad a = 1 \\ \dfrac{1-a^N}{1-a} & if \quad a \neq 1 \end{cases}$$

The second case can be simplified further by considering the value for a^N:

$$a^N = e^{j\cdot\frac{2\pi}{N}(n-m-l)\times N} = e^{j\cdot 2\pi(n-m-l)} = 1 \qquad since\ (n-m-l) \in \mathbb{Z}$$

Which implies that $\frac{1-a^N}{1-a} = \frac{1-1}{1-a} = 0$. Hence, the results of the sum can be simplified to:

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & if \quad a = 1 \\ 0 & if \quad a \neq 1 \end{cases}$$

For the first case to be true, we need $a = 1$. This can ONLY occur if $(n - m - l)$ is an integer multiple of N. That is to say that:

$$n - m - l = rN \qquad (r \in \mathbb{Z})$$

Alternatively, solving for l we can write this condition as:

$$\begin{aligned} l &= n - m - rN \\ &= (n - m)\ mod\ N \\ &= (n - m)_N \end{aligned}$$

The above condition for l (which is the indexing for the DFT of $x[n]$) suggests that circular convolution is required for $x[n]$. That is:

$$\begin{aligned} y[n] &= \frac{1}{N} \cdot N \cdot \left(\sum_{m=0}^{N-1} x[(n-m)_N] h[m] \right) \\ &= \sum_{m=0}^{N-1} h[m] x[(n-m)_N] \end{aligned}$$

But the above sum is the very definition of **circular convolution!** Let $\circledast$ denote circular convolution. Then, we can say that:

$$y[n] = (h \circledast x)[n] \quad \overset{DFT}{\Longrightarrow} \quad Y[k] = H[k]X[k]$$

Case 2: Padding BOTH $h[n]$ and $x[n]$ to match $y[n]$

The computation is very similar except we want to extend $x[n] \to x_e[n]$ and $h[n] \to h_e[n]$ such that their sample support matches $y[n]$ which has finite support from 0 to $P - 1$. Where $P = N + M - 1$

$$y[n] = \frac{1}{P} \sum_{k=0}^{P-1} e^{j \cdot \frac{2\pi k}{P}(n-m-l)} \cdot \left[\sum_{l=0}^{P-1} x_e[l] \sum_{m=0}^{P-1} h_e[m] \right]$$

Likewise, we will land in the same condition for the index m but performing modulo P instead.

$$l = (n - m)_P$$

Which gives us:

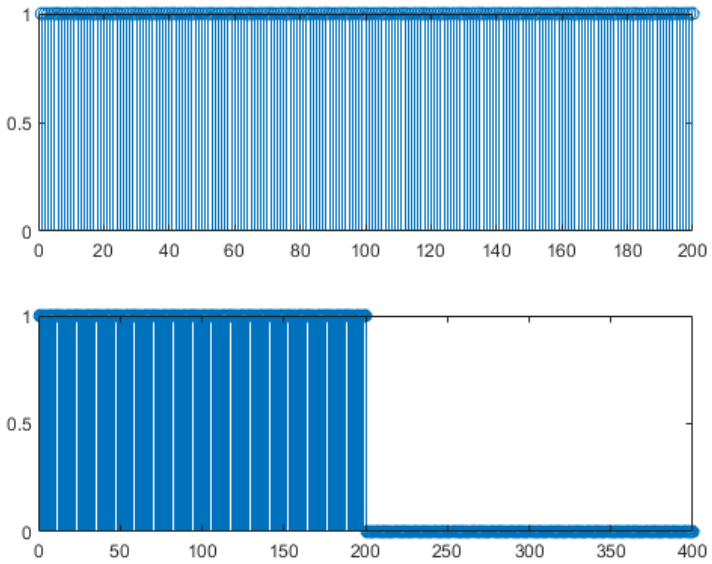
$$y[n] = \sum_{l=0}^{P-1} h[m] x[(n-m)_N]$$

But since the actual region of support for x is well below the support region for the output, the modulo operator does not produce a remainder and circular convolution is not executed. Hence:

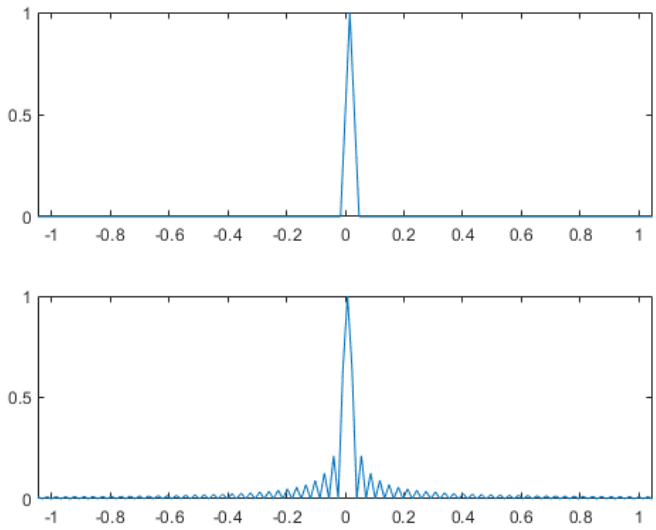
$$y[n] = (h \star x)[n] \quad (Linear\ Convolution)$$

Indirect Consequence of Zero-Padding When Performing The DFT

- Consider two rectangular pulse sequences in the time-domain one with no zero-padding (top graph) and one with zero-padding (bottom graph). The sample support of the original rectangular pulse is from 0 to $N - 1$ where $N = 200$. $M = N = 200$ additional zeros were used for the zero-padding. The total length of the signal is now $P = N + M$ samples



- Performing the DFT on each signal produces the following frequency spectra:



Which shows that the act of zero-padding the sequence introduces side-lobes into the frequency spectra and 'reveals' frequency components that were not there originally.

- The act of *zero-padding* is effectively using a rectangular window in the time-domain which prevents those additional samples from being any other non-zero value.

- In general, we may thing of the input sequence as being:

$$x_e[n] = \left\{x[0], x[1], x[2], \dots, x[N-1], \underbrace{\mathbf{0}, \mathbf{0}, \dots \mathbf{0}}_{P-N\ zeros}\right\} = \{\boldsymbol{x^T}, \mathbf{0^T}\}$$

- The sidelobes are due to an interpolation kernel that arises as a result of the rectangular window effect from the zero-padding in the time-domain.

Proof That Sidelobes Come from Interpolation

Suppose that a discrete signal $x[n]$ with sample support from $n = 0$ to $N-1$ has the following DFT:

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

We saw that its DFT extracted its continuous spectrum, $\hat{x}(\omega)$ if we let $\omega = \frac{2\pi k}{N}$. That is:

$$X[k] = \hat{x}\left(\frac{2\pi k}{N}\right) \quad where \quad \hat{x}(\omega) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

We know that $\hat{x}(\omega)$ is bandlimited to the interval $\omega \in (-\pi, \pi)$. In the same sense, $X[k]$ is also bandlimited in this same interval containing N samples. If we extend the sequence by zero padding such that $x[n] \to x_e[n]$ whose sample support is from $n = 0$ to $P-1$, then we have:

$$X_e[k] = \hat{x}\left(\frac{2\pi k}{P}\right) \quad (\star)$$

That is still bandlimited in $(-\pi, \pi)$ but with a ***higher sample density***. That is, we now have $P > N$ samples in this interval! We can make one conclusion at this point:

*Extending a sequence in the time-domain increases the **density** of samples in the frequency domain. The spectrum, however, is still bandlimited in $(-\pi, \pi)$ just as was the original sequence.*

Our next job is to find a connection between $X_e[k]$ and $X[k]$. We start by realising that if the inverse DFT of $X[k]$ is:

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

Then, we can substitute this into the DTFT representation of $x[n]$:

$$\hat{x}(\omega) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} = \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j.\frac{2\pi k}{N}n} \right) e^{-j\omega n}$$

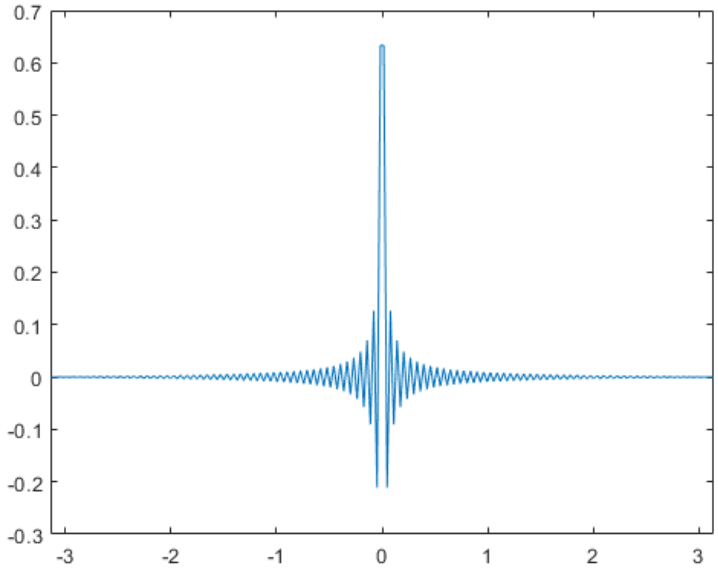
We can swap the sums around such that $X[k]$ is part of the outer summation:

$$\hat{x}(\omega) = \sum_{k=0}^{N-1} X[k] \cdot \underbrace{\frac{1}{N} \sum_{n=0}^{N-1} e^{-j\left(\omega - \frac{2\pi k}{N}\right)n}}_{\substack{Interp.\ Kernel \\ \hat{g}(\omega - 2\pi k/N)}}$$

The interpolation kernel, $\hat{g}(\omega)$ is the summation of the exponential terms in the equation prior.

$$\hat{g}(\omega) = \frac{1}{N}\sum_{n=0}^{N-1} e^{-j\omega n} = \frac{e^{\frac{-j\omega(N+1)}{2}}}{N} \cdot \frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

The magnitude response of the kernel, $|\hat{g}(\omega)|$ is shown in the plot below:



To obtain the frequency spectrum $X_e[k]$, we let $\omega = \frac{2\pi k}{P}$ as shown in Equation $(\star)$. This gives:

$$X_e[k] = \sum_{k=0}^{N-1} X[k] \cdot \hat{g}\left(\frac{2\pi k}{P} - \frac{2\pi k}{N}\right)$$

This result is **crucial** as it shows that we can obtain the DFT of the zero-padded sequence by using a continuous interpolation kernel on the original discrete spectrum $X[k]$.

