EEL 6537 – Spectral Estimation

Jian Li

Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611, USA

“ Spectral Estimation is ⋯ an Art ”

Petre Stoica

“ I hear, I forget;

I see, I remember;

I do, I understand.”

A Chinese Philosopher.

What is Spectral Estimation?

From a finite record of a stationary data sequence, estimate how the total power is distributed over frequencies , or more practically, over narrow spectral bands (frequency bins).

Spectral Estimation Methods:

• Classical (Nonparametric) Methods

Ex. Pass the data through a set of band-pass filters and measure the filter output powers.

• Parametric (Modern) Approaches

Ex. Model the data as a sum of a few damped sinusoids and estimate their parameters.

Trade-Offs: (Robustness vs. Accuracy)

• Parametric Methods may offer better estimates if data closely agrees with assumed model.

• Otherwise, Nonparametric Methods may be better.

Some Applications of Spectral Estimation

- Speech
  - Formant estimation (for speech recognition)
  - Speech coding or compression
- Radar and Sonar
  - Source localization with sensor arrays
  - Synthetic aperture radar imaging and feature extraction
- Electromagnetics
  - Resonant frequencies of a cavity
- Communications
  - Code-timing estimation in DS-CDMA systems

REVIEW OF DSP FUNDAMENTALS

Continuous-Time Signals

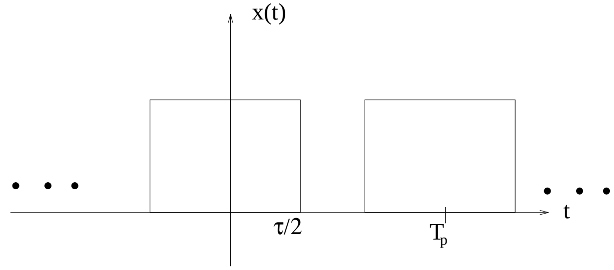
- Periodic signals

$$x(t) = x(t + T_p)$$

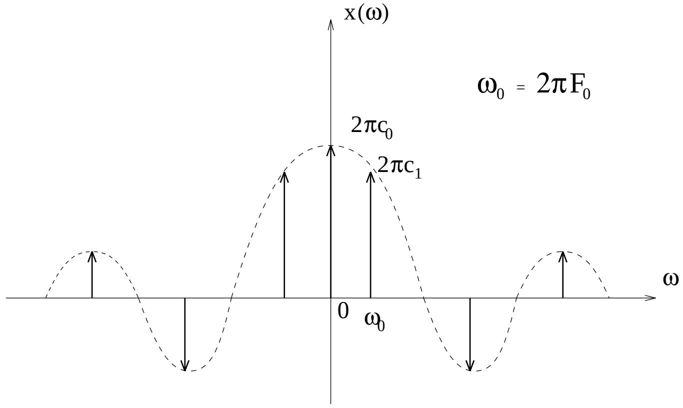
Fourier Series:

$$\begin{aligned} x(t) &= \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_o t} \\ c_k &= \frac{1}{T_p} \int_{T_p} x(t) e^{-j2\pi k F_o t} dt, \\ F_o &= \frac{1}{T_p}. \end{aligned}$$

Ex.

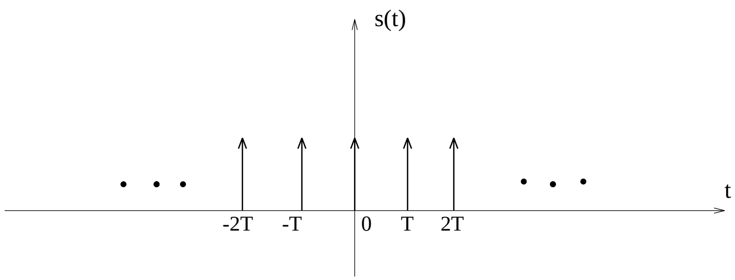


$$e^{j\omega_o t} \overset{FT}{\longleftrightarrow} 2\pi\delta(\omega - \omega_o)$$

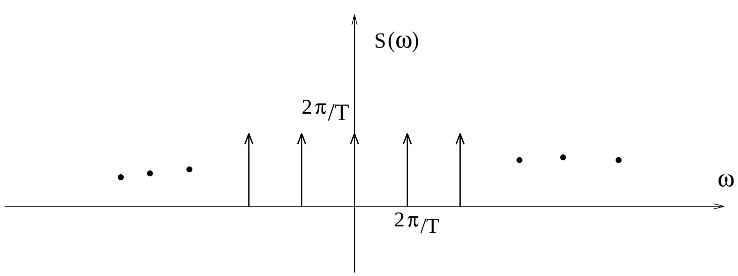


Ex.

$s(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT)$

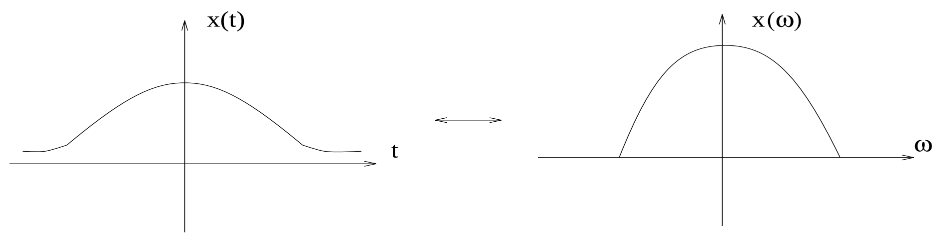


$C_k = \frac{1}{T}$ for all k

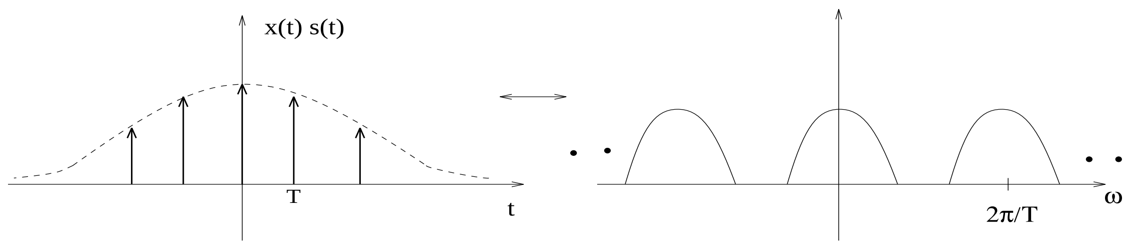


Remark: **Periodic Signals $\longleftrightarrow$ Discrete Spectra.**

- Discrete signals



Ex:

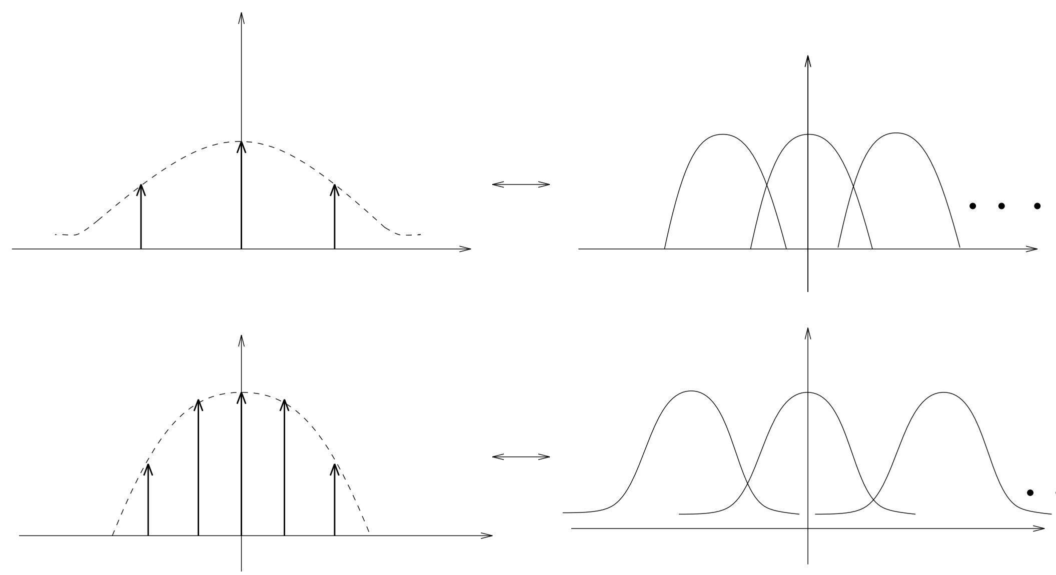


Remark: **Discrete Signals $\longleftrightarrow$ Periodic Spectra.**

Discrete Periodic Signals $\longleftrightarrow$ Periodic Discrete Spectra.

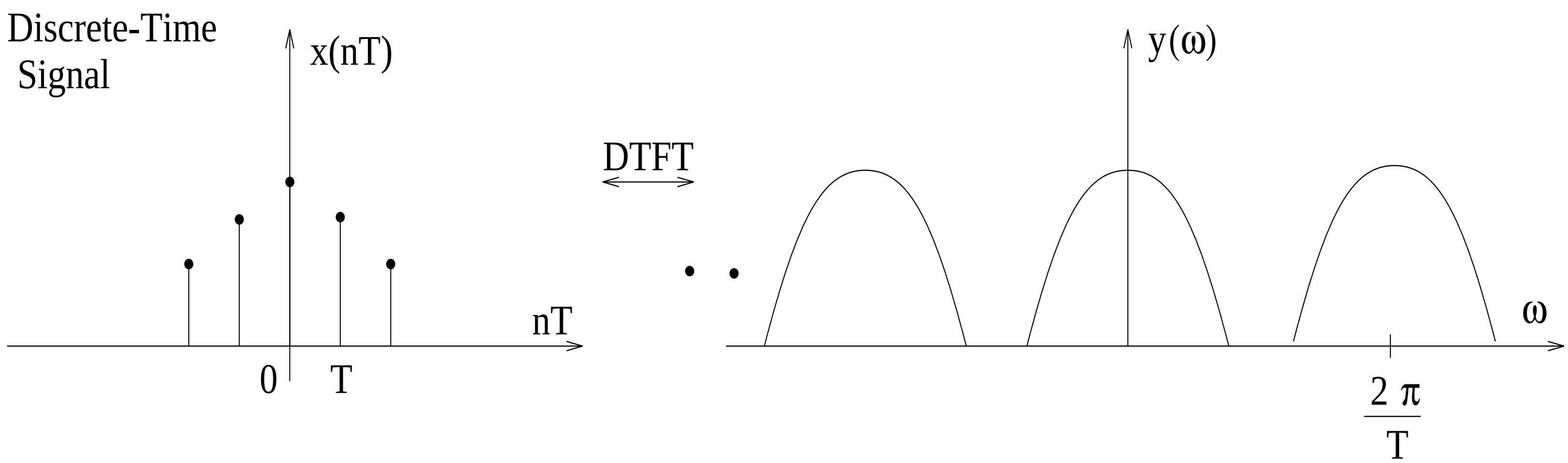
Aliasing Problem:

Ex.



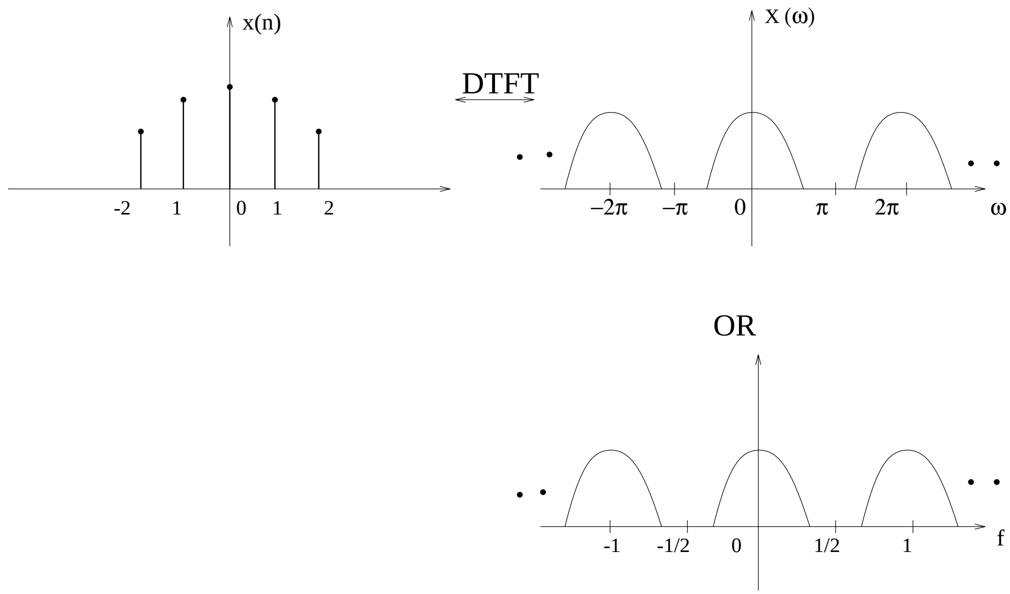
* Fourier Transform (Continuous - Time vs. Discrete-Time)

$$
\begin{aligned}
Let \quad y(t) = x(t)s(t) &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \\
CTFT: \quad Y(\omega) &= \int_{-\infty}^{\infty} y(t)e^{-j\omega t}dt \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)e^{-j\omega t}dt \\
&= \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT} \\
DTFT: \quad Y(\omega) &= \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}
\end{aligned}
$$



Remarks: Discrete-Time Fourier Transform (DTFT) is the same as Continuous-Time Fourier Transform (CTFT) with $x(nT)\ \delta(t - nT)$ replaced by $x(nT)$ and $\int$ replaced by $\sum$ (easy for computers).

For simplicity, we drop T.

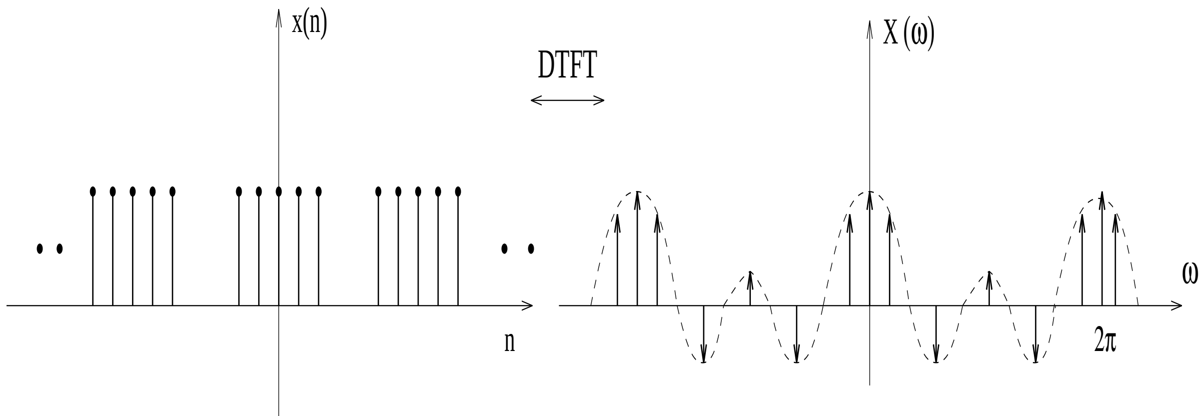


$$DTFT \quad Pair: \begin{cases} X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \\ x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \end{cases}$$

Remark: For DTFT, we also have:

$$\text{Discrete Periodic Signals} \overset{DTFT}{\longleftrightarrow} \text{Periodic Discrete Spectra.}$$

Ex.