Overlap-Add

Overlap-Save

Random Processes and Power Spectrum Estimation

Statistics Fundamentals

- A random variable is a measurable function, $X : \Omega \to \mathcal{F}$ that maps a set of possible outcomes $x \in \Omega$ (called the *sample space*) to a measurable field $\mathcal{F}$ which is generally the set of real numbers $\mathbb{R}$.

Discrete VS Continuous Random Variables

- The probability that $X(x)$ takes on a value in $S \subseteq \mathcal{F}$ can be written a conditional probability

$$\mathbb{P}(X \in S) = \mathbb{P}(\{x \in \Omega \mid X(x) \in S\})$$

To understand this, we use a simple example of tossing an unbiased 6-sided die. The sample space, Ω would be:

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

We ask for *"the probability that we roll a number greater than 4"*. In this case, X would be the number we end up with on the die. Rewording as an expression we are asking for:

$$\mathbb{P}(X > 4) = ?$$

This event associated with the random variable generates a new subset $S \subseteq \Omega$ such that:

$$S = \{5, 6\}$$

The probably can be computed as the ratio *cardinality* (size) of S to Ω. That is:

$$\mathbb{P}(X > 4) = \frac{|S|}{|\Omega|} = \frac{2}{6} = \frac{1}{3}$$

Which is as we expected! These kinds of computations can be achieved if and only if X is a **discrete random variable**. However, if X is **continuous**. Then it is **impossible** to compute probabilities. Consider the following problem:

$$Find\ the\ probability\ of\ picking\ the\ value\ 1 \in \mathbb{R}\ from\ the\ set\ of\ real\ numbers$$

This is impossible since the cardinality of $\mathbb{R}$ is essentially infinite! We would always end up computing something like:

$$\mathbb{P}(X = 1) = \frac{|\{1\}|}{|\mathbb{R}|} = \frac{1}{\infty} = 0$$

Definitions for Continuous Random Variables

- To account for this, associated with a continuous random variable X is the ***probability density function (PDF)*** of X is defined as

Probability Density Function (PDF)

$$f_X(x) = \lim_{\Delta \to 0^+} \frac{\mathbb{P}(x < X < x + \Delta)}{\Delta}$$

Where Δ is a very small interval which contains the value of X. $f_X(x)$ basically, computes the probability density at the point x. Associated with X is the ***cumulative distribution function (CDF),*** $F_X(x)$ which directly computes $0 \leq \mathbb{P}(X = x) \leq 1$. Using this we can see that:

$$\begin{aligned} \mathbb{P}(x < X < x + \Delta) &= \mathbb{P}(X < x + \Delta) - \mathbb{P}(X < x) \\ &= F_X(X < x + \Delta) - F_X(X < x) \end{aligned}$$

The CDF can then be expressed as:

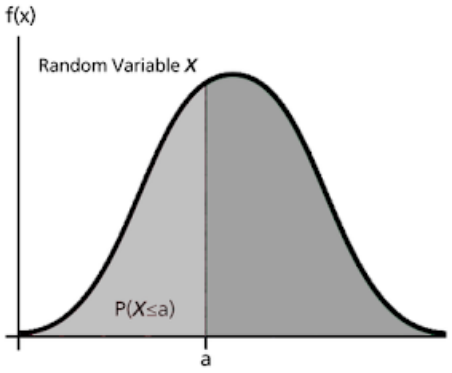
$$f_X(x) = \lim_{\Delta \to 0} \frac{F_X(X < x + \Delta) - F_X(X < x)}{\Delta}$$

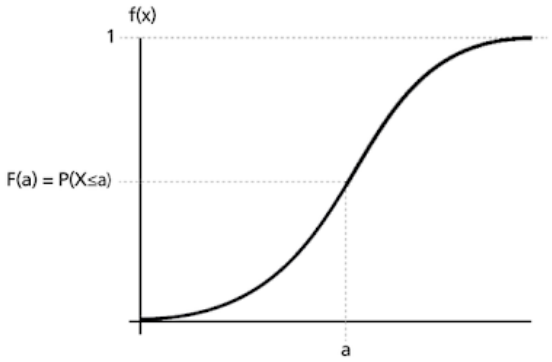
The above is essentially the differentiation of $F_X(x)$ by first principle. Hence, we can say that the cumulative density function is related to the PDF by:

PDF Relation With CDF

$$f_X(x) = \frac{dF_X(x)}{dx} \quad \Longrightarrow \quad F_X(x) = \int_{-\infty}^{x} f_X(x)dx$$

- The image below diagrammatically shows how a continuous probability is computed by either finding the function value $F_X(x)$ OR by integrating $f_X(x)$ in the interval $(-\infty, x]$.





- The **expected value** of a random variable is the mean/average of X:

$$\mathbb{E}[X] = \mu_X = \int_{-\infty}^{\infty} x f_X(x) dx$$

It is also a **linear operator** where for some $\alpha, a, b \in \mathbb{R}$ that is a deterministic constant, we have:

$$\mathbb{E}[\alpha X] = \alpha \mathbb{E}[X] \quad and \quad \mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

- The **variance** of a random variable is the expected value of the squared deviation of a random variable, X from the mean, μ_X

$$\sigma_X^2 = Var(X) = \mathbb{E}[(X - \mu_X)^2] = \mathbb{E}[X^2] - \mu_X^2 = \int_{-\infty}^{\infty} x^2 f_X(x) dx$$

- The **law of the unconscious statistician (LOTUS)** states that if a random variable Y is a function of X such that: $Y = g(X) \Longrightarrow y = g(x)$ then:

$$\mathbb{E}[Y] = \int y f_Y(y) dy = \int g(x) f_X(x) dx$$

- A **random vector** is nothing more than just a sequence of X_k random variables in $\mathbb{R}^n$

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}$$

Each random variable in $\boldsymbol{X}$ is individually subject to the same properties of the expected value and variance. That is:

$$\mathbb{E}[\boldsymbol{X}] = \begin{pmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \\ \vdots \\ \mathbb{E}[X_n] \end{pmatrix} = \begin{pmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_n} \end{pmatrix} \quad similarly\ for\ Var(\boldsymbol{X})$$

- If two random variables X and Y are statistically independent, then their PDFs are separable:

$$f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$$

- If $f_{X|Y}(x, y)$ is the conditional probability distribution of X given that $Y = y$, then:

$$f_{X|Y}(x, y) = \frac{f_{X,Y}(x, y)}{f_Y(y)}$$

- The **correlation** of two random variables, X and Y is the joint expectation of their joint PDF $f_{X,Y}(x,y)$ such that:

$$\mathbb{E}[XY] = \int\int xyf_{X,Y}(x,y)dxdy$$

If X and Y are statistically independent, then their joint PDF is separable. Hence, their expected values must be independent as well! That is:

$$\mathbb{E}[XY] = \int\int xyf_{X,Y}(x,y)dxdy = \int xf_X(x)dx \int yf_Y(y)dy = \mu_X\mu_Y$$

If $\boldsymbol{X}$ is a random vector such that $\boldsymbol{X} = (X_1, X_2, X_3 \dots X_m)^T$, then the correlation is computed as a matrix, $R_{\boldsymbol{X}}$ such that:

$$R_{\boldsymbol{X}} = \begin{pmatrix} \mathbb{E}[X_1X_1] & \mathbb{E}[X_1X_2] & \cdots & \mathbb{E}[X_1X_m] \\ \mathbb{E}[X_2X_1] & \mathbb{E}[X_2X_2] & \cdots & \mathbb{E}[X_2X_m] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[X_mX_1] & \mathbb{E}[X_mX_2] & \dots & \mathbb{E}[X_mX_m] \end{pmatrix}$$

- The **covariance** of two random variables X and Y measures the joint variance between these two random variables. This can be computed as:

$$\begin{aligned} Cov(X,Y) &= \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] \\ &= \mathbb{E}[XY] - \mu_X\mu_Y \end{aligned}$$

If we measure the covariance of a single variable on itself, we have:

$$\begin{aligned} Cov(X,X) &= \mathbb{E}[X^2] - \mu_X^2 \\ &= Var(X) \\ &= \sigma_X^2 \end{aligned}$$

As a result, if $\boldsymbol{X}$ is a random vector, then the covariance is a matrix, $C_{\boldsymbol{X}}$**.** Then, we have:

$$C_{\boldsymbol{X}} = \begin{pmatrix} Cov[X_1X_1] & Cov[X_1X_2] & \cdots & Cov[X_1X_m] \\ Cov[X_2X_1] & Cov[X_2X_2] & \cdots & Cov[X_2X_m] \\ \vdots & \vdots & \ddots & \vdots \\ Cov[X_mX_1] & Cov[X_mX_2] & \dots & Cov[X_mX_m] \end{pmatrix} = \begin{pmatrix} \sigma_{X_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{X_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{X_m}^2 \end{pmatrix}$$

Where if X_i and X_j are statistically independent for all $i \neq j$, then $Cov(X_iX_j) = 0$. However, the diagonal elements compute $Cov(X_iX_i) = \sigma_{X_i}^2$. As a result, we have the fully diagonal matrix shown above.

- A notable property of $R_{\boldsymbol{X}}$ and $C_{\boldsymbol{X}}$ is that they are equivalent to their Hermitian. That is:

$$R_{\boldsymbol{X}} = R_{\boldsymbol{X}}^H \quad \textbf{\textit{and}} \quad C_{\boldsymbol{X}} = C_{\boldsymbol{X}}^H$$

- It is important to note that:

Statistical independence between X and Y implies uncorrelation
BUT
X and Y being uncorrelated does NOT imply statistical independence

- The second statement, however, has an exception and that is when both $\boldsymbol{X}$ and $\boldsymbol{Y}$ have a Gaussian statistical distribution. That is, for some random variable $f_{\boldsymbol{X}}(\boldsymbol{x})$:

$$f_{\boldsymbol{X}}(x) = \frac{1}{\sqrt{(2\pi)^m \|\boldsymbol{C}\|}} e^{-\frac{1}{2}(\boldsymbol{x}-\mu_x)^H C_{\boldsymbol{X}}^{-\mathbf{1}}(\boldsymbol{x}-\mu_x)}$$

Where $m = 1\ or\ 2$ random variables in $\boldsymbol{X}$. If all random variables in $\boldsymbol{X}$ are Gaussian and mutually uncorrelated, then we have $C_{\boldsymbol{X}}$ as being diagonal, then we have:

$$f_{\boldsymbol{X}}(x) = \frac{1}{\sqrt{2\pi\sigma_{X_i}^2}} e^{-\frac{1}{2}\cdot\frac{x_i^2}{\sigma_{X_i}^2}}$$

Random Processes

- Lower-case letters are used to symbolise a **deterministic** process. For instance:

$$x_o[n] = \{0, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, \dots\}$$

If we add some random noise sequence, $v[n]$ to $x_o[n]$, then we have:

$$x[n] = x_o[n] + v[n]$$

Since $v[n]$ is completely random, so is $x[n]$!

- This means $x[n]$ is ONE realisation of the random process, $X[n]$. The random process (denoted with a capital letter like random variables) satisfies the following relationship:

$$X[n] = x_0[n] + V[n]$$

Where $V[n]$ is stochastic and could change drastically at an inconsistent rate (after all, it is generally white noise)

- The statistics (e.g. μ, σ etc.) cannot be accessed for a random process by monitoring it over time despite how *long* we monitor the signal
    - That one long chunk of the random process is only **one** realisation.

- The above process works only if we restrict the random process to a narrower class of 'ergodic processes' instead.

Ergodic Process: *An ergodic process is a random process whose actual statistical properties can be deduced by the statistics by a collection of sufficiently long random sample of the process.*

- Another property of random processes related to ergodicity is the idea of **stationarity**. What does it mean for a random process to be **stationary?**

Stationarity: A random process is stationary if the statistics of the signal are invariant to time-shifts. Formally speaking, if $\boldsymbol{X_{\mathcal{I}}}$ is a random process for a set of indices $\mathcal{I} = \{i_1, i_2, \dots i_k\}$. Then for some index-shift $m \in \mathbb{Z}$, $\boldsymbol{X}$ is stationary if the joint distribution of $\boldsymbol{X_{\mathcal{I}}}$ is the same as $\boldsymbol{X_{\mathcal{I}+m}}$.

- A stationary process $X[n]$ has a constant mean since the PDF of the n^{th} realisation of the random process is the same as the PDF of the of 0th (initial) realisation. That is to say that $f_{X[n]}(x) = f_{X[0]}(x)$. Mathematically, this is shown as:

$$\mu_{X[n]} = \int x \cdot f_{X[n]}(x)\, dx = \int x f_{X[0]}(x) = \mu_{X[0]}$$

- Along with the stationarity of a random process is its **autocorrelation**. We provide the first definition below:

Autocorrelation (Definition 1): Autocorrelation is the correlation of a random process and a delayed version of itself by some delay or *lag, m.* Mathematically, it is the expected value of the joint expected value between the random process and its delayed self.

$$R_{XX}[n, n-m] = \mathbb{E}\big[X[n]X[n-m]\big]$$

- Since $\boldsymbol{X}$ is stationary, we may simplify the above equation to:

$$R_{XX}[n, n-m] = \mathbb{E}\big[X[0]X[-m]\big] = R_{XX}[m]$$

  Where the autocorrelation between two random processes is purely dependent on the lag of the delayed signal.

- Another property of stationary random processes is the idea of **Wide-Sense Stationarity (WSS)**. For a random process to be WSS, it must meet the following:
  - Stationary mean function
  - Stationary autocorrelation function

- A stationary process is WSS. However, a WSS process may NOT necessarily be truly stationary for all time (not all its statistical properties may be time-invariant)

Time Averages, Mean Ergodicity and Autocorrelation

- The *time-average* (m_x) of a realisation of a random process, $x[n]$ is defined as the mean of the realisation for all $n \in \mathbb{Z}$ in the interval $(-\infty, \infty)$.

$$m_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{n=N} x[n]$$

- A signal is *mean-ergodic* if the expected value of the time-average is equivalent to the ensemble (statistical average). That is:

$$\mathbb{E}[\boldsymbol{m_x}] = \boldsymbol{\mu_X} \qquad (for\ mean\ ergodicity)$$

- Where the signal is mean ergodic, the time-autocorrelation sequence for any realisation $x[n]$ of $X[n]$, we have the second definition for autocorrelation:

Autocorrelation (Definition 2):

$$r_{xx}[m] = \lim_{N-\infty} \frac{1}{2N+1} \sum_{n=-N}^{n=N} x[n]x[n-m]$$

- A *correlation-ergodic* random process means its time average **always** gives the same autocorrelation sequence, $r_x[m]$.

Power Spectral Density (PSD) & Cross Correlation

- The power spectral density (PSD), $S_{XX}(\omega)$ of a random process is the DTFT of its autocorrelation statistic, $R_{XX}[m]$:

Power Spectral Density Definition

$$S_{XX}(\omega) = \hat{R}_{XX}(\omega) = \sum_{m} r_{xx}[m]e^{-j\omega m}$$

- Since $R_{XX}[m] = R_{XX}[-m]$ (symmetric), this means $S_{XX}(\omega) \in \mathbb{R}$.

- Suppose $Y[n]$ is the output random process produced by passing $X[n]$ through some LTI system with impulse response $h[n]$. That is:

$$Y[n] = (h \star X)[n]$$

Then, the **cross-correlation** between X and Y is defined as the following can be computed by convolving $R_{XX}[m]$ with $h[k]$

Cross-Correlation:

$$R_{XY}[m] = \sum_{m} h[k] R_{XX}[m-k]$$

$$\Downarrow$$

$$S_{XY}(\omega) = \hat{h}(\omega) S_{XX}(\omega)$$

Like autocorrelation, cross correlation measures the similarity between two random processes X and Y by some lag $m \in \mathbb{Z}$. However, at $m = 0$, the cross correlation may not be 100% (as expected with autocorrelation).

- Cross-correlation can be used to measure the impulse response of the LTI system, $h[n]$. If we let $X[n]$ be white noise, we know that:

$$S_{XX}(\omega) = 1$$

Hence, if the output of the system is $Y[n]$, the cross correlation between the output and input sequence would be:

$$S_{XY}(\omega) = \hat{h}(\omega) S_{XX}(\omega) = \hat{h}(\omega) \times 1 = \hat{h}(\omega)$$

The inverse DTFT can be used on $S_{XY}(\omega)$ to compute $h[n]$.

Estimating Autocorrelation and Power Spectra

- An estimate for $R_{xx}[m]$ can computed by the following time-average for a realisation $r_{xx}[m]$:

$$r_{xx}^{unbiased}[m] = \frac{1}{N-|m|} \left[\sum_{n=\max\{0,m\}}^{N-1-m+\max\{0,m\}} x[n]x[n-m] \right] \qquad |m| < N$$

Where the $\max\{0, m\}$ *is used to ensure sum stays in the range* $0 \leq n \leq N$

- The above estimate is unbiased since time-average is taken over the entire length N. However, this poses a critical issue

    - As $m \to \pm N$, the 'average' will only involve one sample. That is, for instance:

$$r_{xx}[N-1] = r_{xx}[1-N] = x[0]x[N-1]$$

    Hence, the variance will markedly increase as $m \to \pm N$. This has a devastating effect on the computed PSD and its usefulness.

- To make the estimate more conservative (which is *biased*), we can average independent of of the lag m. That is:

$$r_{xx}^{biased}[m] = \frac{1}{N}\left[\sum_{n=\max\{0,m\}}^{N-1-m+\max\{0,m\}} x[n]x[n-m]\right] \qquad |m| < N$$

However, note that from the first equation, we have:

$$\sum_{n=\max\{0,m\}}^{N-1-m+\max\{0,m\}} x[n]x[n-m] = r_{xx}^{unbiased}[m] \times (N-|m|)$$

Substituting this into the summation in the equation for the biased estimate, we have:

$$r_{xx}^{biased}[m] = \left(\frac{N-|m|}{N}\right) r_{xx}^{unbiased}[m] = w_B[m] \cdot r_{xx}^{unbiased}[m]$$

Where $w_B[m] = \frac{N-|m|}{N} = 1 - \frac{|m|}{N}$ is the ***Bartlett*** window and is attained by sampling its continuous version defined as the following (note that $L = N$ in our case):

$$w_B(t) = \begin{cases} 1 - \dfrac{|t|}{J} & |t| < J \\ 0 & |t| \geq J \end{cases}$$

- There is a price to pay; the mean of the autocorrelation is now biased as a result of us trying to minimise the variance as the lag nears the end of the sequence.

The Periodogram

- The periodogram is essentially the *estimated PSD* from the *biased autocorrelation sequence* that uses the Bartlett window. That is:

$$S_{xx}^{biased}(\omega) = \sum_m w_B[m] r_{xx}[m] e^{-j\omega n} = \sum_m r_{xx}^{biased}[m] e^{-j\omega n}$$

- The biased spectral density is computationally derived from the circular DFT of a realisation of a random process (circular since S_{XX} convolved with w_B is generally not bandlimited).

- To see how this is the case, let $x[n]$ be, in actuality, an infinite sequence. If we window it using a rectangular window $x_o[n]$ with sample support $0 \leq n < M$, we have:

$$x_o[n] = x[n]w[n]$$

Naturally, when performing a PSD calculation on MATLAB, we are implicitly applying this window (since we are taking an N length portion of an otherwise infinite random process).