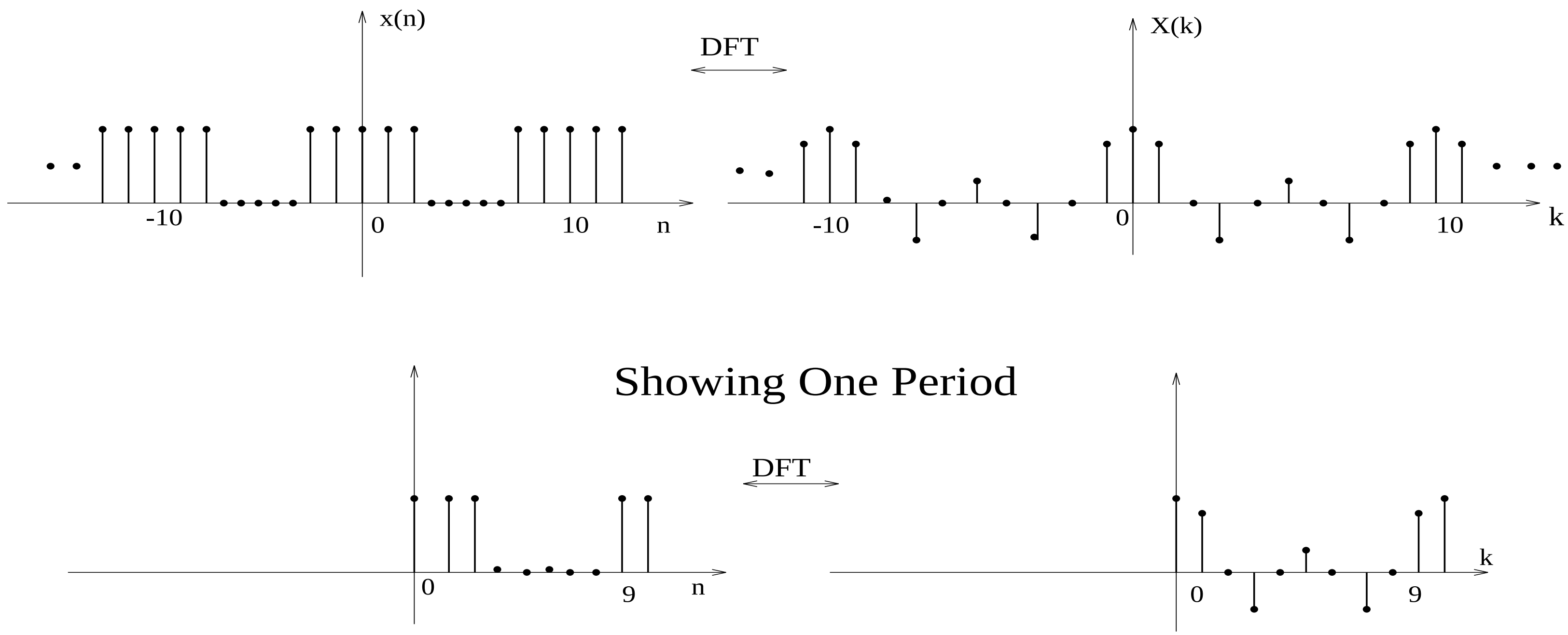


Note The Aliasing

When $x(n+N) = x(n)$,

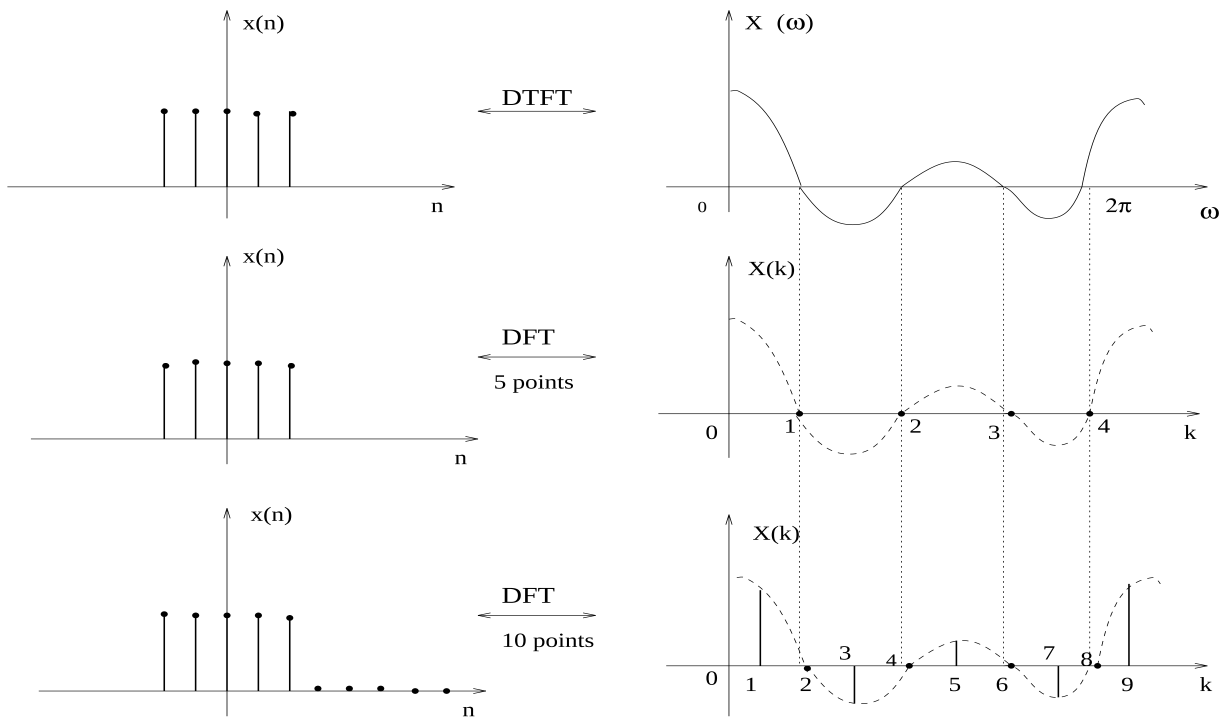
$$DFT \quad Pair: \begin{cases} x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi k \frac{n}{N}}, \\ X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}} \end{cases}$$

Ex. Note the Aliasing



Remarks: For periodic sequences, DFT and DTFT yield similar spectra. IDFT (Inverse DFT) is the same as IDTFT (inverse DTFT) with $X\left(\frac{2\pi k}{N}\right)\delta\left(\omega - \frac{2\pi k}{N}\right)$ replaced by $X(k)$ and $\int$ replaced by $\sum$ (easy for computers).

Effects of Zero-Padding:



Remark:• The more zeroes padded, the closer X(k) is to X(ω).

• X(k) is a sampled version of X(ω) for finite duration sequences.

Z-Transform

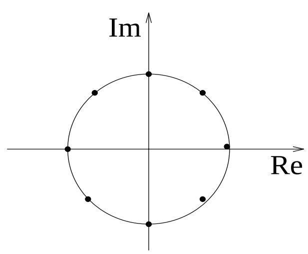
$$\begin{cases} X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \\ x(n) = \frac{1}{2\pi j} \int_c X(z) z^{n-1} dz \end{cases}$$

For finite duration $x(n)$,

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

The DFT X(k) is related to X(z) as follows:

$$X(k) = X(z)\big|_{z=e^{j\frac{2\pi}{N}k}}.$$



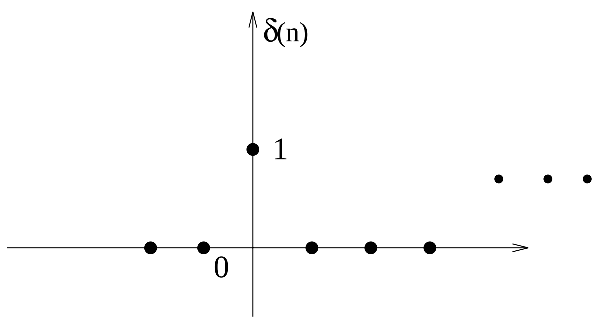
(X(k) evenly sampled on the unit circle of the z-plane)

Linear Time-Invariant (LTI) Systems.

- N^{th} order difference equation:

$$\sum_{k=0}^{N-1} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k)$$

- Impulse Response:



$$h(n) = y(n)|_{x(n)=\delta(n)}$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}.$$

- Bounded-Input Bounded-Output (BIBO) Stability:

All poles of $H(z)$ are inside the unit circle for a causal system (where h(n)=0, n$<$ 0).

- FIR Filter: N=0.
- IIR Filter: N$>$0.
- Minimum Phase: All poles and zeroes of H(z) are inside the unit circle.

ENERGY AND POWER SPECTRAL DENSITIES

- **Energy Spectral Density of Deterministic Signals.**

Finite Energy Signal if

$$0 < \sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty$$

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

Parseval's Energy Theorem:

$$\begin{cases} \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega)d\omega, \\ S(\omega) = |X(\omega)|^2 \end{cases}$$

Remark: $|X(\omega)|^2$ "measures" the length of orthogonal projection of $\{x(n)\}$ onto basis sequence $\left\{e^{-j\omega n}\right\}$, $\omega \in [-\pi, \pi]$.

Let $\rho(k) = \sum_{n=-\infty}^{\infty} x(n)x^*(n-k)$.

$$\begin{aligned}\sum_{k=-\infty}^{\infty} \rho(k)e^{-j\omega k} &= \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(n)x^*(n-k)e^{-j\omega n}e^{j\omega(n-k)} \\ &= \left[\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right]\left[\sum_{s=-\infty}^{\infty} x(s)e^{-j\omega s}\right]^* \\ &= |X(\omega)|^2 = S(\omega).\end{aligned}$$

Remark: $S(\omega)$ is the DTFT of the "autocorrelation" of finite energy sequence $\{x(n)\}$.

- **Power Spectral Density (PSD) of Random Signals.**

Let $\{x(n)\}$ be wide-sense stationary (WSS) sequence with

$$E[x(n)] = 0.$$

$$r(k) = E[x(n)x^*(n-k)].$$

Properties of autocorrelation function $r(k)$.

- $r(k) = r^*(-k)$.
- $r(0) \geq |r(k)|$, for all k
- $0 \leq r(0) =$ average power of $x(n)$.

Def: <u>$\mathbf{A}$ is positive semidefinite if $\mathbf{z}^H \mathbf{A} \mathbf{z} \geq 0$ for any $\mathbf{z}$.</u>

$(\mathbf{z}^H = (\mathbf{z}^T)^*$ Hermitian transpose).

Let

$$\mathbf{A} = \begin{bmatrix} r(0) & r(k) \\ r^*(k) & r(0) \end{bmatrix} = E\left\{ \begin{bmatrix} x(n) \\ x(n-k) \end{bmatrix} \begin{bmatrix} x^*(n) & x^*(n-k) \end{bmatrix} \right\}$$

Obviously, $\mathbf{A}$ is positive semidefinite.

Then all eigenvalues of $\mathbf{A}$ are ≥ 0.

$\Rightarrow$ determinant of $\mathbf{A} \geq 0$.

$\Rightarrow r^2(0) - |r(k)|^2 \geq 0$.

Covariance matrix:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(m-2) & r(m-1) \\ r^*(1) & r(0) & \ddots & \ddots & r(m-2) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r^*(m-1) & r^*(m-2) & \cdots & r^*(1) & r(0) \end{bmatrix}$$

- It is easy to show that $\mathbf{R}$ is positive semidefinite.
- $\mathbf{R}$ is also Toeplitz.
- Since $\mathbf{R} = \mathbf{R}^H$, $\mathbf{R}$ is Hermitian.

- Eigendecomposition of $\mathbf{R}$

$\mathbf{R} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^H$,

where $\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$

($\mathbf{U}$ is unitary matrix whose columns are eigenvectors of $\mathbf{R}$)

$\boldsymbol{\Sigma}$= diag(λ_1, ..., λ_m),

(λ_i are the eigenvalues of $\mathbf{R}$, real, and ≥ 0).

First Definition of PSD:

$$P(\omega) = \sum_{k=-\infty}^{\infty} r(k) e^{-j\omega k}$$

$$r(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) e^{j\omega k} d\omega$$

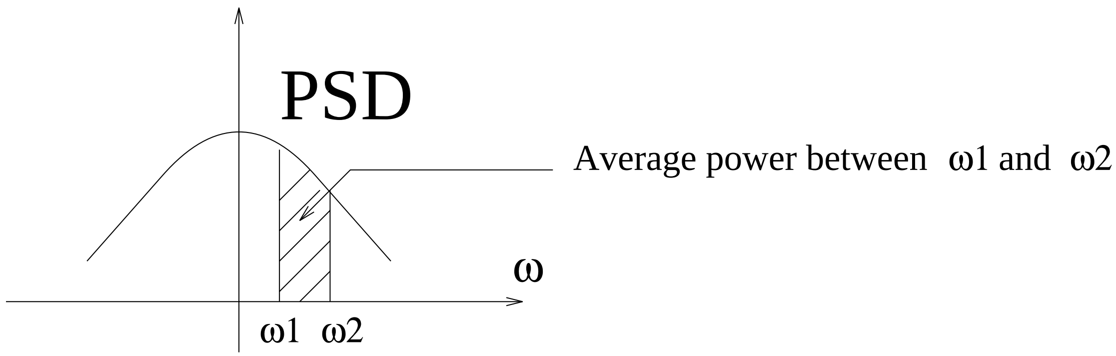
Or

$$P(f) = \sum_{k=-\infty}^{\infty} r(k) e^{-j2\pi f k}$$

$$r(k) = \int_{-\frac{1}{2}}^{\frac{1}{2}} P(f) e^{j2\pi f k} df$$

Remark: • Since $r(k)$ is discrete, $P(\omega)$ and $P(f)$ are periodic, with period 2π (ω) and 1 (f), respectively.

- We usually consider $\omega \in [-\pi, \pi]$ or $f \in [-\frac{1}{2}, \frac{1}{2}]$.
- $r(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\omega) d\omega$ = Average power for all frequency.



Second Definition of PSD.

$$P(\omega) = \lim_{N\to\infty} E\left\{ \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j\omega n}\right|^2 \right\}.$$

This definition is equivalent to the first one under

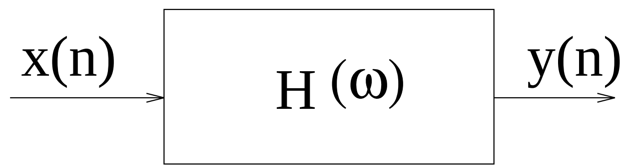
$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=-N+1}^{N-1} |k|\,|r(k)| = 0$$

(which means that $\{r(k)\}$ decays sufficiently fast).

Properties of PSD.

- $P(\omega) \geq 0$ for all ω.
- For real $x(n), r(k) = r(-k), \Rightarrow P(\omega) = P(-\omega), \omega \in [-\pi, \pi]$.
- For complex $x(n), r(k) = r^*(-k)$.

PSD for LTI Systems.



$$P_y(\omega) = P_x(\omega)|H(\omega)|^2.$$

Complex (DE) Modulation.

$$y(n) = x(n)e^{j\omega_0 n}.$$

It is easy to show that

$$r_y(k) = r_x(k)e^{j\omega_0 k}.$$

$$P_y(\omega) = P_x(\omega - \omega_0).$$

Spectral Estimation Problem

From a finite-length record $\{x(0), ..., x(N-1)\}$, determine an estimate $\hat{P}(\omega)$ of the PSD, $P(\omega)$, for $\omega \in [-\pi,\pi]$.

NonParametric Methods:

Periodogram:

Recall the second definition of PSD:

$$P(\omega) = \lim_{N\to\infty} E\left\{ \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j\omega n}\right|^2 \right\}.$$

$$\text{Periodogram} = \underline{\hat{P}_p(\omega) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j\omega n}\right|^2.}$$

Remark: • $\hat{P}_p(\omega) \geq 0$ for all ω.

• If $x(n)$ is real, $\hat{P}_p(\omega)$ is <u>even.</u>

• $E[\hat{P}_p(\omega)] = ?$ Var$[\hat{P}_p(\omega)] = ?$ (to be discussed later on)

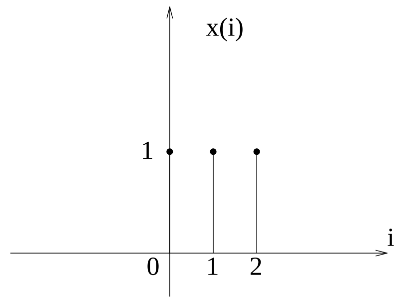
Correlogram (See first PSD definition)

$$\text{Correlogram} = \hat{P}_c(\omega) = \sum_{k=-(N-1)}^{N-1} \hat{r}(k) e^{-j\omega k}.$$

Unbiased Estimate of r(k):

$$\begin{cases} k \geq 0, & \hat{r}(k) = \frac{1}{N-k} \sum_{i=k}^{N-1} x(i) x^*(i-k) \\ k < 0, & \hat{r}(k) = \hat{r}^*(-k) \end{cases}$$

Ex.

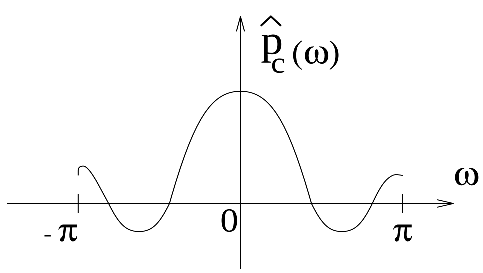


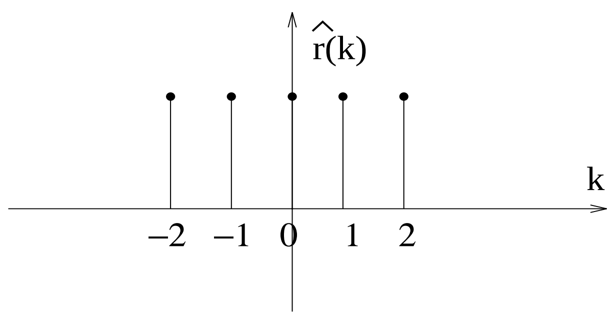
$\hat{r}(0) = \frac{1}{3}\sum_0^2 (1)(1) = 1$, (average of 3 points)

$\hat{r}(-1) = \hat{r}(1) = \frac{1}{2}\sum_1^2 (1)(1) = 1$, (average of 2 points)

$\hat{r}(-2) = \hat{r}(2) = \frac{1}{1}\sum_2^2 (1)(1) = 1$, (average of 1 point)

$\hat{r}(-3) = \hat{r}(3) = 0.$





Remark:

- $\hat{r}(k)$ is a bad estimate of $r(k)$ for large k.
- $E[\hat{r}(k)] = r(k)$ (unbiased)

Proof:

$$E[\hat{r}(k)] = E\left[\frac{1}{N-k}\sum_{i=k}^{N-1} x(i)x^*(i-k)\right]$$

$$= \frac{1}{N-k}\sum_{i=k}^{N-1} r(k) = r(k)$$

- $\hat{P}_c(\omega)$ based on unbiased $\hat{r}(k)$ may be ≤ 0.

Biased Estimate of $r(k)$ (used more often!)

$$\begin{cases} k \geq 0, & \hat{r}(k) = \frac{1}{N}\sum_{i=k}^{N-1} x(i)x^*(i-k), \\ k < 0, & \hat{r}(k) = \hat{r}^*(-k), \end{cases}$$

Remark:

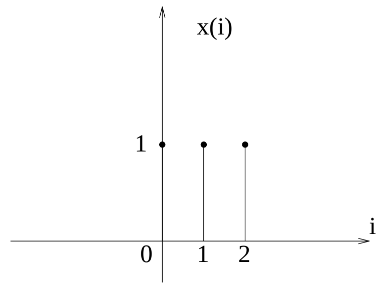
$$E[\hat{r}(k)] = \frac{1}{N}\sum_{i=k}^{N-1} E[x(i)x^*(i-k)]$$

$$= \frac{1}{N}\sum_{i=k}^{N-1} r(k) = \frac{N-k}{N} r(k)$$

$$\longrightarrow r(k), \text{ as } N \to \infty$$

(Asymptotically unbiased)

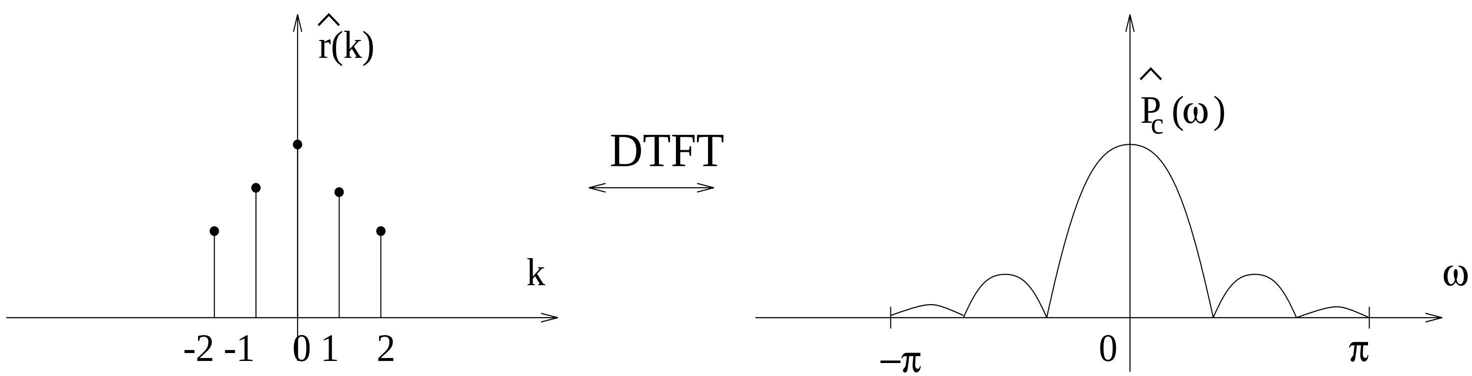
Ex.



$\hat{r}(0) = \frac{1}{3}\sum_0^2 (1)(1) = 1.$

$\hat{r}(-1) = \hat{r}(1) = \frac{1}{3}\sum_1^2 (1)(1) = \frac{2}{3}.$

$\hat{r}(-2) = \hat{r}(2) = \frac{1}{3}\sum_2^2 (1)(1) = \frac{1}{3}.$



Remark:

- With biased $\hat{r}(k), \hat{P}_c(\omega) = \hat{P}_p(\omega) \geq 0$,for all ω
- $E[\hat{r}(k)] \neq r(k)$

$E[\hat{r}(k)] \longrightarrow r(k)$, as $N \rightarrow \infty \Rightarrow$ Asymptotically unbiased.

- $$\hat{\mathbf{R}} = \begin{bmatrix} \hat{r}(0) & \hat{r}(1) & \cdots & \hat{r}(N-1) \\ \hat{r}^*(1) & \hat{r}(0) & \cdots & \hat{r}(N-2) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{r}^*(N-1) & \hat{r}^*(N-2) & \cdots & \hat{r}(0) \end{bmatrix},$$

with $\hat{r}(k)$ biased estimate. Then $\hat{\mathbf{R}}$ is positive semidefinite.

General Comments on $\hat{P}_p(\omega)$ and $\hat{P}_c(\omega)$.

- $\hat{P}_p(\omega)$ and $\hat{P}_c(\omega)$ provide POOR estimate of $P(\omega)$. (The variances of $\hat{P}_p(\omega)$ and $\hat{P}_c(\omega)$ are high.)

Reason: $\hat{P}_p(\omega)$ and $\hat{P}_c(\omega)$ are from a single realization of a random process.

- Compute $\hat{P}_p(\omega)$ via FFT.

Recall DFT: (N^2 complex multiplication)

$$X(k) = \sum_{i=0}^{N-1} x(i) e^{-j\frac{2\pi}{N}ki}$$

$$\hat{P}_p(k) = \frac{1}{N}|X(k)|^2.$$

Let

$$\begin{aligned} W &= e^{-j\frac{2\pi}{N}}, N = 2^m \\ X(k) &= \sum_{n=0}^{N-1} x(n)W^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W^{kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W^{kn} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right) W^{\frac{Nk}{2}} \right] W^{kn} \end{aligned}$$

Note:

$$W^{\frac{Nk}{2}} = e^{-j\frac{2\pi}{N}\frac{Nk}{2}} = e^{-j\pi k}$$

$$= \begin{cases} 1, & \text{even k} \\ -1, & \text{odd k} \end{cases}$$

$$\begin{cases} X(2p) = \sum_{n=0}^{N-1} \left[x(n) + x(n + \frac{N}{2})\right] W^{kn}, \quad k = 2p = 0, 2, \ldots \\ X(2p+1) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x(n + \frac{N}{2})\right] W^{kn}, \quad k = 2p+1, \end{cases}$$

which requires $2\left(\frac{N}{2}\right)^2$ complex multiplication

This process is continued till 2 points.

- Remark: An $N = 2^m$ -pt FFT requires $O(N \log_2 N)$ complex multiplications.
- Zero padding may be used so that $N = 2^m$.
- Zero padding will not change resolution of $\hat{P}_p(\omega)$.

FUNDAMENTALS OF ESTIMATION THEORY

Properties of a Good Estimator for a constant scalar a

- <u>Small Bias:</u>

$$\text{Bias} = E[\hat{a}] - a$$

- <u>Small Variance:</u>

$$\text{Variance} = E\left\{(\hat{a} - E[\hat{a}])^2\right\}$$

- <u>Consistent:</u>

$$\hat{a} \rightarrow a \text{ as Number of measurements} \rightarrow \infty.$$

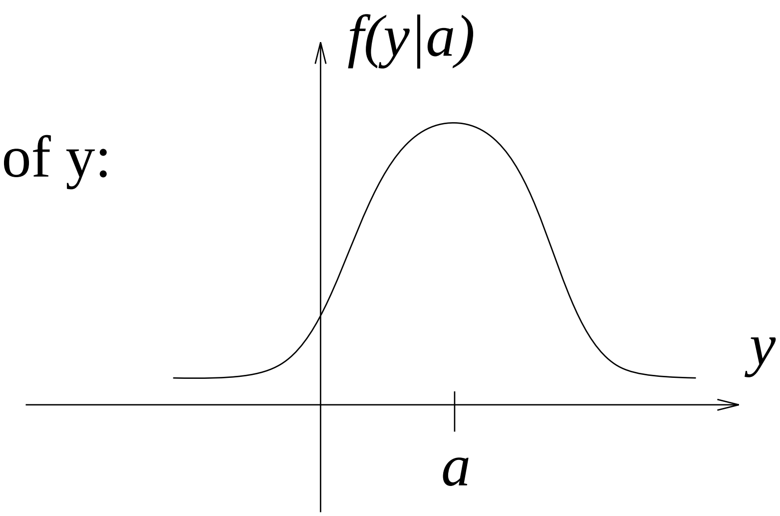
Ex. Measurement

$$y = a + e,$$

Where a is an unknown constant and e is $N(0,\sigma^2)$.

Find $\hat{a}$ from y ?

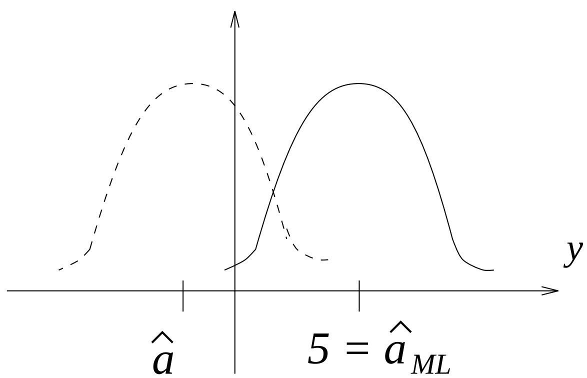
Pdf of y:



Maximum Likelihood (ML) Estimate of a:

Say $y = 5$, we want to find $\hat{a}$ so that it is most likely that the measurement is 5

$$\underline{\frac{\partial f(y|a)}{\partial a}\Big|_{a=\hat{a}_{ML}} = 0.}$$



- $\Rightarrow \underline{\hat{a}_{ML} = y}$
- $E[\hat{a}_{ML}] = E[y] = E[a+n] = a$
- $Var[\hat{a}_{ML}] = Var[y] = \sigma^2$

Ex. $y = a + e$

Three independent measurements y_1, y_2, y_3 are taken.

$\hat{a}_{ML} = ?$ Bias $= ?$ Variance $= ?$

$$f(y_i|a) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y_i-a)^2}{2\sigma^2}}.$$

$$f(y_1, y_2, y_3|a) = \Pi_{i=1}^{3} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y_i-a)^2}{2\sigma^2}}.$$

$$\frac{\partial f(y_1,y_2,y_3|a)}{\partial a}\Big|_{a=\hat{a}_{ML}} = 0$$

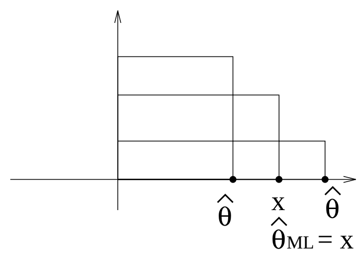
$$\Rightarrow \hat{a}_{ML} = \frac{1}{3}(y_1 + y_2 + y_3).$$

$$E[\hat{a}_{ML}] = E\left[\frac{1}{3}(y_1 + y_2 + y_3)\right] = a.$$

$$\begin{aligned} Var[\hat{a}_{ML}] &= \frac{1}{9} Var(y_1 + y_2 + y_3) \\ &= \frac{1}{9}(\sigma^2 + \sigma^2 + \sigma^2) = \frac{\sigma^2}{3}. \end{aligned}$$

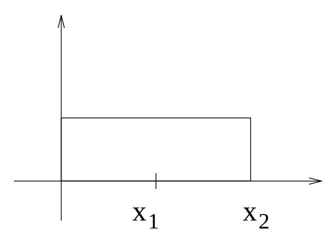
Ex. x is a measurement of an uniformly distributed random variable on $[0, \theta\ ]$, where θ is an unknown constant. $\hat{\theta}_{ML} =\ ?$

$\hat{\theta}_{ML} = x$



Question: What if two independent measurements x_1 and x_2 are taken ?

$\hat{\theta}_{ML}$ =max $(x_1, x_2\ )$.



Cramér - Rao Bound.

Let $B(a) = E\left[\hat{a}(r)|a\right] - a$ denote the bias of $\hat{a}(r)$, where r is the measurement.

Then

$$MSE = E\left[(\hat{a}(r) - a)^2|a\right] \geq \frac{\left[1+\frac{\partial}{\partial a}B(a)\right]^2}{E\left\{\left[\frac{\partial}{\partial a}\ln f(r|a)\right]^2|a\right\}} \; .$$

* The denominator of the CRB is known as Fisher's Information, $I(a)$.