Then, the biased autocorrelation estimate would be:

$$r_{xx}^{biased}[m] = \frac{1}{N}\sum_{n=-\infty}^{\infty} x_o[m]x_o[n-m] = \frac{1}{N}\sum_{n=-\infty}^{\infty} x_o[m]\tilde{x}_o[m-n]$$

Where $\tilde{x}_o[m] = x_o[-m] \Longrightarrow \mathcal{F}\{\tilde{x}_o\} = \hat{x}_o^*(\omega)$. Then:

$$S_{xx}^{biased}(\omega) = \frac{1}{N} \cdot \hat{x}_o(\omega)\hat{x}_o^*(\omega) = \frac{1}{N}|\hat{x}_o(\omega)|^2$$

And noting that computationally, we really are performing a DFT, we require that:

$$X_o[k] = \hat{x}_o(\omega)|_{\omega=\frac{2\pi}{N}mod(2\pi)}$$

Then, we can obtain the biased PSD by computing the circular DFT on our realisation of the random process, take the magnitude-squared of the DFT result and then average it by the length of the signal N! That is:

$$S_{xx}^{biased}(\omega)|_{\omega=\frac{2\pi}{N}mod(2\pi)} = \frac{1}{N}|X_o[k]|^2$$

The above is an **estimate** of the true PSD of the random process. This crude estimate is what we call a **periodogram**.

Problems with The Periodogram

- The variance of the periodogram $S_{XX}^{biased}(\omega)$ is very large even though the biased estimate $r_{xx}^{biased}[m]$ was enforced to minimise it in the first place.

- Contrary to intuition, increasing the sample support of the random process does NOT reduce this variance!

- Hence, the estimate PSD via the DFT is unreliable and its reliability is NOT improved by increasing the number of samples. **The core issue is using ALL the available samples N to compute the PSD.**

- A way to improve the reliability of the PSD estimate is to split the realisation into P contiguous chunks, $x_p[n]$ of length $L < N$ such that $L = NP$. Then, the periodograms of each chunk, $S_P(\omega)$ is computed such that we can calculate the **average periodogram**. That is:

$$S_{xx}^{biased}(\omega) = \frac{1}{P}\sum_{p=0}^{P-1} S_p(\omega) = \frac{1}{P}\sum_{p=0}^{P-1}\frac{1}{L}|X_k[p]|^2 \qquad 0 \le k \le L$$

Blackman-Tukey (BT) Method

- The BT method uses windowing to improve the reliability of PSD estimates. Let $w_{BT}[m]$ be the autocorrelation window. Then:

$$S_{xx}^{BT}(\omega) = \sum_{m=-L}^{m=L}\left(w_{BT}[m]\cdot r_{xx}^{biased}[m]\right)\cdot e^{-j\omega n}$$

  Here, the variance decreases by $1/N$ while the spectral resolution remains fixed assuming the length of the chunk and the BT window, L remains fixed.

$$Var[S_{XX}^{BT}(\omega)] = S_{XX}^2(\omega)\cdot\left[\frac{1}{N}\sum_{m=-L}^{L} w_{BT}^2[m]\right]$$

Introduction to Linear Prediction

- Suppose we have a random process $X[n]$ with realisations $x[n]$ of length N. Suppose we want to compute a prediction for $x[n]$ which we denote as $\bar{x}[n]$ such that:

$$\bar{x}[n] = a_1 x[n-1] + a_2 x[n-2] + \cdots + a_N x[n-N]$$

Where $a_k \in \mathbb{R}$ are weights for each sample of $\bar{x}[n]$

- The goal is to **optimise** the weights $a_k \in \mathbb{R}$ or *Linear Prediction (LP) Coefficients* such that we minimise the error, $e[n]$ between the **actual** realisation and the predicted one:

$$e[n] = x[n] - \bar{x}[n] = x[n] - \sum_{k=1}^{N} a_k x[n-k]$$

In terms of the random processes with which they were derived from:

$$E[n] = X[n] - \bar{X}[n] = X[n] - \sum_{k=1}^{N} a_k X[n-k]$$

- The error $E[n]$ is called the ***innovations process*** since it represents new information in $x[n]$ which could not have been anticipated.

- Like the Parkes-McClellan method for designing FIR and IIR filters, optimisation is best done by considering and minimising the squared error, ε:

$$E^2[n] = (X[n] - \bar{X}[n])^2 = \left(X[n] - \sum_{k=1}^{N} a_k X[n-k] \right)^2$$

And assuming stationarity, we have:

$$\varepsilon[n] = \mathbb{E}\left[E^2[n]\right] = \mathbb{E}[(X[n] - \bar{X}[n])^2] = \mathbb{E}\left[\left(X[n] - \sum_{k=1}^{N} a_k X[n-k] \right)^2 \right]$$

Differentiating both sides with respect to the p^{th} weight, a_p and setting the derivative to zero, we aim to solve the equation:

$$\frac{\partial \varepsilon}{\partial a_p} = 0$$

Applying this to our equation, we have:

$$\frac{\partial \varepsilon}{\partial a_p} = \mathbb{E}\left[2\left(X[n] - \sum_{k=1}^{N} a_k X[n-k] \right) \cdot X[n-p] \right] = 0$$

Noting that 2 is constant and is not affected by $\mathbb{E}$, expanding the brackets, we have:

$$\mathbb{E}\big[X[n]X[n-p]\big] - \mathbb{E}\left[\sum_{k=1}^{N} a_k X[n-k]X[n-p]\right] = 0$$

$$\mathbb{E}\big[X[n]X[n-p]\big] - \sum_{k=1}^{N} a_k \mathbb{E}\big[X[n-k]X[n-p]\big] = 0$$

$$R_{XX}[p] - \sum_{k=1}^{N} a_k R_{XX}[p-k] = 0$$

Hence, our final equation looks like:

$$\sum_{k=1}^{N} a_k R_{XX}[p-k] = R_{XX}[p]$$

Where the $X[n-k]$ component introduces a fixed shift of k to the autocorrelation sequence

The summation on the left-hand side are called the ***normal equations*** for reasons specified later.

The left-hand side is also essentially the dot product between the vector $\mathbf{a} = (a_1, a_2 \dots a_N)^T$ and the N^{th} order *auto-correlation matrix* $R_{X_{0:N}}$ comprising of elements $R_{XX}[p-k]$. The dot product yields the elements of the vector $\mathbf{r} = (R_{XX}[1], R_{XX}[2], R_{XX}[3] \dots R_{XX}[N])^{\mathrm{T}}$

$$R_{\boldsymbol{X}_{0:N}}\mathbf{a} = \mathbf{r}$$

Solving for the weights $\mathbf{a}$, we have:

$$\mathbf{a} = R^{-1}_{\boldsymbol{X}_{0:N}}\mathbf{r}$$

Properties of the Autocorrelation Matrix

- The auto-correlation matrix is symmetric $\Longrightarrow R^T_{\boldsymbol{X}_{0:N}} = R_{\boldsymbol{X}_{0:N}}$ or Hermitian if the signal was complex in the first place.
- The main diagonal elements of the auto-correlation matrix are the same value $R_{XX}[0]$. This matrix exhibits a *Toeplitz* structure because of this.
- Other properties are in Taubman's notes (which I don't really understand and cannot be bothered understanding)

Normal Equations & Orthogonality Principle for Random Variables

- The term *normal* is used in linear algebra to mean vectors which are perpendicular or **orthogonal** to each other. Fundamentally, this is when their dot product is equal to 0.

$$\langle \boldsymbol{i}, \boldsymbol{j} \rangle = 0$$

- Orthogonality is observed stochastically in a similar way. Performing the derivative on ε in the previous section yielded the following result (the multiple of 2 is omitted)

$$\mathbb{E}\left[\left(X[n] - \sum_{k=1}^{N} a_k X[n-k]\right) \cdot X[n-p]\right] = 0$$

  But if the error above is really $E[n]$, then, we can express this simply as:

$$\mathbb{E}\big[E[n]X[n-p]\big] = 0 \quad (\star)$$

- The error $E[n]$ **must be zero-mean** (Gaussian) and **uncorrelated** with $X[n]$ in order for the above to be satisfied.

- Notice that equation $(\star)$ is very much like the dot product of orthogonal vectors. For any two random processes A and B, we define the inner product as:

$$\langle A, B \rangle \triangleq \mathbb{E}[AB]$$

  Then equation $(\star)$ can be re-expressed as:

$$\langle E[n], X[n-p] \rangle = 0$$

  That is, **if the optimal coefficients are found, $\boldsymbol{E[n]}$ MUST be orthogonal to all predictor inputs $\boldsymbol{X[n-1], X[n-2] \ldots X[n-N]}$**. This is known as the **orthogonality principle**.

Whiteness of Innovations Process

- The n^{th} instance of the linear prediction error, $E[n]$ can be expressed as:

$$E[n] = X[n] - \sum_{k=1}^{N} a_k X[n-k]$$

Then the next $(n-m)^{th}$ instance of the error can be expressed as:

$$E[n-m] = X[n-m] - \sum_{k=1}^{N} a_k X[n-k-m]$$

From Equation $(\star)$ we saw that $\mathbb{E}\big[E[n]X[n-m]\big] = 0$. That is, the error at $n \in \mathbb{Z}$ is uncorrelated with the random variable at some other instance $(n-m) \in \mathbb{Z}$.

It should clearly follow that if $E[n]$ is uncorrelated with $X[n-m]$, then it should also be uncorrelated with the prediction error at $n-m$ which is $E[n-m]$. That is:

$$\mathbb{E}\big[E[n]E[n-m]\big] = 0 \qquad for\ m \neq 0$$

Where the ONLY instance where the above is non-zero is when $m = 0$ (i.e. no lag between the errors → maximum autocorrelation) but zero otherwise. The **only** thing that is consistent with this behaviour is **white noise** where it is completely unpredictable from one sample to the next!