* If $B(a) = 0$, the numerator of CRB is 1.

Proof:

$$\begin{aligned}
B(a) &= E\left[\hat{a}(r) - a|a\right] \\
&= \int_{-\infty}^{\infty} \left[\hat{a}(r) - a\right] f(r|a) dr \\
\frac{\partial}{\partial a} B(a) &= \int_{-\infty}^{\infty} \left[\hat{a}(r) - a\right] \frac{\partial}{\partial a} f(r|a) dr - \underbrace{\int_{-\infty}^{\infty} f(r|a) dr}_{=1} \\
1 + \frac{\partial}{\partial a} B(a) &= \int_{-\infty}^{\infty} \left[\hat{a}(r) - a\right] f(r|a) \frac{\partial}{\partial a} f(r|a) \frac{1}{f(r|a)} dr
\end{aligned}$$

But

$$\begin{aligned}
\frac{\partial}{\partial a} \ln f(r|a) &= \frac{\frac{\partial}{\partial a} f(r|a)}{f(r|a)} \\
1 + \frac{\partial}{\partial a} B(a) &= \int_{-\infty}^{\infty} \left[\hat{a}(r) - a\right] f(r|a) \frac{\partial}{\partial a} \ln f(r|a) dr
\end{aligned}$$

$$\Rightarrow \left\{ \int_{-\infty}^{\infty} \left[\hat{a}(r) - a\right] \sqrt{f(r|a)} \left[\left(\tfrac{\partial}{\partial a} \ln f(r|a) \right) \sqrt{f(r|a)} \right] dr \right\}^2$$

$$= \left[1 + \tfrac{\partial}{\partial a} B(a)\right]^2.$$

Schwarz Inequality:

$$\int_{-\infty}^{\infty} g_1(x) g_2(x) dx \leq \left[\int_{-\infty}^{\infty} {g_1}^2(x) dx\right]^{\frac{1}{2}} \left[\int_{-\infty}^{\infty} {g_2}^2(x) dx\right]^{\frac{1}{2}},$$

where " = " holds iff $g_1(x) = c g_2(x)$ for some constant c (c is independent of x).

$$\Rightarrow \left[1 + \frac{\partial}{\partial a} B(a)\right]^2 \leq \left\{\int_{-\infty}^{\infty} [\hat{a}(r) - a]^2 f(r|a) dr\right\} \cdot \underbrace{\left\{\int_{-\infty}^{\infty} \left[\frac{\partial}{\partial a} \ln f(r|a)\right]^2 f(r|a) dr\right\}}_{I(a)}$$

where "= " holds iff

$$\boxed{\hat{a}(r) - a = c \frac{\partial}{\partial a} \ln f(r|a).}$$

(where c is a constant independent of r).

Efficient Estimate:

An estimate is efficient if

(a.) It is unbiased

(b.) It achieves the CR - bound, i.e, $E\left\{[\hat{a}(r)-a]^2|a\right\} = \text{CRB}$.

Ex. $r = a + e$

where a is unknown constant, $e \sim N(0, \sigma^2)$. $\hat{a}_{ML} = ?$ efficient ?

$$f(r|a) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(r-a)^2}$$

$$\ln f(r|a) = \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}(r-a)^2.$$

$$\frac{\partial}{\partial a} \ln f(r|a) = -\frac{1}{2\sigma^2} 2(r-a)$$

$$= \frac{1}{\sigma^2}(a-r).$$

$$\frac{\partial}{\partial a}\ln f(r|a)\Big|_{a=\hat{a}_{ML}} = 0 \quad \Rightarrow \underline{\hat{a}_{ML} = r}$$

$$\begin{aligned} \frac{\partial}{\partial a}\ln f(r|a) &= \frac{1}{\sigma^2}(a - \hat{a}_{ML}) \\ \Rightarrow -\sigma^2 \frac{\partial}{\partial a}\ln f(r|a) &= \hat{a}_{ML} - a \end{aligned}$$

$$\Rightarrow \quad \hat{a}_{ML} \text{ efficient} \begin{cases} E\left[(\hat{a}_{ML} - a)^2 \Big| a \right] = CRB \\ E\left[\hat{a}_{ML}\right] = E\left[r\right] = a, \quad \text{unbiased} \end{cases}$$

Remark: • MSE $= Var[\hat{a}_{ML}] = Var[r] = \sigma^2$.

$$\bullet I(a) = E\left\{ \left[\frac{\partial}{\partial a}\ln f(r|a)\right]^2 \Bigg| a \right\} = E\left\{ \left[\frac{1}{\sigma^2}(a-r)\right]^2 \right\} = \frac{1}{\sigma^4}\sigma^2 = \frac{1}{\sigma^2}$$

$$\Rightarrow \quad \text{CRB} = \frac{1}{I(a)} = \sigma^2 = Var[\hat{a}_{ML}].$$

Remarks:

(1) If $\hat{a}(r)$ is unbiased, $Var[\hat{a}(r)\ ] \geq$ CRB.

(2) If an efficient estimate $\hat{a}(r)$ exists, i.e,

$$\frac{\partial}{\partial a} \ln f(r|a) = c[\hat{a}(r) - a]. \quad (c \quad \text{is independent of} \quad r.)$$

then

$$0 = \frac{\partial}{\partial a} \ln f(r|a)|_{a=\hat{a}_{ML}(r)} \text{ results in } \hat{a}_{ML}(r) = \hat{a}(r).$$

$\Rightarrow$ If an efficient estimate exists, it is $\hat{a}_{ML}$.

(3) If an efficient estimate <u>does not</u> exist, how good $\hat{a}_{ML}(r)$ is depends on each specific problem.

<u>No estimator can achieve the CR-bound</u>. Bounds (for example, Bhattacharya, Barankin) larger than the CR-bound may be found.

Independent measurements $r_1, ..., r_N$ available, where r_i may or may not be Gaussian.

Assume

$$\hat{a}_{ML} = \frac{1}{N} \sum_{i=1}^{N} r_i.$$

Law of large numbers: $\hat{a}_{ML} \underset{N \to \infty}{\longrightarrow} a$

Central Limit Theorem:

$\hat{a}_{ML}$ has Gaussian distribution as $N \to \infty$.

Asymptotic Properties of $\hat{a}_{ML}(r_1, ..., r_N)$

(a) $\hat{a}_{ML}(r_1, ..., r_N) \underset{N\to\infty}{\longrightarrow} a$ ($\hat{a}_{ML}$ is a consistent estimate.)

(b) $\hat{a}_{ML}$ is asymptotically efficient.

(c) $\hat{a}_{ML}$ is aymptotically Gaussian.

Ex. $r = g^{-1}(a) + e, \quad e \sim N(0, \sigma^2). \quad \hat{a}_{ML} = ? \quad$ efficient ?

$$\text{Let} \quad b = g^{-1}(a). \quad \text{Then} \quad a = g(b)$$

$$\frac{\partial}{\partial a} \ln f(r|a) = \frac{1}{\sigma^2}\left(r - g^{-1}(a)\right) \frac{dg^{-1}(a)}{da}\Big|_{a=\hat{a}_{ML}} = 0$$

$$\hat{a}_{ML} = g(r) = g(\hat{b}_{ML}).$$

<u>Invariance property of ML estimator</u>

- If $a = g(b)$ then $\hat{a}_{ML} = g(\hat{b}_{ML})$.
- $\hat{a}_{ML}$ may not be efficient. $\hat{a}_{ML}$ is not efficient if $g(\cdot)$ is a nonlinear function.

PROPERTIES OF PERIODOGRAM

Bias Analysis

- When $\hat{r}(k)$ is a biased estimate,

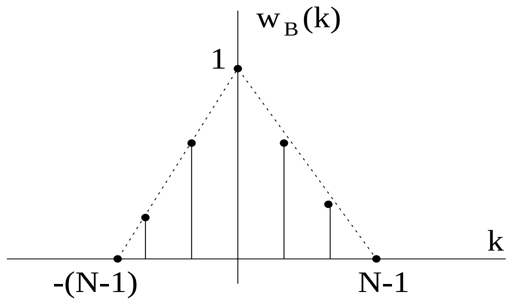
$$E\left[\hat{P}_p(\omega)\right] = E\left[\hat{P}_c(\omega)\right] = E\left\{\sum_{k=-(N-1)}^{N-1} \hat{r}(k)e^{-j\omega k}\right\}$$

$$k \geq 0, \quad E\left[\hat{r}(k)\right] = \frac{N-k}{N} r(k),$$

$$k < 0, \quad E\left[\hat{r}(k)\right] = E\left[r^*(-k)\right] = \frac{N+k}{N} r^*(-k) = \frac{N-|k|}{N} r(k),$$

$$E\left[\hat{P}_p(\omega)\right] = \sum_{k=-(N-1)}^{N-1} \left(1 - \frac{|k|}{N}\right) r(k)e^{-j\omega k}.$$

Bartlett or Triangular Window.



$$E\left[\hat{P}_p(\omega)\right] = \sum_{k=-\infty}^{\infty} \left[w_B(k)r(k)\right] e^{-j\omega k}$$

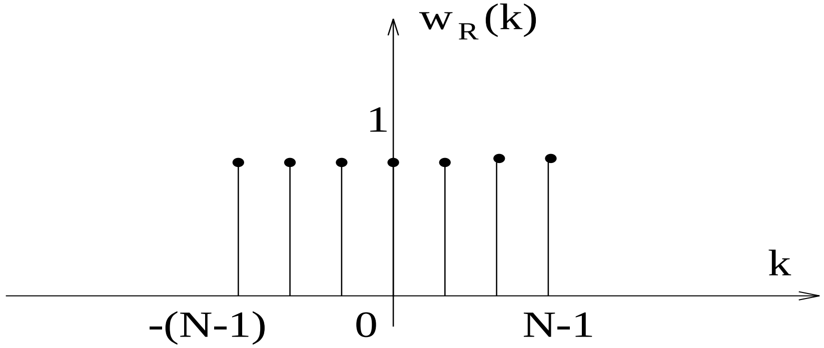
$$\text{Let} \quad w_B(k) \overset{DTFT}{\longleftrightarrow} W_B(\omega)$$

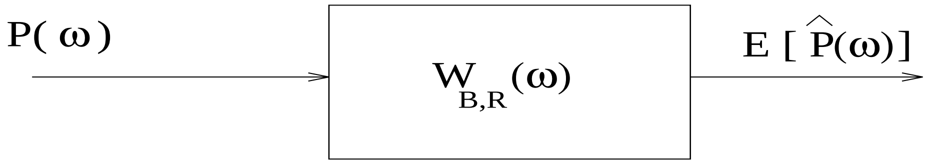
$$\boxed{E\left[\hat{P}_p(\omega)\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\psi)W_B(\omega-\psi)d\psi.}$$

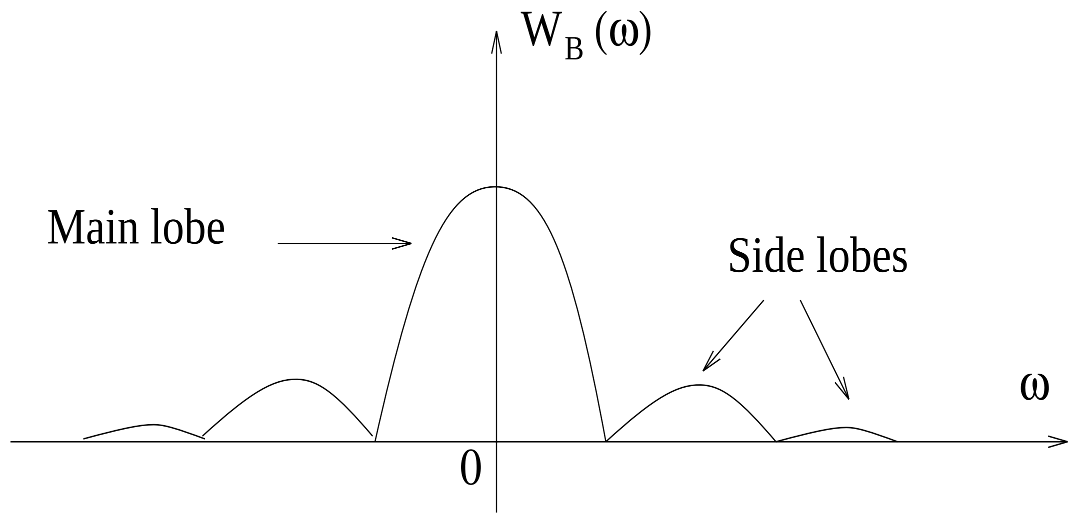
- When $\hat{r}(k)$ is unbiased estimate,

$$\boxed{E\left[\hat{P}_p(\omega)\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\psi)W_R(\omega-\psi)d\psi \ .}$$

$$w_R(k) \overset{DTFT}{\longleftrightarrow} W_R(\omega)$$







$$\text{3 dB power width of main lobe} \approx \frac{2\pi}{N} \text{ (or } \frac{1}{N} \text{ in Hz) .}$$

Remark:• The main lobe of $W_B(\omega)$ smears or smooths $P(\omega)$.

• Two peaks in $P(\omega)$ that are separated less than $\frac{2\pi}{N}$ cannot be resolved in $\hat{P}_p(\omega)$.

• $\frac{1}{N}$ in Hz is called spectral resolution limit of periodogram methods.

Remark:

- The side lobes of $W_B(\omega)$ transfer power from high power frequency bins to low power frequency bins —— leakage.
- Smearing and leakage cause more problems to peaky $P(\omega)$ than to flat $P(\omega)$.

If $P(\omega) = \sigma^2$, for all ω, $E[\hat{P}_p(\omega)] = P(\omega)$.

- Bias of $\hat{P}_p(\omega)$ decreases as $N \to \infty$. (asymptotically unbiased.)

Variance Analysis

We shall consider the case $x(n)$ is zero-mean circularly symmetric complex Gaussian white noise.

$$\odot \begin{cases} E[x(n)x^*(k)] = \sigma^2\delta(n-k). \\ E[x(n)x(k)] = 0 \quad \text{for all} \quad n,k. \end{cases}$$

$\odot$ is equivalent to:

$$\begin{cases} E[\,\mathrm{Re}(x(n))\mathrm{Re}(x(k))] = \frac{\sigma^2}{2}\delta(n-k). \\ E[\mathrm{Im}(x(n))\mathrm{Im}(x(k))] = \frac{\sigma^2}{2}\delta(n-k). \\ E[\mathrm{Re}(x(n))\mathrm{Im}(x(k))] = 0. \end{cases}$$

Remark: The real and imaginary parts of $x(n)$ are $N(0, \frac{\sigma^2}{2})$ and independent of each other.

Remark: If $x(n)$ is zero-mean complex Gaussian white noise, $\hat{P}_p(\omega)$ is an unbiased estimate.