- Hence, the autocorrelation sequence for $E[n]$ is given by:

$$R_{EE}[n] = \mathbb{E}\big[E[n]E[n-m]\big] = \begin{cases} \sigma_E^2 & m = 0 \\ 0 & otherwise \end{cases}$$

Which means that the power spectral density of the linear prediction error is simply:

$$S_{EE}(\omega) = \sigma_E^2$$

Which is effectively constant. This is consistent of our understanding of the power of white noise which should be constant for all frequencies in $\omega \in (-\pi, \pi)$.

- The implied impulse response of a filter $h[n]$ with a PSD characteristic of $S_{EE}(\omega)$ would be:

$$h[n] = \begin{cases} 1 & n = 0 \\ -a_n & 1 \leq n \leq N \\ 0 & otherwise \end{cases}$$

The application of the filter onto $X[n]$ produces the linear prediction error! This can be seen by simply re-writing our summation expression for the error:

$$E[n] = X[n] - \sum_{k=1}^{N} a_k X[n-k] = \sum_{k=1}^{N} h[k]X[n-k] = (h \star X)[n]$$

Yule-Walker Algorithm for Computing $R_{XX}[m]$

- The unbiased estimation saw that the computed autocorrelation, $R_{XX}^{unbiased}$ had an increasing variance due to averaging by $\frac{1}{N-|m|}$ as $m \to N$ where N is the length of the entire data sequence and m is the lag of the delayed version of X.

- The biased estimation, R_{XX}^{biased} sought to reduce this variance, the average was fixed by a factor of $\frac{1}{N}$

- An **alternative** is to divide the entire data vector into K consecutive **overlapping blocks** of M samples. The minimisation problem to compute these blocks now becomes:

$$\frac{1}{K-N}\sum_{n=N}^{K-1}\left(x[n] - \sum_{k=1}^{N} a_k x[n-k]\right)^2$$

- Computing the derivative and setting to zero yields the same resulting normal equations with an *approximate* autocorrelation matrix, $\hat{R}$ that is symmetric if $K > N$ (i.e. more overlapping blocks than actual samples). However, $\hat{R}$ has less structure → not Toeplitz any further → inverting becomes less trivial. However, the matrix equation to compute still holds:

$$\boldsymbol{a} = \hat{R}^{-1}\boldsymbol{r}$$

  The approximate autocorrelation matrix, $\hat{R}$ can be computed via the following summation independent $\mathbf{a}$.

$$R = \sum_{k=1}^{N}\left(\sum_{n=N}^{K-1} x[n-p]x[n-k]\right) \qquad for\ 1 \le p \le N$$

Signal Detection

- Signals measured in the physical world will inevitably have noise superimposed. This makes the signal stochastic with some statistical variations. Such signals can be written in the form:

$$x[n] = s[n] + w[n]$$

  Where $s[n]$ is the deterministic component of the signal while $w[n]$ is the noise added on top of the deterministic component.

- The goal is to estimate certain parameters of the deterministic signal such as the frequency, amplitude, phase etc. from the stochastic signal. We exploit the probability density function (PDF) of the noise and ultimately $x[n]$ to do this.

Likelihood Function ($\mathcal{L}$)

- This is achieved via what we call the ***likelihood function***. Suppose some we some random process X with a PDF $f_X(x)$ parameterised by some parameter θ (e.g. frequency, amplitude).

- Let a realisation, $\boldsymbol{x}$ with support $n = 0 \dots N-1$ be data samples from the random process.

- Let the joint PDF between each sample in the realisation (i.e. $x_0, x_1 \dots x_{N-1}$) be denoted by $f_X(\boldsymbol{x}|\theta)$ where $\boldsymbol{x} = [x_0, x_1, x_2, \dots x_{n-1}]^T$.

- Then the likelihood function, denoted by $\mathcal{L}(\theta|\boldsymbol{x})$ that is parameterised by θ is equal to the PDF of the realisation. That is:

$$\mathcal{L}(\theta|\boldsymbol{x}) = f_X(\boldsymbol{x}|\theta)$$

- The main aim is to determine an estimate for θ denoted as $\hat{\theta}$ that **maximises** the likelihood function (i.e. the PDF of the random process). We can write this mathematically as:

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta|\boldsymbol{x})$$

Log-Likelihood ($\mathcal{LL}$) & Maximum Likelihood Estimation (MLE)

- Alternatively, owing the **monotonic** properties of natural logarithmic functions, taking the natural log of $\mathcal{L}$ gives us the log-likelihood ($\mathcal{LL}$) function. With PDFs involving factors and terms with higher powers, the $\mathcal{LL}$ will make computations comparatively more convenient.

$$\mathcal{LL}(\theta|\boldsymbol{x}) = \log_e \mathcal{L}(\theta|\boldsymbol{x}) = \ln \mathcal{L}(\theta|\boldsymbol{x})$$

- The maximisation objective can be written as:

$$\hat{\theta} = \arg\max_{\theta} \mathcal{LL}(\theta|\boldsymbol{x})$$

- The maximisation objective is met by computing the derivative of $\mathcal{LL}$ with respect to the parameter of interest, setting to zero, and then solving for $\theta = \hat{\theta}$. This process is called **maximum likelihood estimation (MLE)**.

$$\frac{\partial}{\partial\theta}\mathcal{LL}(\theta|\boldsymbol{x}) = 0$$

- Caveat: The solution to the above may not have a closed-form solution or may have many maxima and minima that may make it difficult to find the maximum likelihood.

Example 1 – Single Parameter MLE

Consider a complex signal $\boldsymbol{x} \in \mathbb{C}^N$ from a random process $\boldsymbol{X}$ with support $n = 0 \dots N-1$ comprised of an equal length deterministic signal $\boldsymbol{s} \in \mathbb{C}^N$ and complex additive white gaussian noise (AWGN), $w[n]$ with mean 0 and variance σ^2. Suppose the deterministic signal was $s[n] = e^{j2\pi fn}$ with amplitude of 1. Then:

$$x[n] = s[n] + w[n] = e^{j2\pi fn} + w[n]$$

We want to find an estimate for $\hat{f}$ that maximises the likelihood function of the random variable with respect to f. We start by getting an expression for the PDF of the white noise which is Gaussian which we know has a real and imaginary part which are real:

$$w[n] = w_R[n] + w_I[n] \qquad (where\ \boldsymbol{w}_R, \boldsymbol{w}_I \in \mathbb{R}^N)$$

This means that the real part of the noise has variance $\frac{\sigma^2}{2}$ and the imaginary part also has variance $\frac{\sigma^2}{2}$. If the real and imaginary components are statistically independent, we can say that the PDF of the white noise is the joint PDF of the real and imaginary parts[1].

$$f_X(w[n]) = f_X(w_R[n]) \cdot f_X(w_I[n])$$

For any real random process, $\boldsymbol{A}$ that has a Gaussian (normal) distribution, its characteristic PDF is given by:

$$f_{\boldsymbol{A}}(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{a^2}{2\sigma^2}}$$

This means:

$$\begin{aligned} f_X(w[n]) &= f_X(w_R[n]) \cdot f_X(w_I[n]) \\ &= \left(\frac{1}{\sqrt{2\pi\frac{\sigma^2}{2}}} e^{-\frac{w_R^2[n]}{2\frac{\sigma^2}{2}}}\right) \cdot \left(\frac{1}{\sqrt{2\pi\frac{\sigma^2}{2}}} e^{-\frac{w_I^2[n]}{2\frac{\sigma^2}{2}}}\right) \\ &= \frac{1}{\pi\sigma^2} e^{-\frac{1}{\sigma^2}(w_R^2 + w_I^2)} = \frac{1}{\pi\sigma^2} e^{-\frac{|w[n]|^2}{\sigma^2}} \qquad (Where\ |w[n]|^2 = w_R^2 + w_I^2) \end{aligned}$$

[1] https://www.casact.org/pubs/forum/15fforum/Halliwell_Complex.pdf

However, we know that $w[n] = x[n] - e^{j2\pi fn}$. Hence, we can express the PDF of $\boldsymbol{X}$ in terms of the parameter of interest, f as:

$$f_X(w[n]) = f_X\left(x[n] - e^{j2\pi fn}\right) = \frac{1}{\pi\sigma^2} e^{-\frac{|x[n]-e^{j2\pi fn}|^2}{\sigma^2}}$$

For brevity, we let $f_X\left(x[n] - e^{j2\pi fn}\right) = f_X(x[n])$. Hence, the joint PDF of all samples in the realisation in terms of the parameter f is given by:

$$\begin{aligned} f_{\boldsymbol{X}}(\boldsymbol{x}|f) &= \prod_{n=0}^{N-1} \frac{1}{\pi\sigma^2} e^{-\frac{|x[n]-e^{j2\pi fn}|^2}{\sigma^2}} \\ &= \left(\frac{1}{\pi\sigma^2}\right)^N e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-e^{j2\pi fn}|^2} \\ &= \frac{e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-e^{j2\pi fn}|^2}}{(\pi\sigma^2)^N} \\ &= \mathcal{L}(f|\boldsymbol{x}) \end{aligned}$$

Taking the log-likelihood of the PDF, we have:

$$\begin{aligned} \mathcal{LL}(f|\boldsymbol{x}) &= \ln\left[e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-e^{j2\pi fn}|^2}\right] - \ln[(\pi\sigma^2)^N] \\ &= \underbrace{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-e^{j2\pi fn}|^2}_{Dependent\ on\ f} - \underbrace{\ln[(\pi\sigma^2)^N]}_{Constant} \end{aligned}$$

The maximisation objective with respect to f can be written as:

$$\hat{f} = \arg\max_f \mathcal{LL}\,(f|\boldsymbol{x})$$

We can see the summation is the only part of the log-likelihood function that is dependent on f. Since the $\ln[(\pi\sigma^2)^N]$ is just constant and $\frac{-1}{2\sigma^2}$ is just a scaling factor that makes $\mathcal{LL}$ contain a global maxima, the maximisation objective to be achieved can be rephrased as a minimisation objective of the summation alone which is a least-squares problem with some global minima.