- $$r(k) = \sigma^2 \delta(k).$$

$$E\left[\hat{P}_p(\omega)\right] = \sum_{k=-(N-1)}^{N-1} \left(1 - \frac{|k|}{N}\right) r(k) e^{-j\omega k} = \sigma^2$$

- $$P_p(\omega) = \sum_{k=-\infty}^{\infty} r(k) e^{-j\omega k} = \sigma^2$$

$$= E\left[\hat{P}_p(\omega)\right].$$

For Gaussian complex white noise,

$$E\left[x(k)x^*(l)x(m)x^*(n)\right] = \sigma^4\left[\delta(k-l)\delta(m-n) + \delta(k-n)\delta(l-m)\right].$$

$$\begin{aligned}
E\left[\hat{P}_p(\omega_1)\hat{P}_p(\omega_2)\right] &= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} E\left[x(k)x^*(l)x(m)x^*(n)\right] \\
&\quad e^{-j\omega_1(k-l)}e^{-j\omega_2(m-n)} \\
&= \sigma^4 + \frac{\sigma^4}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} e^{-j(\omega_1-\omega_2)(k-l)} \\
&= \sigma^4 + \frac{\sigma^4}{N^2}\left|\sum_{k=0}^{N-1} e^{j(\omega_1-\omega_2)k}\right|^2 \\
&= \sigma^4 + \frac{\sigma^4}{N^2}\left\{\frac{\sin[(\omega_1-\omega_2)\frac{N}{2}]}{\sin\frac{(\omega_1-\omega_2)}{2}}\right\}^2
\end{aligned}$$

$$\lim_{N\to\infty} E\left[\hat{P}_p(\omega_1)\hat{P}_p(\omega_2)\right] = P(\omega_1)P(\omega_2) + P^2(\omega_1)\delta(\omega_1 - \omega_2).$$

$$\lim_{N\to\infty} E\left\{\left[\hat{P}_p(\omega_1) - P(\omega_1)\right]\left[\hat{P}_p(\omega_2) - P(\omega_2)\right]\right\}$$

$$= \begin{cases} P^2(\omega_1), & \omega_1 = \omega_2 \\ 0, & \omega_1 \neq \omega_2 \text{ (uncorrelated if } \omega_1 \neq \omega_2) \end{cases}$$

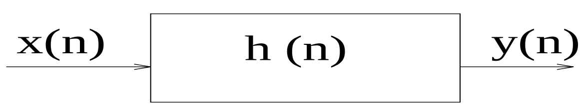
Remark: • $\hat{P}_p(\omega)$ is not a consistent estimate.

• If $\omega_1 \neq \omega_2$, $\hat{P}_p(\omega_1)$ and $\hat{P}_p(\omega_2)$ are uncorrelated with each other.

• This variance result is also true for

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k),$$

where $x(n)$ is zero-mean complex Gaussian white noise.



REFINED METHODS

Decrease variance of $\hat{P}(\omega)$ by increasing bias or decreasing resolution .

Blackman - Tukey (BT) Method

Remark: The $\hat{r}(k)$ used in $\hat{P}_c(\omega)$ is poor estimate for large lags k.

$$M < N : \quad \hat{P}_{BT}(\omega) = \sum_{k=-(M-1)}^{M-1} w(k)\hat{r}(k)e^{-j\omega k},$$

where $w(k)$ is called lag window.

Remark: If $w(k)$ is rectangular, $w(k)\hat{r}(k)$ is a truncated version of $\hat{r}(k)$.

If $\hat{r}(k)$ is a biased estimate, and $w(k) \overset{DTFT}{\longleftrightarrow} W(\omega)$

$$\boxed{\hat{P}_{BT}(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{W}(\omega - \psi)\hat{P}_p(\psi)d\psi \ .}$$

Remark: • BT spectral estimator is "locally" weighted average of periodogram $\hat{P}_p(\omega)$.

• The smaller the M, the poorer the resolution of $\hat{P}_{BT}(\omega)$ but the lower the variance.

• Resolution of $\hat{P}_{BT}(\omega) \propto \frac{1}{M}$.

• Variance of $\hat{P}_{BT}(\omega) \propto \frac{M}{N} \quad \xrightarrow[N\to\infty]{M \text{ fixed}} 0$.

• For fixed M , $\hat{P}_{BT}(\omega)$ is asymptotically biased but variance $\longrightarrow 0$.

Question: When is $\hat{P}_{BT}(\omega) \geq 0 \; \forall \; \omega$?

Theorem: Let $Y(\omega) \overset{DTFT}{\longleftrightarrow} y(n), \quad -(N-1) \leq n \leq N-1$

Then $Y(\omega) \geq 0 \; \forall \; \omega$ iff

$$\begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) & 0 & \cdots \\ y(-1) & y(0) & \cdots & y(N-2) & y(N-1) & \cdots \\ \vdots & & \ddots & & & \\ y[-(N-1)] & \cdots & & y(0) & y(1) & \cdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & & & & \end{bmatrix}$$

is positive semidefinite.

In other words, $Y(\omega) \geq 0 \; \forall \; \omega$ iff

$\cdots, 0, \cdots, 0, y[-(N-1)], \cdots, y(0), y(1), \cdots, y(N-1), 0, \cdots$ is a positive semidefinite sequence.

Remark: • $\hat{P}_{BT}(\omega) \geq 0 \; \forall \; \omega$ iff $\{w(k)\hat{r}(k)\}$ is a positive semidefinite sequence.

• $\hat{P}_{BT}(\omega) \geq 0 \; \forall \; \omega$ iff

$$\hat{\mathbf{R}}_{BT} =$$

$$\begin{bmatrix} w(0)\hat{r}(0) & \cdots & w(M-1)\hat{r}(M-1) & 0 & \cdots \\ & \ddots & & \ddots & \\ w[-(M-1)]\hat{r}[-(M-1)] & \cdots & w(0)\hat{r}(0) & & \\ 0 & \ddots & & & \ddots \\ \vdots & & & & \end{bmatrix}$$

is positive semidefinite, i.e, $\hat{\mathbf{R}}_{BT} \geq 0$.

$$\hat{\mathbf{R}}_{BT} = \begin{bmatrix} w(0) & \cdots & w(M-1) & 0 & \cdots \\ & \ddots & & & \\ w[-(M-1)] & \cdots & w(0) & & \\ 0 & \ddots & & \ddots & \\ \vdots & & & & \end{bmatrix}$$

$$\odot \begin{bmatrix} \hat{r}(0) & \cdots & \hat{r}(N-1) & 0 & \cdots \\ & \ddots & & & \\ \hat{r}[-(N-1)] & \cdots & \hat{r}(0) & & \\ 0 & \ddots & & \ddots & \\ \vdots & & & & \end{bmatrix}$$

$\odot$ = Hadamard matrix product:

$(ij)^{th}$ element: $(\mathbf{A} \odot \mathbf{B})_{ij} = \mathbf{A}_{ij}\mathbf{B}_{ij}$

Theorem:

If $\mathbf{A} \geq 0$ (positive semidefinite) $\mathbf{B} \geq 0$ then $\mathbf{A} \bigodot \mathbf{B} \geq 0$.

Remark: If $\hat{r}(k)$ is a biased estimate, $\hat{P}_p(\omega) \geq 0 \; \forall \; \omega$. Then if $\mathrm{W}(\omega) \geq 0 \; \forall \; \omega$, we have $\hat{P}_{BT}(\omega) \geq 0 \; \forall \; \omega$.

Remark: Nonnegative definite (positive semidefinite) window sequences: Bartlett, Parzen.

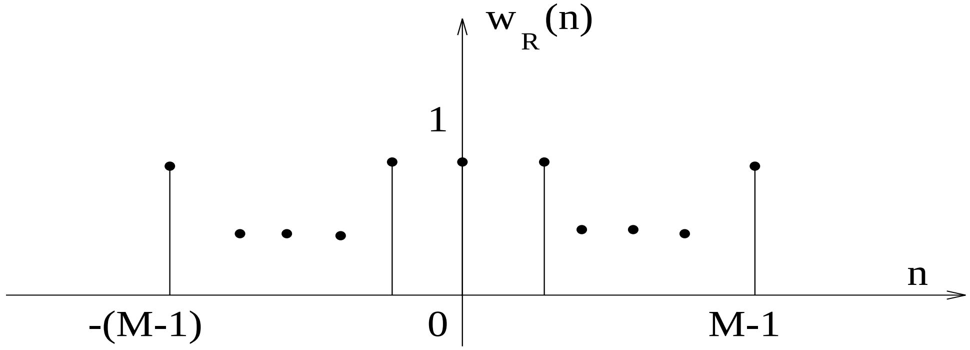
Time-Bandwidth Product

- **Equivalent Time Width N_e:**

$$N_e = \frac{\sum_{n=-(M-1)}^{M-1} w(n)}{w(0)}$$

Ex.

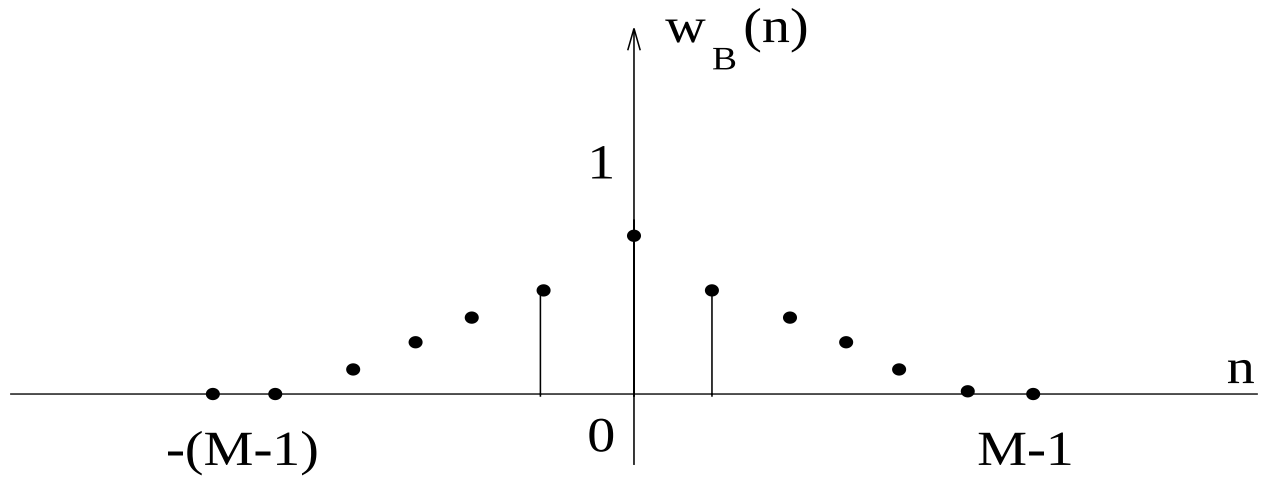
$$N_e = \frac{\sum_{k=-(M-1)}^{M-1}(1)}{1} = 2M - 1.$$



Ex.

$$w_B(n) = \begin{cases} 1 - \frac{|n|}{M}, & -(M-1) \leq n \leq (M-1) \\ 0, & \text{else} \end{cases}$$

$N_e = M$



- **Equivalent Bandwidth** β_e:

$$\boxed{2\pi\beta_e = \frac{\int_{-\pi}^{\pi} W(\omega)d\omega}{W(0)}}$$

$$\text{Since} \quad w(n) \overset{DTFT}{\longleftrightarrow} W(\omega).$$

$$w(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)e^{j\omega n}d\omega.$$

$$\Rightarrow w(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)d\omega.$$

$$W(\omega) = \sum_{n=-(M-1)}^{M-1} w(n)e^{-j\omega n}.$$

$$\Rightarrow W(0) = \sum_{n=-(M-1)}^{M-1} w(n)$$

$$N_e\beta_e = \frac{\sum_{n=-(M-1)}^{M-1} w(n)}{\frac{1}{2\pi}\int_{-\pi}^{\pi} W(\omega)d\omega} \frac{\int_{-\pi}^{\pi} W(\omega)d\omega}{2\pi \sum_{n=-(M-1)}^{M-1} w(n)} = 1$$

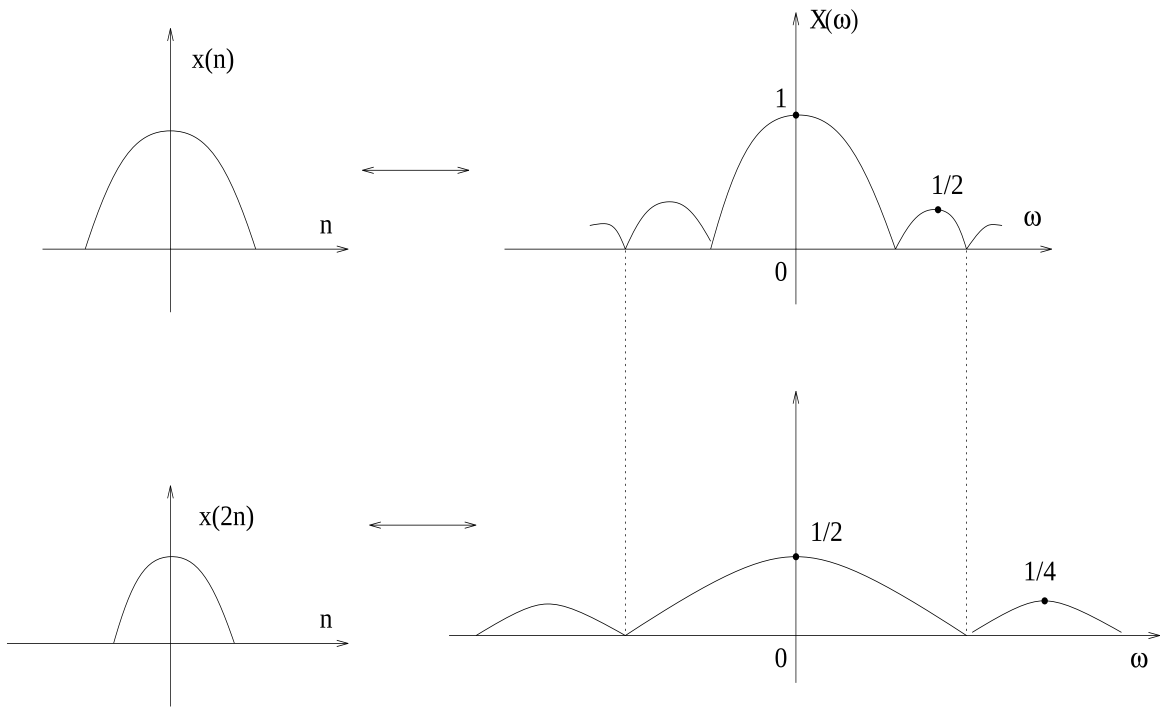
$\Rightarrow$ $\boxed{N_e\beta_e = 1}$ (Time Bandwidth product.)

Remark:

- If a signal decays slowly in one domain, it is more concentrated in the other domain.
- Window shape determines the side lobe level relative to $W(0)$.

Ex:

$$x(2n) \overset{DTFT}{\longleftrightarrow} \frac{1}{2} X\left(\frac{\omega}{2}\right).$$



Remark: • Once the window shape is fixed, M $\uparrow \rightarrow N_e \uparrow \rightarrow \beta_e \downarrow$.

$\Rightarrow$ M $\uparrow \rightarrow$ main lobe width $\downarrow$.

Window design for $\hat{P}_{BT}(\omega)$

Let β_m = 3dB main lobe width.

Resolution of $\hat{P}_{BT}(\omega) \sim \beta_m$ Variance of $\hat{P}_{BT}(\omega) \sim \frac{1}{\beta_m}$.

- Choice of β_m is based on the trade-off between resolution and variance, and N
- Choice of window shape is based on leakage, and N.

- **Practical rule of thumb:**

1. $M \leq \frac{N}{10}$.
2. Window shape based on trade-off between smearing and leakage.
3. Window shape for $\hat{P}_{BT}(\omega) \geq 0, \quad \forall \omega$

Remark: • Other methods for Non-parametric Spectral Estimation include: Bartlett, Welch, Daniell Methods.

- All try to reduce variance at the expense of poorer resolution.

Bartlett Method

x(n): $\underbrace{\bullet\bullet\bullet\cdots}_{x_1(n)}\underbrace{\bullet\bullet\bullet\cdots}_{x_2(n)}\bullet\bullet\bullet\cdots\underbrace{\bullet\bullet\bullet\cdots}_{x_L(n)}$

- $x(n)$ is an N point sequence.
- $x_l(n), l = 1, \cdots, L$, are M point sequences.
- $x_l(n)$ are non-overlapping. $L = \frac{N}{M}$.

$$\hat{P}_l(\omega) = \frac{1}{M}\left|\sum_{n=0}^{M-1} x_l(n)e^{-j\omega n}\right|^2$$

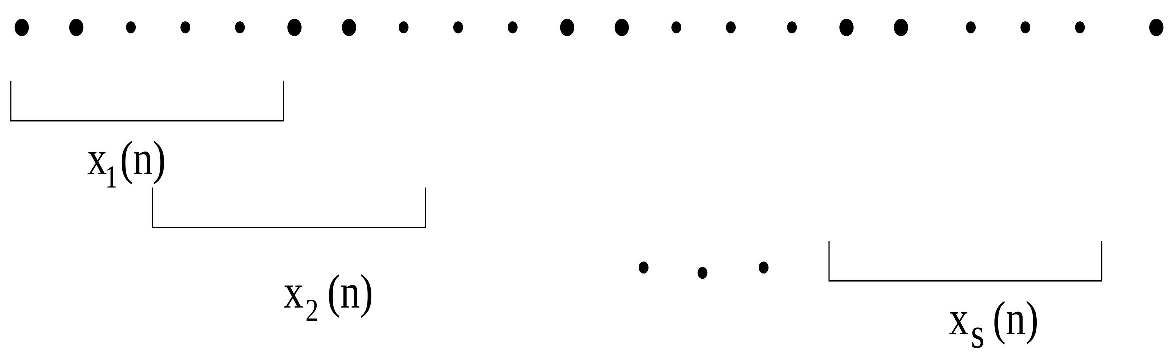
$$\hat{P}_B(\omega) = \frac{1}{L}\sum_{l=1}^{L} \hat{P}_l(\omega).$$

Remark:

- $\hat{P}_B(\omega) \geq 0, \forall\ \omega$.
- For large M and L, $\hat{P}_B(\omega) \approx [\ \hat{P}_{BT}(\omega)$ using $w_R(n)\ ]$

Welch Method:

- $x_l(n)$ may overlap in the Welch method.
- $x_l(n)$ may be windowed before computing Periodogram.



Let $w(n)$ be the window applied to $\quad x_l(n), l = 1, .., S, n = 0, .., M-1$

Let

$$\text{P} = \text{power of} \quad w(n) = \frac{1}{M} \sum_{n=0}^{M-1} |w(n)|^2$$

$$\hat{P}_l(\omega) = \frac{1}{MP}\left|\sum_{n=0}^{M-1} w(n)x_l(n)e^{-j\omega n}\right|^2$$

$$\hat{P}_W(\omega) = \frac{1}{S}\sum_{l=1}^{S} \hat{P}_l(\omega)$$

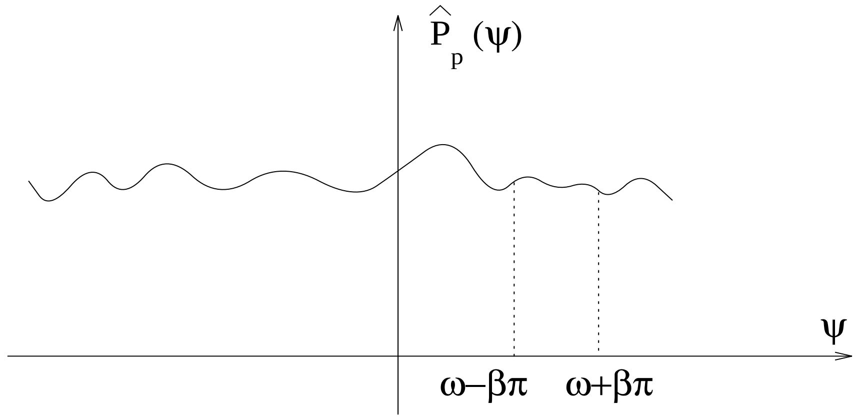
Remarks: • By allowing $x_l(n)$ to overlap, we hope to have a larger S, the number of $\hat{P}_j(\omega)$ we average. 50% overlap in general.

Practical examples show that $\hat{P}_W(\omega)$ may offer lower variance than $\hat{P}_B(\omega)$, but not significantly.

- $\hat{P}_W(\omega)$ may be shown to be $\hat{P}_{BT}(\omega)$ -type estimator, under reasonable approximation.
- $\hat{P}_W(\omega)$ can be easily computed with FFT -favored in practice
- $\hat{P}_{BT}(\omega)$ is theoretically favored.

Daniell Method:

$$\hat{P}_D(\omega) = \frac{1}{2\pi\beta} \int_{\omega-\beta\pi}^{\omega+\beta\pi} \hat{P}_p(\psi) d\psi.$$



Remark: • $\hat{P}_D(\omega)$ is a special case of $\hat{P}_{BT}(\omega)$ with

$$w(n) \text{ in } \hat{P}_{BT}(\omega) \overset{DTFT}{\longleftrightarrow} W(\omega) = \begin{cases} \frac{1}{\beta}, \omega \in [-\beta\pi, \beta\pi] \\ 0, \text{else} \end{cases}.$$

• The larger the β, the lower the variance, but the poorer the resolution.

Implementation of $\hat{P}_D(\omega)$

- Zero pad $x(n)$ so that $x(n)$ has N' points, $N' >> N$.
- Calculate $\hat{P}_p(\omega_k)$ with FFT.

$$\omega_k = \frac{2\pi}{N'}k, \quad k = 0, \cdots, N'-1.$$

-

$$\hat{P}_D(\omega_k) = \frac{1}{2J+1} \sum_{j=k-J}^{k+J} \hat{P}_p(\omega_j).$$

$$\hat{P}_p(\omega) \quad \bullet\bullet\bullet \quad \underbrace{\cdots \quad \bullet\bullet \quad \cdots}_{\text{2J+1 points averaging}} \quad \bullet$$

$$\Downarrow$$

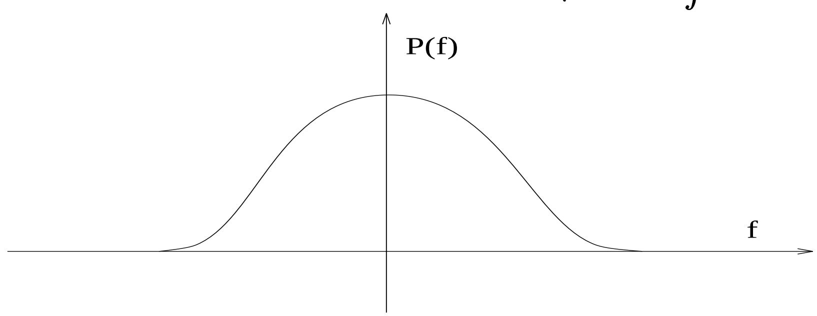
$$\hat{P}_D(\omega_k)$$

PARAMETRIC METHODS

Parametric Modeling

Ex.

$$P(f) = \frac{r(0)}{\sqrt{2\pi}\sigma_f} e^{-\frac{1}{2}\left(\frac{f}{\sigma_f}\right)^2}, |f| \leq \frac{1}{2}$$



Remark: • $P(f)$ is described by 2 unknowns: $r(0)$ and σ_f.

• Once we know $r(0)$ and σ_f, we know $P(f)$, the PSD.

• Nonparametric methods assume no knowledge on $P(f)$ – too many unknowns.

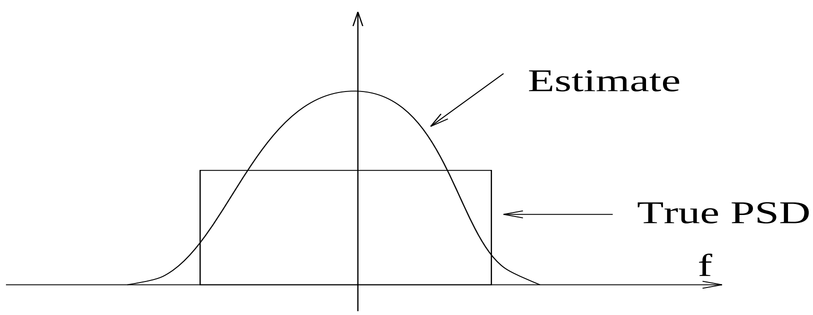
• Parametric Methods attempt to estimate $r(0)$ and σ_f.

Parsimony Principle:

Better estimates may be obtained by using an appropriate data model with fewer unknowns.

Appropriate Data Model.

- If data model wrong, $\hat{P}(f)$ will always be biased.



- To use parametric methods, reasonably correct *'a priori'* knowledge on data model is necessary.

Rational Spectra:

$$\boxed{P(\omega) = \sigma^2 \left| \frac{B(\omega)}{A(\omega)} \right|^2}$$

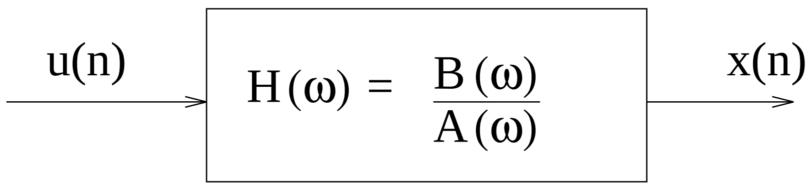
$$A(\omega) = 1 + a_1 e^{-j\omega} + \cdots + a_p e^{-jp\omega}$$

$$B(\omega) = 1 + b_1 e^{-j\omega} + \cdots + b_q e^{-jq\omega}.$$

Remark: • We mostly consider real valued signals here.

• $a_1, \cdots, a_p, b_1, \cdots, b_q$ are real coefficients.

• Any continuous PSD can be approximated arbitrarily close by a rational PSD.



u(n) = zero-mean white noise of variance σ^2.

$$P_{xx}(\omega) = \sigma^2 \left| \frac{B(\omega)}{A(\omega)} \right|^2 .$$

Remark:

The rational spectra can be associated with a signal obtained by filtering white noise of power σ^2 through a rational filter with

$$H(\omega) = \frac{B(\omega)}{A(\omega)}.$$

In <u>Difference Equation</u> Form,

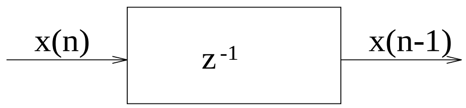
$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + \sum_{k=0}^{q} b_k u(n-k).$$

In <u>Z-transform</u> Form, $\underline{\ z = e^{j\omega}\ }$

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

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p}$$

$$B(z) = 1 + b_1 z^{-1} + \cdots + b_q z^{-q}$$



Unit Delay line

<u>Notation sometimes used :</u> $z^{-1}x(n) = x(n-1)$

$$\text{Then:} \quad x(n) = \frac{B(z)}{A(z)} u(n)$$

ARMA Model: ARMA(p,q)

$$P(\omega) = \sigma^2 \left| \frac{B(\omega)}{A(\omega)} \right|^2.$$

AR Model: AR(p)

$$P(\omega) = \sigma^2 \left| \frac{1}{A(\omega)} \right|^2.$$

MA Model: MA(q)

$$P(\omega) = \sigma^2 |B(\omega)|^2.$$

Remark: • AR models peaky PSD better .

• MA models valley PSD better.

• ARMA is used for PSD with both peaks and valleys.

Spectral Factorization:

$$H(\omega) = \frac{B(\omega)}{A(\omega)}$$

$$P(\omega) = \sigma^2 \left| \frac{B(\omega)}{A(\omega)} \right|^2 = \frac{\sigma^2 B(\omega) B^*(\omega)}{A(\omega) A^*(\omega)}.$$

$$A(\omega) = 1 + a_1 e^{-j\omega} + \cdots + a_p e^{-jp\omega}$$

$b_1, \cdots, b_q, a_1, \cdots, a_p$ are real coefficients.

$$\begin{aligned} A^*(\omega) &= 1 + a_1 e^{j\omega} + \cdots + a_p e^{jp\omega} \\ &= 1 + a_1 \frac{1}{z} + \cdots + a_p \frac{1}{z^p} = A(\frac{1}{z}) \end{aligned}$$

$$\boxed{P(z) = \sigma^2 \frac{B(z) B(\frac{1}{z})}{A(z) A(\frac{1}{z})}.}$$

Remark: If $a_1, \cdots, a_p, b_1, \cdots, b_q$ are complex,

$$P(z) = \sigma^2 \frac{B(z) B^*(\frac{1}{z^*})}{A(z) A^*\left(\frac{1}{z^*}\right)}$$

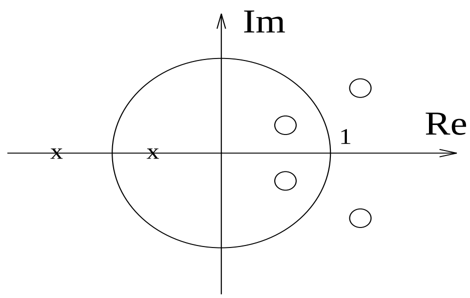
Consider

$$P(z) = \sigma^2 \frac{B(z)B(\frac{1}{z})}{A(z)A(\frac{1}{z})}.$$

Remark: • If α is zero for $P(z)$, so is $\frac{1}{\alpha}$.

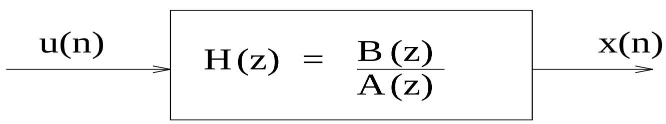
• If β is a pole for $P(z)$, so is $\frac{1}{\beta}$.

• Since the $a_1, \cdots, a_p, b_1, \cdots, b_q$ are real, the poles and zeroes of $P(z)$ occur in complex conjugate pairs.



Remark:

- If poles of $\frac{1}{A(z)}$ inside unit circle, $H(z) = \frac{B(z)}{A(z)}$ is BIBO stable.
- If zeroes of $B(z)$ inside unit circle, $H(z) = \frac{B(z)}{A(z)}$ is minimum phase.
- We chose $H(z)$ so that both its zeroes and poles are inside unit circle.



Stable and Minimum Phase system

Relationships Among Models

- An MA(q) or ARMA(p,q) model is equivalent to an AR(∞).
- An AR(p) or ARMA(p,q) model is equivalent to an MA(∞) model

Ex:

$$H(z) = \frac{1+0.9z^{-1}}{1+0.8z^{-1}} = \quad \text{ARMA(1,1)}$$

$$\begin{aligned} H(z) &= \frac{1}{(1+0.8z^{-1})\frac{1}{(1+0.9z^{-1})}} \\ &= \frac{1}{(1+0.8z^{-1})(1-0.9z^{-1}+0.81z^{-2}+\cdots)} \\ &= \quad \text{AR}(\infty). \end{aligned}$$

Remark:Let ARMA(p,q) $= \frac{B(z)}{A(z)} = \frac{1}{C(z)} =$ AR(∞).