$$\hat{f} = \arg\min_f \sum_{n=0}^{N-1}|x[n]-e^{j2\pi fn}|^2 = \arg\min_f \sum_{n=0}^{N-1} L(f)$$

Computationally, we may find a more efficient minimisation strategy by considering the expansion of least squares. Let $(\cdot)^*$ denote the complex conjugate. Then:

$$\begin{aligned} L(f) &= |x[n]-e^{j2\pi fn}|^2 \\ &= \left(x[n]-e^{j2\pi fn}\right)\left(x[n]-e^{j2\pi fn}\right)^* \\ &= \left(x[n]-e^{j2\pi fn}\right)\left(x^*[n]-e^{-j2\pi fn}\right) \\ &= |x[n]|^2 - x[n]e^{-2\pi fn} - x^*[n]e^{j2\pi fn} + 1 \\ &= |x[n]|^2 - \left\{x[n]e^{-2\pi fn} + \left(x[n]e^{-2\pi fn}\right)^*\right\} + 1 \\ &= |x[n]|^2 - Re\left\{x[n]e^{-2\pi fn}\right\} + 1 \end{aligned}$$

Hence, the summation to be minimised is then:

$$I(f) = \sum_{n=0}^{N-1} L(f)$$
$$= \sum_{n=0}^{N-1} |x[n]|^2 - \underbrace{Re\{x[n]e^{-2\pi fn}\}}_{\substack{Frequency \\ Dependent}} + 1$$

The minimum of the sum is achieved by maximising the component of sum dependent on f which is the middle term. That is:

$$\hat{f} = \arg\max_f \sum_{n=0}^{N-1} Re\{x[n]e^{-2\pi fn}\} = \arg\max_f Re\left[\sum_{n=0}^{N-1} x[n]e^{-j2\pi fn}\right]$$

The estimate $\hat{f}$ is found by maximising the **real part** of the DFT (note this is not the periodogram).

Case 2 – Multi Parameter MLE

Now suppose the deterministic part of the signal is modulated by some amplitude factor, A. That is:

$$x[n] = Ae^{j2\pi fn} + w[n]$$

Our goal is to compute estimates $\hat{A}$ and $\hat{f}$ that maximises the log-likelihood $\mathcal{LL}(\boldsymbol{\theta}|\boldsymbol{x})$ of the random process with which $\boldsymbol{x}$ was realised from where $\boldsymbol{\theta} = \left[\hat{A}, \hat{f}\right]^T$. We start from the PDF of the noise term which can be re-phrased in terms of $x[n]$ and the deterministic component:

$$f_{\boldsymbol{X}}(w[n]) = f_X\left(x[n] - Ae^{j2\pi fn}\right)$$
$$\implies f_{\boldsymbol{X}}(x[n]) = \frac{1}{\pi\sigma^2} e^{-\frac{|x[n]-Ae^{j2\pi fn}|^2}{\sigma^2}}$$

Then, the joint PDF of the entire realisation with respect to the parameter vector $\boldsymbol{\theta}$ is the likelihood function of the random process:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\theta})$$
$$= \prod_{n=0}^{N-1} \frac{1}{\pi\sigma^2} e^{-\frac{|x[n]-Ae^{j2\pi fn}|^2}{\sigma^2}}$$
$$= \left(\frac{1}{\pi\sigma^2}\right)^N e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2}$$
$$= \frac{e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2}}{(\pi\sigma^2)^N}$$

The log-likelihood function can, once more, be found by taking the natural log on both sides:

$$\mathcal{LL}(\boldsymbol{\theta}|\boldsymbol{x}) = \ln\left[\frac{e^{-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2}}{(\pi\sigma^2)^N}\right] = -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2 - \ln[(\pi\sigma^2)^N]$$

We can either solve for $\hat{A}$ or $\hat{f}$ subject to the maximisation problem of the log-likelihood function to be solved. However, it may be easier to solve for $\hat{A}$ first and back-substituting to solve for $\hat{f}$. Then, considering the portion of the

$$\hat{A} = \arg\max_A -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2$$

The squared magnitude in the summation is quadratic and, hence, differentiable. Differentiating the sum with respect to A and setting to 0, we have:

$$\frac{\partial}{\partial A}\left[-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-Ae^{j2\pi fn}|^2\right] = 0$$

$$\sum_{n=0}^{N-1}\frac{\partial}{\partial A}|x[n]-Ae^{j2\pi fn}|^2 = 0$$

$$\sum_{n=0}^{N-1}2\left(x[n]-Ae^{j2\pi fn}\right)e^{j2\pi fn} = 0$$

Expanding and rearranging gives:

$$\sum_{n=0}^{N-1}x[n]e^{j2\pi fn} = A\sum_{n=0}^{N-1}\underbrace{e^{j2\pi fn}e^{-j2\pi fn}}_{=1}$$

$$\sum_{n=0}^{N-1}x[n]e^{j2\pi fn} = AN$$

$$\therefore \hat{A} = \frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{j2\pi fn}$$

Noting this, we can substitute $\hat{A}$ into our log-likelihood function to find some an expression only in terms of f that we can compute an estimate for.

$$\mathcal{LL}\left(\boldsymbol{\theta}|\boldsymbol{x},\hat{A}\right) = -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]-\hat{A}e^{j2\pi fn}|^2 - \lambda \qquad (\lambda = \ln[(\pi\sigma^2)^N] \in \mathbb{R})$$

Noting once more that for any $z \in \mathbb{C}$ that $zz^* = |z|^2$, we have:

$$\mathcal{LL}\left(\boldsymbol{\theta}|\boldsymbol{x},\hat{A}\right) = -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(x[n]-\hat{A}e^{j2\pi fn}\right)\left(x^*[n]-\hat{A}^*e^{-j2\pi fn}\right) - \lambda$$

The expansion within the summation is identical to Case 1 with the single parameter MLE. We can use that result to simplify:

$$\mathcal{LL}(\boldsymbol{\theta}|\boldsymbol{x},\hat{A}) = -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]|^2 - \frac{1}{\sigma^2}\sum_{n=0}^{N-1}Re\{\hat{A}x[n]e^{-2\pi fn}\} + \frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left|\hat{A}\right|^2$$

Substituting $\hat{A}$, we have:

$$\begin{aligned}\mathcal{LL}(\boldsymbol{\theta}|\boldsymbol{x},\hat{A}) &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]|^2 - \frac{1}{\sigma^2}\sum_{n=0}^{N-1}Re\left\{\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{j2\pi fn}\,x[n]e^{-2\pi fn}\right\} + \frac{1}{\sigma^2}\sum_{n=0}^{N-1}|\hat{A}|^2 \\ &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]|^2 - \frac{1}{N\sigma^2}\sum_{n=0}^{N-1}Re\{|x[n]|^2\} + \frac{1}{\sigma^2}\sum_{n=0}^{N-1}|\hat{A}|^2\end{aligned}$$

But since $|x[n]|^2 \in \mathbb{R}$, the $Re\{\cdot\}$ can be omitted. Hence:

$$\begin{aligned}\mathcal{LL}(\boldsymbol{\theta}|\boldsymbol{x},\hat{A}) &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}|x[n]|^2 - \frac{1}{N\sigma^2}\sum_{n=0}^{N-1}|x[n]|^2 + \frac{1}{\sigma^2}\sum_{n=0}^{N-1}|\hat{A}|^2 \\ &= -\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(1+\frac{1}{N}\right)|x[n]|^2 + \frac{N}{\sigma^2}\left|\hat{A}\right|^2\end{aligned}$$

And because we know that $\hat{A} = \hat{A}(f)$, the last term of the log-likelihood is the only term that needs to be optimised to compute the maximum of $\mathcal{LL}$. That is, to maximise $\mathcal{LL}$, we need to maximise $\left|\hat{A}\right|^2$ that has a global maximum. Our objective then is:

$$\begin{aligned}\hat{f} &= \arg\max_f \left|\hat{A}\right|^2 \\ &= \arg\max_f \left|\frac{1}{N}\sum_{n=0}^{N-1}x[n]e^{j2\pi fn}\right|^2\end{aligned}$$

This is effectively computing for the frequency corresponding to the global maximum of the periodogram of the signal which unfortunately does not have a closed form solution. Seeking for the maxima may prove difficult as the periodogram will be a sinc-function with many maxima and minima.

Maximum Likelihood Detector (MLD)

We cannot know with absolute certainty whether our received signal is just noise or contains the deterministic signal of interest. We can set this up as a hypothesis test comprising the null hypothesis (H_0) and the alternative hypothesis (H_1) for some realisation, $\boldsymbol{x}$:

$$\begin{aligned} H_0&: \quad \boldsymbol{x} \sim f_0(\boldsymbol{x}) \\ H_1&: \quad \boldsymbol{x} \sim f_1(\boldsymbol{x}) \end{aligned}$$

Where in each hypothesis, the signal is characterised by some PDF. To determine which hypothesis we choose to accept, we compute the **likelihood ratio (LR)** which is computed as:

$$LR(\boldsymbol{x}) = \frac{f_1(\boldsymbol{x})}{f_0(\boldsymbol{x})}$$

We then establish an appropriately set threshold $\lambda \in \mathbb{R}$. If $LR(\boldsymbol{x}) > \lambda$, we choose the alternative hypothesis, H_1. Otherwise, we choose the null hypothesis, H_0. This can be written succinctly as:

$$LR(\boldsymbol{x}) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

The log-likelihood ratio, $LLR(\boldsymbol{x})$ will be computed more often. Taking natural log of both sides, the test to perform becomes:

$$LLR(\boldsymbol{x}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

Where $\gamma = \ln(\lambda)$ and $LLR(\boldsymbol{x}) = \ln f_1(\boldsymbol{x}) - \ln f_0(\boldsymbol{x})$. We now consider two scenarios where MLD is used directly with no parameters whose estimates need to be computed and one where this is a necessity.

Case 1 – MLD With KNOWN Signal of Interest

Consider two scenarios where the signal received may either be just noise or noise along with the deterministic signal to be detected. We consider these signals to be real in these cases.

$$\begin{aligned} H_0&: \quad x[n] = w[n] \\ H_1&: \quad x[n] = s[n] + w[n] \end{aligned}$$

The PDF of the signal in the null hypothesis is the PDF of the white noise alone. Assume the noise is Gaussian with 0 mean and variance σ^2. Then:

$$f_0(\boldsymbol{x}) = \prod_{n=0}^{N-1} f_X(w[n]) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{w^2[n]}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{\sum_{n=0}^{N-1} -\frac{x^2[n]}{2\sigma^2}}$$

For the alternative, note that $w[n] = x[n] - s[n]$. Then:

$$f_1(\boldsymbol{x}) = \prod_{n=0}^{N-1} f_X(w[n]) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{w^2[n]}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{\sum_{n=0}^{N-1} -\frac{(x[n]-s[n])^2}{2\sigma^2}}$$

Taking the ratio of both PDFs, we can obtain the likelihood ratio function:

$$LR(\boldsymbol{x}) = \frac{f_1(\boldsymbol{x})}{f_0(\boldsymbol{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

Of course, the log-likelihood ratio, is then computed as:

$$LLR(\boldsymbol{x}) = \ln f_1(\boldsymbol{x}) - \ln f_0(\boldsymbol{x}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \qquad (\gamma = \ln \lambda)$$

We want an expression for $LLR(\boldsymbol{x})$ in terms of our signal. We simplify below:

$$\begin{aligned} LLR(\boldsymbol{x}) &= \left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-s[n])^2 - \ln\left(\sqrt{2\pi}\sigma\right)^N\right] - \left[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n] - \ln\left(\sqrt{2\pi}\sigma\right)^N\right] \\ &= -\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-s[n])^2 + \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n] \\ &= \frac{1}{2\sigma^2}\left\{\sum_{n=0}^{N-1}x^2[n] - \sum_{n=0}^{N-1}(x[n]-s[n])^2\right\} \\ &= \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(2x[n]s[n] - s^2[n]) \end{aligned}$$

Our ratio test can now be written as:

$$LLR(\boldsymbol{x}) = \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}2x[n]s[n] - \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}s^2[n] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

Because the signal of interest (SOI) is known beforehand, we can move the 2nd summation to the RHS to give:

$$\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}2x[n]s[n] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma + \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}s^2[n]$$

$$\underbrace{\sum_{n=0}^{N-1}x[n]s[n]}_{y} \underset{H_0}{\overset{H_1}{\gtrless}} \underbrace{\sigma^2\gamma + \frac{1}{2}\sum_{n=0}^{N-1}s^2[n]}_{Let\ this\ is\ \rho}$$

Our ratio test can now be written as:

$$y \underset{H_0}{\overset{H_1}{\gtrless}} \rho$$

We compute the statistic $y = \sum_{n=0}^{N-1}x[n]s[n]$ and compare the value to the threshold ρ defined as:

$$\rho = \sigma^2\gamma + \frac{1}{2}\sum_{n=0}^{N-1}s^2[n]$$

If $y > \rho$, we accept the alternative hypothesis that the signal $s[n]$ has been transmitted/received! Else, we say that the signal transmitted/received is just noise.

Case 2 – With UNKNOWN Signal of Interest

Suppose now that our signal is modulated by some amplitude factor, $\alpha \in \mathbb{R}$ that is unknown. This means our signal of interest $y[n] = \alpha s[n]$ is effectively unknown. The hypotheses are:

$$\begin{aligned} H_0&: \quad x[n] = w[n] \\ H_1&: \quad x[m] = \alpha s[n] + w[n] \end{aligned}$$

As with the previous case, the PDF of the signal in the null hypothesis is:

$$f_0(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{\sum_{n=0}^{N-1} -\frac{x^2[n]}{2\sigma^2}}$$

The PDF for the signal with the deterministic component is now:

$$f_1(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{\sum_{n=0}^{N-1} -\frac{(x[n]-\alpha s[n])^2}{2\sigma^2}}$$

In order to compute $LLR(\boldsymbol{x})$, the amplitude of the deterministic component needs to be known. We can use maximum likelihood estimation to do this prior to performing the likelihood ratio test. The likelihood function for the signal in the alternative hypothesis parameterised by α is:

$$\mathcal{L}(\alpha|\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{\sum_{n=0}^{N-1} -\frac{(x[n]-\alpha s[n])^2}{2\sigma^2}}$$

Then, the log-likelihood function will be:

$$\mathcal{LL}(\alpha|\boldsymbol{x}) = \frac{-1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-\alpha s[n])^2 - \ln\left(\sqrt{2\pi}\sigma\right)^N$$

The maximisation of $\mathcal{LL}(\alpha|\boldsymbol{x})$ can be found by minimising the summation in the first term (without the $-\frac{1}{2\sigma^2}$ factor. Then, our objective minimisation problem to find the estimate $\hat{\alpha}$ is:

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{n=0}^{N-1}(x[n]-\alpha s[n])^2 = \arg\min_{\alpha} I(\alpha)$$

Compute the derivative of $I(\alpha)$, set to 0 and find the estimate $\alpha = \hat{\alpha}$ that minimises $I(\alpha)$:

$$\frac{\partial I}{\partial \alpha} = \sum_n 2(x[n]-\alpha s[n])\cdot s[n] = 0$$

$$\implies \hat{\alpha} = \frac{\sum_n s[n]x[n]}{\sum_n s^2[n]}$$

We can now compute the log-likelihood ratio given our computed estimate $\hat{\alpha}$:

$$LLR(\boldsymbol{x}, \hat{\alpha}) = -\frac{1}{2\sigma^2}\left[\sum_n (x[n]-\hat{\alpha}s[n])^2 - \sum_n x^2[n]\right]$$

Simplifying the quadratic in the first summation gives us the following LLR for the ratio test:

$$LLR(\boldsymbol{x}, \hat{\alpha}) = -\frac{1}{2\sigma^2}\left[\hat{\alpha}\sum_n 2x[n]s[n] - \hat{\alpha}^2\sum_n s[n]\right] \underset{H_0}{\overset{H_1}{\gtrless}} \lambda$$

Multiplying both sides by $2\sigma^2$ and letting $\gamma = 2\sigma^2\lambda$, we have:

$$\hat{\alpha}^2 \sum_n s[n] - \hat{\alpha} \sum_n 2x[n]s[n] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

Substituting our expression for $\hat{\alpha}$, we have:

$$\hat{\alpha} = \left(\frac{\sum_n s[n]x[n]}{\sum_n s^2[n]}\right)^2 \sum_n s[n] - \left(\frac{\sum_n s[n]x[n]}{\sum_n s^2[n]}\right) \sum_n 2x[n]s[n] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

Simplifying yields:

$$\frac{(\sum_n x[n]s[n])^2}{\sum_n s^2[n]} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

The detector constructs the statistic y given by:

$$y = \frac{(\sum_n x[n]s[n])^2}{\sum_n s^2[n]}$$

And compares it to the threshold $\gamma = 2\sigma^2\lambda$. Once again, if the computed statistic is over the threshold, we accept the alternative hypothesis that the desired signal of interest was transmitted/received. Else, we accept the null hypothesis where the signal transmitted/received was just noise.

Wiener Filtering

- Wiener filtering allows us to compute the *minimum mean-squared error (MMSE)* prediction $\tilde{y}[n]$ out of the outcomes $y[n]$ from a random process $Y[n]$ whose values come from $x[n]$ from the random process $X[n]$.

- The objective function, J is given by:

$$J = \mathbb{E}[(Y[n] - \bar{Y}[n])^2] = \mathbb{E}\left[\left(Y[n] - \sum_{n=0}^{N} a_k X[n-k]\right)^2\right]$$

  Which very much looks like the objective function for linear prediction for $Y[n] = X[n+1]$.

- The error process $E[n] = Y[n] - \bar{Y}[n]$ should satisfy the orthogonality principle when minimising the objective function (via direct differentiation).

$$0 = \mathbb{E}\big[E[n]X[n-p]\big] = \mathbb{E}\left[\left(Y[n] - \sum_{k=0}^{N} a_k X[n-k]\right)X[n-p]\right]$$
$$= R_{YX}[p] - \sum_{k=0}^{N} a_k R_{XX}[p-k]$$

- Representing the above as a matrix equation gives us:

$$\underbrace{\begin{bmatrix} R_{XX}[0] & R_{XX}[1] & \cdots & R_{XX}[N] \\ \vdots & R_{XX}[0] & \ddots & \vdots \\ R_{XX}[N] & R_{XX}[N-1] & \cdots & R_{XX}[0] \end{bmatrix}}_{\boldsymbol{R_X}} \cdot \underbrace{\begin{bmatrix} a_0 \\ \vdots \\ a_N \end{bmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{bmatrix} R_{YX}[0] \\ \vdots \\ R_{YX}[N] \end{bmatrix}}_{\boldsymbol{r}}$$

$$\boldsymbol{R_X a} = \boldsymbol{r}$$

  The coefficients in $\boldsymbol{a}$ represent the N^{th} order Wiener FIR filter coefficients applied to the source sequence $x[n]$ to produce the predicted output $\bar{y}[n]$.