From $a_1, \cdots, a_p, b_1, \cdots, b_q$, we can find $c_1, c_2, \cdots$ and vice versa.

$$\text{Since} \quad \frac{B(z)}{A(z)} = \frac{1}{C(z)} \Rightarrow B(z)C(z) = A(z)$$

$$\Rightarrow \left[1 + b_1 z^{-1} + \cdots + b_q z^{-q}\right] \left[1 + c_1 z^{-1} + \cdots\right]$$

$$= \left[1 + a_1 z^{-1} + \cdots + a_p z^{-p}\right]$$

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ c_1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ c_p & \ddots & \ddots & \ddots & \\ c_{p+1} & \ddots & \ddots & \ddots & 1 \\ \vdots & \ddots & \ddots & \ddots & c_1 \\ \vdots & & & & c_p \\ & & & & \vdots \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \\ 0 \\ \vdots \end{bmatrix} \qquad (\diamond)$$

$$\begin{bmatrix} c_{p+1} & c_p & \cdots & c_{p-q+1} \\ \vdots & \ddots & \ddots & \vdots \\ c_{p+q} & & \ddots & c_p \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ \vdots \\ b_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} c_p & \cdots & c_{p-q+1} \\ \vdots & \ddots & \\ c_{p+q-1} & \cdots & c_p \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_q \end{bmatrix} = - \begin{bmatrix} c_{p+1} \\ \vdots \\ c_{p+q} \end{bmatrix} . (*)$$

Remark: Once $b_1, \cdots, b_q$ are computed with $(*)$ $\quad a_1, \cdots, a_p$ can be computed with $(\diamond)$.

Computing Coefficients from $r(k)$.

AR signals.

Let $\frac{1}{A(z)} = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots$

$x(n) = \frac{1}{A(z)} u(n) = u(n) + \alpha_1 u(n-1) + \cdots$

$$\begin{cases} E\left[x(n)u(n)\right] = \sigma^2 \\ E\left[x(n-k)u(n)\right] = 0, \ \text{k} \geq 1 \end{cases}$$

Since $A(z)x(n) = u(n)$

$x(n) + a_1 x(n-1) + \cdots + a_p x(n-p) = u(n)$

$$\begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \cdots \\ a_p \end{bmatrix} = u(n)$$

$\underline{\mathrm{k} = 0}$,

$$E\left\{ x(n) \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} \right\} = \sigma^2.$$

$$\Rightarrow \begin{bmatrix} r(0) & r(-1) & \cdots & r(-p) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \sigma^2. \quad (*)$$

k $\geq$ 1,

$$E\left\{x(n-k)\begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-p) \end{bmatrix}\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}\right\} = 0.$$

$$\Rightarrow \begin{bmatrix} r(k) & r(k-1) & \cdots & r(k-p) \end{bmatrix}\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = 0. \quad (**)$$

$$\Rightarrow \begin{bmatrix} r(0) & r(-1) & \cdots & r(-p) \\ r(1) & r(0) & \cdots & r(-p+1) \\ \vdots & \ddots & \ddots & \\ r(p) & r(p-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

$$\mathbf{Ra} = -\mathbf{r} \Leftrightarrow \begin{bmatrix} r(0) & \cdots & r(-p+1) \\ \vdots & \ddots & \\ r(p-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r(1) \\ \vdots \\ r(p) \end{bmatrix}.$$

Remarks:

- When we only have N samples, $\{r(k)\}$ is not available. $\{\hat{r}(k)\}$ may be used to replace $\{r(k)\}$ to obtain $\hat{a}_1, \cdots, \hat{a}_p$.

$\Rightarrow$ This is the Yule - Walker Method.

- **R** is a positive semidefinite matrix. **R** is positive definite unless $x(n)$ is a sum of less than $\lfloor \frac{p}{2} \rfloor$ sinusoids.
- **R** is Toeplitz.
- Levinson - Durbin algorithm is used to solve for a efficiently
- AR models are most frequently used in practice.
- Estimation of AR parameters is a well-established topic.

Remarks:

- If $\{\hat{r}(k)\}$ is a positive definite sequence and if $a_1, \cdots, a_p$ are found by solving $\mathbf{Ra} = -\mathbf{r}$, then the roots of polynomial $1 + a_1 z^{-1} + \cdots + a_p z^{-p}$ are inside the unit circle.
- The AR system thus obtained is BIBO stable
- Biased estimate $\{\hat{r}(k)\}$ should be used in YW-equation to obtain a stable AR system:

Efficient Methods for solving

$$\mathbf{Ra} = -\mathbf{r} \quad \text{or} \quad \hat{\mathbf{R}}\hat{\mathbf{a}} = -\hat{\mathbf{r}}$$

- Levinson - Durbin Algorithm.
- Delsarte - Genin Algorithm.
- Gohberg - Semencul Formula for $\mathbf{R}^{-1}$ or $\hat{\mathbf{R}}^{-1}$

(Sometimes, we may be interested in not only $\mathbf{a}$ but also $\mathbf{R}^{-1}$)

Levinson - Durbin Algorithm (LDA)

Let

$$\mathbf{R}_{n+1} = \begin{bmatrix} r(0) & r(1) & \cdots & r(n) \\ r(1) & r(0) & & \\ \vdots & & \ddots & \\ r(n) & r(n-1) & & r(0) \end{bmatrix}, \qquad \text{(real signal)}$$

$$n = 1, 2, \cdots, p$$

Let $\quad \boldsymbol{\theta}_n = \begin{bmatrix} a_{n,1} \\ \vdots \\ a_{n,n} \end{bmatrix},$

LDA solves

$$\mathbf{R}_{n+1} \begin{bmatrix} 1 \\ \cdots \\ \boldsymbol{\theta}_n \end{bmatrix} = \begin{bmatrix} \delta_n \\ \cdots \\ \mathbf{0} \end{bmatrix}$$

recursively in n, starting from $n = 1$.

Remark:

For $n = 1, 2, \cdots, p$,

- LDA needs $\approx p^2$ flops
- Regular matrix inverses need $\approx p^4$ flops.

Let $\mathbf{A}$ = Symmetric and Toeplitz.

$$\text{Let} \quad \tilde{\mathbf{b}} = \begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_1 \end{bmatrix}, \text{with} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Then if $\mathbf{c} = \mathbf{A}\mathbf{b}$

$$\Rightarrow \tilde{\mathbf{c}} = \mathbf{A}\tilde{\mathbf{b}}$$

Proof:

$$\mathbf{A} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}$$

$$\begin{aligned} \Rightarrow \mathbf{A}_{ij} &= a_{|i-j|} \\ \tilde{\mathbf{c}}_i &= \mathbf{c}_{n-i+1} = \sum_{k=1}^{n} \mathbf{A}_{n-i+1,k} b_k \\ &= \sum_{k=1}^{n} a_{|n-i+1-k|} b_k \\ &= \sum_{m=1}^{n} a_{|m-i|} b_{n-m+1} = \sum_{m=1}^{n} \mathbf{A}_{m,i} \tilde{b}_m \quad (m = n-k+1) \\ &= (\mathbf{A}\tilde{\mathbf{b}})_i \end{aligned}$$

Consider:

$$\mathbf{R}_{n+2}\begin{bmatrix}1\\ \boldsymbol{\theta}_n\\ 0\end{bmatrix} = \left[\begin{array}{cc:c} \mathbf{R}_{n+1} & & r(n+1)\\ & & r(n)\\ & & \vdots\\ & & r(1)\\ r(n+1) & \cdots & r(0)\end{array}\right]\begin{bmatrix}1\\ \boldsymbol{\theta}_n\\ \cdots\\ 0\end{bmatrix} = \begin{bmatrix}\delta_n\\ \mathbf{0}\\ \cdots\\ \alpha_n\end{bmatrix}$$

$$\text{Let} \quad \mathbf{r}_n = \begin{bmatrix} r(1)\\ \vdots\\ r(n)\end{bmatrix}.$$

Then $\alpha_n = r(n+1) + \boldsymbol{\theta}_n{}^T\tilde{\mathbf{r}}_n$.

Result:

Let $\quad k_{n+1} = -\frac{\alpha_n}{\delta_n}. \quad$ Then

$$\boldsymbol{\theta}_{n+1} = \begin{bmatrix} \boldsymbol{\theta}_n \\ 0 \end{bmatrix} + k_{n+1} \begin{bmatrix} \tilde{\boldsymbol{\theta}}_n \\ 1 \end{bmatrix}.$$

$$\delta_{n+1} = \delta_n (1 - k_{n+1}^2)$$

Proof:

$$\mathbf{R}_{n+2}\begin{bmatrix}1\\ \boldsymbol{\theta}_{n+1}\end{bmatrix} = \mathbf{R}_{n+2}\left\{\begin{bmatrix}1\\ \boldsymbol{\theta}_n\\ 0\end{bmatrix} + k_{n+1}\begin{bmatrix}0\\ \tilde{\boldsymbol{\theta}}_n\\ 1\end{bmatrix}\right\}$$

$$= \begin{bmatrix}\delta_n\\ \mathbf{0}\\ \alpha_n\end{bmatrix} + k_{n+1}\begin{bmatrix}\alpha_n\\ \mathbf{0}\\ \delta_n\end{bmatrix}$$

$$= \begin{bmatrix}\delta_n + k_{n+1}\alpha_n\\ \mathbf{0}\\ \alpha_n + k_{n+1}\delta_n\end{bmatrix} = \begin{bmatrix}\delta_{n+1}\\ \mathbf{0}\\ 0\end{bmatrix}.$$

LDA: Initialization:

$$n = 1 : \quad \mathbf{R}_2 = \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \delta_1 \\ 0 \end{bmatrix}$$

$$\boxed{\theta_1 = -\frac{r(1)}{r(0)}} \quad O(1) \quad \text{flops}$$

$$\boxed{\delta_1 = r(0) - \frac{r^2(1)}{r(0)}} \quad O(1) \quad \text{flops}$$

$$\boxed{k_1 = \theta_1}$$

For $n = 1, 2, \cdots, p-1$, do:

$$k_{n+1} = -\frac{r(n+1) + \boldsymbol{\theta}_n^T \tilde{\mathbf{r}}_n}{\delta_n} \qquad \sim n \quad \text{flops}$$

$$\delta_{n+1} = \delta_n(1 - k_{n+1}^2) \qquad O(1) \quad \text{flops}$$

$$\boldsymbol{\theta}_{n+1} = \begin{bmatrix} \boldsymbol{\theta}_n \\ 0 \end{bmatrix} + k_{n+1} \begin{bmatrix} \tilde{\boldsymbol{\theta}}_n \\ 1 \end{bmatrix}. \quad \sim n \quad \text{flops}$$

Ex:

$$\begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ 0 \end{bmatrix}.$$

Straightforward Solution:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}^{-1} \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}$$

$$= -\frac{1}{(1-\rho^2)} \begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix} \begin{bmatrix} \rho \\ \rho^2 \end{bmatrix}$$

$$= \begin{bmatrix} -\rho \\ 0 \end{bmatrix} \Rightarrow \sigma^2 = 1 - \rho^2.$$

LDA: Initialization:

$$\begin{cases} \theta_1 = -\frac{r(1)}{r(0)} = -\frac{\rho}{1} = -\rho \\ \delta_1 = r(0) - \frac{r^2(1)}{r(0)} = 1 - \rho^2. \\ k_1 = \theta_1 = -\rho. \end{cases}$$

$$r_1 = \rho,$$

$$\begin{aligned} k_2 &= -\frac{r(2) + \theta_1^T \tilde{r}_1}{\delta_1} \\ &= -\frac{\rho^2 + (-\rho)\rho}{1 - \rho^2} = 0 \\ \delta_2 &= \delta_1(1 - k_2^2) = (1 - \rho^2)(1 - 0^2) \\ &= 1 - \rho^2 = \sigma^2 \end{aligned}$$

$$
\begin{aligned}
\boldsymbol{\theta}_2 &= \begin{bmatrix} \theta_1 \\ 0 \end{bmatrix} + k_2 \begin{bmatrix} \tilde{\theta}_1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} -\rho \\ 0 \end{bmatrix} + 0 \begin{bmatrix} -\rho \\ 1 \end{bmatrix} = \begin{bmatrix} -\rho \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}
\end{aligned}
$$

Properties of LDA:

- $|k_n| < 1, \quad n = 1, 2, \cdots, p, \quad$ and $\quad r(0) > 0, \text{iff}$

$$A_n(z) = 1 + a_{n,1} z^{-1} + \cdots + a_{n,n} z^{-n} = 0$$

has roots inside the unit circle.

- $|k_n| < 1, \quad n = 1, 2, \cdots, p, \quad$ and $\quad r(0) > 0 \quad \text{iff} \quad \mathbf{R}_{n+1} > 0$

Proof (for the second property above only): We first use induction to prove:

$$\underbrace{\begin{bmatrix} 1 & a_{n,1} & \cdots & a_{n,n} \\ & \ddots & & \vdots \\ 0 & & 1 & a_{1,1} \\ & & & 1 \end{bmatrix}}_{\mathbf{U}_{n+1}^T} \underbrace{\begin{bmatrix} r(0) & \cdots & r(n) \\ & \ddots & \\ r(n) & & r(0) \end{bmatrix}}_{\mathbf{R}_{n+1}} \underbrace{\begin{bmatrix} 1 & & & \\ a_{n,1} & 1 & & \\ \vdots & & \ddots & \\ a_{n,n} & \cdots & a_{1,1} & 1 \end{bmatrix}}_{\mathbf{U}_{n+1}}$$

$$= \underbrace{\begin{bmatrix} \delta_n & & & \\ & \ddots & & \mathbf{0} \\ \mathbf{0} & & \delta_1 & \\ & & & r(0) \end{bmatrix}}_{\mathbf{D}_{n+1}} \qquad (*)$$

<u>n = 1 :</u>

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_{1,1} \end{bmatrix} = \begin{bmatrix} \delta_1 \\ 0 \end{bmatrix}.$$

$$\begin{aligned} \Rightarrow \quad & \begin{bmatrix} 1 & a_{1,1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a_{1,1} & 1 \end{bmatrix} \\ = \quad & \begin{bmatrix} 1 & a_{1,1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 & r(1) \\ 0 & r(0) \end{bmatrix} \\ = \quad & \begin{bmatrix} \delta_1 & 0 \\ 0 & r(0) \end{bmatrix}. \end{aligned}$$

Suppose $(*)$ is true for $n = k-1$, i.e.,

$$\mathbf{U}_k^T \mathbf{R}_k \mathbf{U}_k = \mathbf{D}_k.$$

Consider $n = k$:

$$\mathbf{U}_{k+1}^T \mathbf{R}_{k+1} \mathbf{U}_{k+1} = \begin{bmatrix} 1 & \boldsymbol{\theta}_k^T \\ \mathbf{0} & \mathbf{U}_k^T \end{bmatrix} \begin{bmatrix} r(0) & \mathbf{r}_k^T \\ \mathbf{r}_k & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{\theta}_k & \mathbf{U}_k \end{bmatrix}$$

$$= \begin{bmatrix} r(0) + \boldsymbol{\theta}_k^T \mathbf{r}_k & \mathbf{r}_k^T + \boldsymbol{\theta}_k^T \mathbf{R}_k \\ \mathbf{U}_k^T \mathbf{r}_k & \mathbf{U}_k \mathbf{R}_k \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{\theta}_k & \mathbf{U}_k \end{bmatrix}$$