Unconstrained Wiener Filter ($h_\infty[k]$)

- If we do not restrict $0 \leq k \leq N$, and allow the filter to have infinite support, our objective function is:

$$J = \mathbb{E}\left[\left(Y[n] - \sum_{k=0}^{N} h_\infty[k] \cdot X[n-k]\right)^2\right]$$

- Using the orthogonality principle for the error process, we have:

$$\begin{aligned} 0 &= R_{YX}[p] - \sum_{k=0}^{N} h_\infty[k] \cdot R_{XX}[p-k] \\ &= R_{YX}[p] - (\boldsymbol{h}_\infty \star \boldsymbol{R}_{XX})[p] \end{aligned}$$

Taking the Fourier transform on both sides gives converts $R_{YX}[p]$ and $R_{XX}[p]$ to their cross power spectral density and power spectral density respectively.

$$0 = S_{YX}(\omega) - \hat{h}_\infty(\omega) S_{XX}(\omega) \quad \Longrightarrow \quad \boldsymbol{\hat{h}_\infty(\omega) = \frac{S_{YX}(\omega)}{S_{XX}(\omega)}}$$

This result tells us two things:

1. If we perform $h_\infty[n] = \mathcal{F}^{-1}\{\hat{h}_\infty(\omega)\}$, the impulse response will have infinite support and will be non-causal which means it cannot be practically realised.
2. The filter places a weight on each frequency based on the ratio of the cross power spectral density between $\boldsymbol{Y}$ and $\boldsymbol{X}$ and the power spectral density of $\boldsymbol{X}$.

- It is also instructive to compute the power spectral density of the error process from its autocorrelation sequence:

$$R_{EE}[m] = \mathbb{E}\big[E[m]E[n-m]\big]$$

The power spectral density can be calculated to be:

$$S_{EE}(\omega) = S_{YY}(\omega) \cdot \left(1 - \frac{|S_{YX}(\omega)|^2}{S_{YY}(\omega)S_{XX}(\omega)}\right)$$

Where the ratio $0 \leq \frac{|S_{YX}(\omega)|^2}{S_{YY}(\omega)S_{XX}(\omega)} \leq 1$ depending on the correlation between X and Y at a particular frequency. Here a ratio value of 0 denotes complete uncorrelation while a value of 1 means the process are perfectly correlated.

Relation to FIR Filter Design

- The matrix equation for the design of the FIR filter has a resemblance to the matrix equation produced from the FIR least-squares (FIR-LS) algorithm (Parkes-McClellan).

- For some desired filter response $\hat{h}_d(\omega)$ and the actual response $\hat{h}(\omega)$, their mean-squared error can be written as an objective function to be minimised:

$$\varepsilon = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\hat{\rho}(\omega)|^2 \big|\hat{h}(\omega) - \hat{h}_d(\omega)\big|^2 d\omega$$

- The solution to the FIR-LS objective function (by direct differentiation with respect to the coefficients of $\hat{h}_d(\omega)$) is given by the matrix equation:

$$\underbrace{\begin{bmatrix} r[0] & r[1] & \cdots & r[N] \\ \vdots & r[0] & \ddots & \vdots \\ r[N] & r[N-1] & \cdots & r[0] \end{bmatrix}}_{\boldsymbol{R_X}} \cdot \underbrace{\begin{bmatrix} a_0 \\ \vdots \\ a_N \end{bmatrix}}_{\boldsymbol{a}} = \underbrace{\begin{bmatrix} d[0] \\ \vdots \\ d[N] \end{bmatrix}}_{\boldsymbol{r}}$$

  Where $\hat{r}(\omega) = |\hat{\rho}(\omega)|^2$ and $\hat{d}(\omega) = |\hat{\rho}(\omega)|^2 \cdot \hat{h}_d(\omega)$

- The equivalence between the Wiener filter and FIR-LS optimised filter coefficients is achieved by setting:

  - $r[n] = R_{XX}[n] \implies |\hat{\rho}(\omega)|^2 = S_{XX}(\omega)$
  - $d[n] = R_{YX}[n] \implies |\hat{\rho}(\omega)|^2 \cdot \hat{h}_d(\omega) = S_{YX}(\omega) \implies \hat{h}_d(\omega) = \frac{S_{YX}(\omega)}{|\hat{\rho}(\omega)|^2} = \hat{h}_\infty(\omega)$

- This assignment gives rise to these important results:

  - The constrained Wiener filter $h[n]$ is obtained by minimizing a weighted squared-error between $\hat{h}_d(\omega)$ and $\hat{h}_d(\omega)$.
  - The equivalence of the direct Wiener filter design and the filter design via FIR-LS is by setting $|\hat{\rho}(\omega)|^2 = S_{XX}(\omega)$. The value of the weighting function will be the smallest where $X[n]$ has very little power!
  - However, the FIR-LS implicitly applies a rectangular window which is not possible in this case since $S_{XX}(\omega)$ is never uniform unless it is white noise (which in our case is not). **Hence, it remains better to design the Wiener filter directly.**

The LMS Algorithm

- Wiener filter design requires that we know the correlation sequences $R_{YX}[m]$ and $R_{XX}[m]$ beforehand. In a real-time scenario, this is not possible.

- Another alternative is to use the ***least-mean squares (LMS)*** **algorithm** which updates the Wiener filter coefficients as they arrive in real-time.

- This is achieved by using a ***gradient descent*** strategy to reach the minimum of the objective function incrementally.

- Consider the objective function, J as a function of the filter coefficients $\boldsymbol{a} = [a_0, a_1, \ldots, a_N]^T$

$$J = \mathbb{E}\left[\left(Y[n] - \sum_{k=0}^{N} a_k X[n-k]\right)^2\right]$$

  The idea is computing a vector of derivatives, ΔJ to optimise all coefficients $a_j \in \boldsymbol{a}$**.** That is:

$$\nabla J = \begin{pmatrix} \frac{\partial J}{\partial a_0} \\ \frac{\partial J}{\partial a_1} \\ \vdots \\ \frac{\partial J}{\partial a_N} \end{pmatrix}$$

- For some initial values for $\boldsymbol{a}$, the gradient vector $\nabla J(\boldsymbol{a})$ points in the direction of **steepest ascent**. This means $-\nabla J(\boldsymbol{a})$ will point in the direction of **steepest descent.**

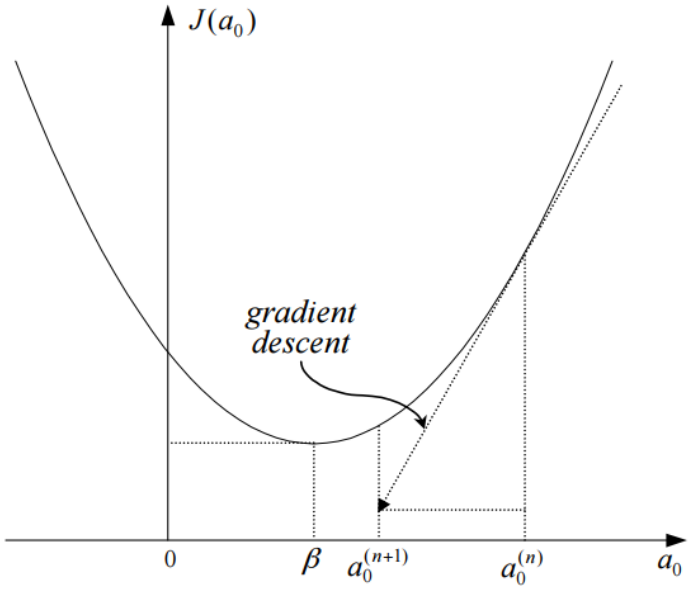
- The goal is to update the values of the parameters in $\boldsymbol{a}$ such ∇J reaches the global minimum of the objective function! When this happens, we have reached the best-case values for the Wiener filter coefficients.

- Let $\delta \in \mathbb{R}$ be a hyperparameter (a value initialised prior) called the **learning rate** which is used to update the coefficients iteratively (computationally via a for-loop). This is usually a small value, something like $\delta = 0.01$ or 0.001.

- If we let $\boldsymbol{a}^{(n)}$ be the n^{th} (or *current*) parameter values and $\boldsymbol{a}^{(n+1)}$ be the $(n+1)^{th}$ (or *next/updated)* parameter values, then:

$$\boldsymbol{a}^{(n+1)} = \boldsymbol{a}^{(n)} - \delta \Delta J$$

The following figure shows a graph of a parabolic cost function J for one of the Wiener coefficient parameters $a_0 \in \boldsymbol{a}$ with some gradient vector that is updated to reach the global minimum $a_0 = \beta \in \mathbb{R}$.



- For more complicated cost functions with many local minima and maxima, a bad initialisation for $a_j \in \boldsymbol{a}$ may mean we will never reach the minimum we desire.

- We can derive the gradient vector for our Wiener filter coefficients by directly differentiating the objective function as we have done before:

$$\frac{\partial J}{\partial a_p} = -2 \cdot \mathbb{E}\left[\left(\underbrace{Y[n] - \sum_{k=0}^{N} a_k X[n-k]}_{E[n]}\right) \cdot X[n-p]\right] = -2 \cdot \mathbb{E}\big[E[n] \cdot X[n-p]\big]$$

Our gradient vector is then:

$$\nabla J = -2 \cdot \begin{pmatrix} \mathbb{E}\big[E[n] \cdot X[n-0]\big] \\ \mathbb{E}\big[E[n] \cdot X[n-1]\big] \\ \vdots \\ \mathbb{E}\big[E[n] \cdot X[n-N]\big] \end{pmatrix} = -2 \cdot \begin{pmatrix} R_{YX}[0] \\ R_{YX}[1] \\ \vdots \\ R_{YX}[N] \end{pmatrix}$$

- If we want to do this in real-time or computationally without having to compute the cross correlation with knowledge of the processes beforehand, we make the crude approximation with the instantaneous error and input signal.

$$\nabla J^{(n)} = -2 \cdot \begin{pmatrix} e[n] \cdot x[n-0] \\ e[n] \cdot x[n-1] \\ \vdots \\ e[n] \cdot x[n-N] \end{pmatrix}$$

Where $e[n]$ and $x[n]$ are the **instantaneous** error and input signal sequences at time n.

- While the approximation computes a largely inaccurate gradient vector, if we use a small learning rate δ, the errors average out and we will reach the cost function minimum and the coefficient values will remain stable. The consequence is a slow learning rate.

- Using a large value for δ will increase the learning rate that tracks the statistical variations more rapidly but results in unstable filter coefficients that will bounce around their optimum values.