Since $\mathbf{R}_{k+1} \begin{bmatrix} 1 \\ \boldsymbol{\theta}_k \end{bmatrix} = \begin{bmatrix} \delta_k \\ \mathbf{0} \end{bmatrix}$

$$\Rightarrow \begin{bmatrix} r(0) & \mathbf{r}_k^T \\ \mathbf{r}_k & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{\theta}_k \end{bmatrix} = \begin{bmatrix} \delta_k \\ \mathbf{0} \end{bmatrix}$$

$$\Rightarrow r(0) + \mathbf{r}_k^T \boldsymbol{\theta}_k = \delta_k \quad \Rightarrow r(0) + \boldsymbol{\theta}_k^T \mathbf{r}_k \quad = \delta_k$$

$$\underline{r_k + \mathbf{R}_k \boldsymbol{\theta}_k = 0} \quad \Rightarrow \quad \mathbf{r}_k^T + \boldsymbol{\theta}_k^T \mathbf{R}_k^T$$

$$= \quad \mathbf{r}_k^T + \boldsymbol{\theta}_k^T \mathbf{R}_k \quad = 0$$

$$\begin{aligned} \Rightarrow \mathbf{U}_{k+1}^T \mathbf{R}_{k+1} \mathbf{U}_{k+1} &= \begin{bmatrix} \delta_k & \mathbf{0} \\ \mathbf{U}_k^T \mathbf{r}_k & \mathbf{U}_k^T \mathbf{R}_k \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \boldsymbol{\theta}_k & \mathbf{U}_k \end{bmatrix} \\ &= \begin{bmatrix} \delta_k & \mathbf{0} \\ \mathbf{U}_k^T \mathbf{r}_k + \mathbf{U}_k^T \mathbf{R}_k \boldsymbol{\theta}_k & \mathbf{U}_k^T \mathbf{R}_k \mathbf{U}_k \end{bmatrix} \\ &= \begin{bmatrix} \delta_k & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_k \end{bmatrix} = \mathbf{D}_{k+1} \end{aligned}$$

$$\Rightarrow \quad \mathbf{U}_{n+1}^T \mathbf{R}_{n+1} \mathbf{U}_{n+1} = \mathbf{D}_{n+1}.$$

$$\Rightarrow (*) \quad \text{proven !}$$

$$\text{Since} \quad \mathbf{U}_{n+1}^{-1} \mathbf{R}_{n+1}^{-1} \left(\mathbf{U}_{n+1}^T\right)^{-1} = \mathbf{D}_{n+1}^{-1},$$

$$\Rightarrow \qquad \boxed{\mathbf{R}_{n+1}^{-1} = \mathbf{U}_{n+1} \mathbf{D}_{n+1}^{-1} \mathbf{U}_{n+1}^T.}$$

$\mathbf{U}_{n+1}\mathbf{D}_{n+1}^{-\frac{1}{2}}$ is called <u>Cholesky Factor</u> of $\mathbf{R}_{n+1}^{-1}$

- Consider the determinant of $\mathbf{R}_{n+1}$:

$$\det(\mathbf{R}_{n+1}) = \det(\mathbf{D}_{n+1}) = r(0)\Pi_{k=1}^n \delta_k$$

$$\Rightarrow \quad \det(\mathbf{R}_{n+1}) = \delta_n \det(\mathbf{R}_n)$$

$\Rightarrow \quad \mathbf{R}_{n+1} > 0, \quad n = 1, 2, \cdots, p, \quad \text{iff} \quad r(0) > 0$

and $\quad \delta_k > 0, \quad k = 1, 2, \cdots, p.$

Recall

$$\delta_{n+1} = \delta_n(1 - k_{n+1}^2).$$

<u>If $\mathbf{R}_{n+1} > 0$</u>,

$$\Rightarrow r(0) > 0, \quad \delta_n > 0, \quad n = 1, 2, \cdots, p,$$

$$k_{n+1}^2 = \frac{\delta_n - \delta_{n+1}}{\delta_n}$$

$$\text{Since} \quad \delta_n - \delta_{n+1} < \delta_n,$$

$$k_{n+1}^2 < 1 \quad \Rightarrow \quad |k_{n+1}| < 1.$$

<u>If $\quad |k_n| < 1, \quad r(0) > 0$</u>,

$$\Rightarrow \quad k_{n+1}^2 < 1.$$

$$\Rightarrow \begin{cases} \delta_0 = r(0) > 0, \\ \delta_{n+1} = \delta_n(1 - k_{n+1}^2) > 0, \quad n = 1, 2, \cdots, p-1 \end{cases}$$

MA Signals:

$$
\begin{aligned}
x(n) &= B(z)u(n) \\
&= u(n) + b_1 u(n-1) + \cdots + b_q u(n-q) \\
r(k) &= E\left[x(n)x(n-k)\right] \\
&= E\{[u(n) + \cdots + b_q u(n-q)] \\
&\quad [u(n-k) + \cdots + b_q u(n-q-k)]\}
\end{aligned}
$$

$$|k| > q : \qquad r(k) = 0$$

$$
\begin{aligned}
|k| < q : \quad r(k) &= \sigma^2 \sum_{l=0}^{q-k} b_l b_{l+k}, \quad q > k \geq 0 \\
r(k) &= r(-k). \qquad -q < k < 0
\end{aligned}
$$

$b_0 = 1, b_1, \cdots, b_q =$ real.

$$\Rightarrow \qquad P(\omega) = \sum_{k=-q}^{q} r(k) e^{-j\omega k}.$$

Remarks: • Estimating $b_1, \cdots, b_q$ is a nonlinear problem.

A simple estimator is $$\hat{P}(\omega) = \sum_{k=-q}^{q} \hat{r}(k) e^{-j\omega k}.$$

* This is exactly Blackman - Tukey method with rectangular window of length $2q+1$.

* No matter whether $\hat{r}(k)$ is biased or unbiased estimate, this $\hat{P}(\omega)$ may be < 0 .

* When unbiased $\hat{r}(k)$ is used, $\hat{P}(\omega)$ is unbiased.

* To ensure $\hat{P}(\omega) \geq 0$, $\forall\, \omega$, we may use biased $\hat{r}(k)$ and a window with $W(\omega) \geq 0, \quad \forall\, \omega$. For this case, $\hat{P}(\omega)$ is biased.

This is again exactly BT-method.

• A most used MA spectral estimator is based on a Two-Stage Least Squares Method. See the discussions on ARMA later.

ARMA Signals: (Also called Pole -Zero Model).

$$(1 + a_1 z^{-1} + \cdots + a_p z^{-p})x(n) = (1 + b_1 z^{-1} + \cdots + b_q z^{-q})u(n).$$

Let us write $x(n)$ as MA(∞):

$$x(n) = u(n) + h_1 u(n-1) + h_2 u(n-2) + \cdots$$

$$\Rightarrow \begin{cases} E\left[x(n)u(n)\right] = \sigma^2. \\ E\left[u(n)x(n-k)\right] = 0, \quad k \geq 1 \end{cases}$$

ARMA model can be written as

$$\begin{bmatrix} 1 & a_1 & \cdots & a_p \end{bmatrix} \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-p) \end{bmatrix} = \begin{bmatrix} 1 & b_1 & \cdots & b_q \end{bmatrix} \begin{bmatrix} u(n) \\ u(n-1) \\ \vdots \\ u(n-q) \end{bmatrix}$$

- Next we shall multiply both sides by $x(n-k)$ and take $E\{.\}$.

k= 0:

$$\begin{bmatrix} 1 & a_1 & \cdots & a_p \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(p) \end{bmatrix} = \begin{bmatrix} 1 & b_1 & \cdots & b_q \end{bmatrix} \begin{bmatrix} \sigma^2 \\ \sigma^2 h_1 \\ \vdots \\ \sigma^2 h_q \end{bmatrix}$$

k = 1:

$$\begin{bmatrix} 1 & a_1 & \cdots & a_p \end{bmatrix} \begin{bmatrix} r(-1) \\ r(0) \\ \vdots \\ r(p-1) \end{bmatrix} = \begin{bmatrix} 1 & b_1 & \cdots & b_q \end{bmatrix} \begin{bmatrix} 0 \\ \sigma^2 \\ \sigma^2 h_1 \\ \vdots \\ \sigma^2 h_{q-1} \end{bmatrix}$$

$\vdots$

k $\geq$ q+1

$$\begin{bmatrix} 1 & a_1 & \cdots & a_p \end{bmatrix} \begin{bmatrix} r(-k) \\ r(-k+1) \\ \vdots \\ r(-k+p) \end{bmatrix} = \begin{bmatrix} 1 & b_1 & \cdots & b_q \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}.$$

$$\Rightarrow \begin{bmatrix} r(-(q+1)) & r(-q) & \cdots & r(-(q+1)+p) \\ r(-(q+2)) & r(-(q+1)) & \cdots & r(-(q+2)+p) \\ \vdots & & \ddots & \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \mathbf{0}.$$

This is the modifed YW - Equation

To solve for $a_1, \cdots, a_p$ we need p equations. Using $r(k) = r(-k)$ gives

$$\begin{bmatrix} r(q+1) & r(q) & \cdots & r(q-p+1) \\ r(q+2) & r(q+1) & \cdots & r(q-p+2) \\ \vdots & & \ddots & \\ r(q+p) & r(q+p-1) & \cdots & r(q) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = 0.$$

$$\Rightarrow \begin{bmatrix} r(q) & \cdots & r(q-p+1) \\ r(q+1) & \ddots & \\ \vdots & & \\ r(q+p-1) & \cdots & r(q) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r(q+1) \\ r(q+2) \\ \vdots \\ r(q+p) \end{bmatrix}$$

Remarks:

(1) Replacing $\hat{r}(k)$ for $r(k)$ above, we can solve for $\hat{a}_1, \cdots, \hat{a}_p$.

(2) The matrix on the left side

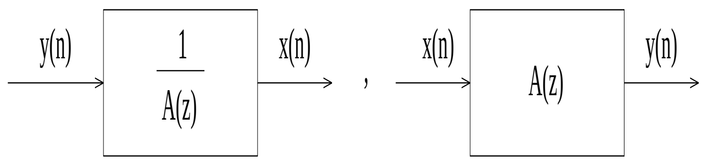
- is nonsingular under mild conditions.
- is Toeplitz.
- is NOT symmetric.
- Levinson - type fast algorithms exist.

What about the MA part of the ARMA PSD?

$$\text{Let} \quad y(n) = (1 + b_1 z^{-1} + \cdots + b_q z^{-q}) u(n).$$

The ARMA model becomes

$$(1 + a_1 z^{-1} + \cdots + a_p z^{-p}) x(n) = y(n)$$



$$P_x(\omega) = \left| \frac{1}{A(\omega)} \right|^2 P_y(\omega).$$

Let γ_k be the autocorrelation function of $y(n)$. Then (see MA signals).

$$P_y(\omega) = \sum_{k=-q}^{q} \gamma_k e^{-j\omega k}$$

$$\begin{aligned}\gamma_k &= E\left[y(n)y(n-k)\right] \\ &= E\left[A(z)x(n)A(z)x(n-k)\right] \\ &= E\left[\sum_{i=0}^{p} a_i x(n-i) \sum_{j=0}^{p} a_j x(n-j-k)\right] \\ &= \sum_{i=0}^{p}\sum_{j=0}^{p} a_i a_j r(k+j-i).\end{aligned}$$

Since $\hat{a}_1, \cdots, \hat{a}_p$ may be computed with the modified YW- Method

$$\begin{cases} \hat{\gamma}_k = \sum_{i=0}^{p}\sum_{j=0}^{p} \hat{r}(k+j-i)\hat{a}_i\hat{a}_j, \quad \hat{a}_0 \overset{\triangle}{=} 1, \quad k = 0, 1, \cdots, q \\ \hat{\gamma}_{-k} = \gamma_k. \end{cases}$$

ARMA PSD Estimate:

$$\hat{P}(\omega) = \frac{\sum_{k=-q}^{q} \hat{\gamma}_k e^{-j\omega k}}{\left|\hat{A}(\omega)\right|^2}$$

Remarks:

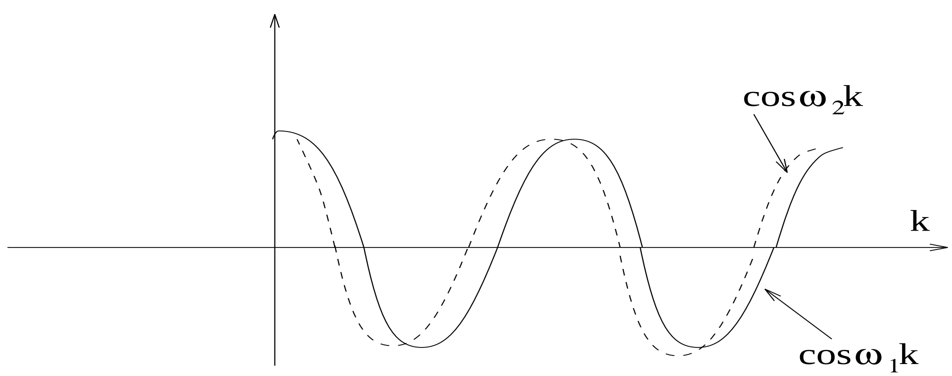
- This method is called modified YW ARMA Spectral Estimator
- $\hat{P}(\omega)$ is not guaranteed to be $\geq 0, \quad \forall\, \omega$, due to the MA part.
- The AR estimates $\hat{a}_1, \cdots, \hat{a}_p$ have reasonable accuracy if the ARMA poles and zeroes are well inside the unit circle.
- Very poor estimates $\hat{a}_1, \cdots, \hat{a}_p$ occur when ARMA poles and zeroes are closely-spaced and nearby unit circle. (This is narrowband signal case).

Ex: Consider

$$x(n) = \cos(\omega_1 n + \phi_1) + \cos(\omega_2 n + \phi_2),$$

where ϕ_1 and ϕ_2 are independent and uniformly distributed on $[0,2\pi]$.

$$r(k) = \frac{1}{2}\cos(\omega_1 k) + \frac{1}{2}\cos(\omega_2 k).$$



Note that when $\omega_1 \approx \omega_2$, large values of k are needed to distinguish $\cos(\omega_1 k)$ and $\cos(\omega_2 k)$.

Remark: This comment is true for both AR and ARMA models.

Overdetermined Modified Yule - Walker Equation $(M > p)$

$$\begin{bmatrix} \hat{r}(q) & \cdots & \hat{r}(q-p+1) \\ \vdots & & \vdots \\ \hat{r}(q+p-1) & \cdots & \hat{r}(q) \\ \vdots & & \vdots \\ \hat{r}(q+M-1) & \cdots & \hat{r}(q+M-p) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_p \end{bmatrix} \approx - \begin{bmatrix} \hat{r}(q+1) \\ \vdots \\ \hat{r}(q+p) \\ \vdots \\ \hat{r}(q+M) \end{bmatrix}$$

Remarks:

- The overdetermined linear equations may be solved with Least Squares or Total Least Squares Methods.
- M should be chosen based on the trade-off between information contained in the large lags of $\hat{r}(k)$ and the accuracy of $\hat{r}(k)$.
- Overdetermined YW -equation may also be obtained for AR signals.

Solving Linear Equations:

Consider $\mathbf{A}^{m\times n}\mathbf{x}^{n\times 1} = \mathbf{b}^{m\times 1}$.

• When $m = n$ and $\mathbf{A}$ is full rank, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

• When $m > n$ and $\mathbf{A}$ is full rank n, then the solution exists if $\mathbf{b}$ is in the n-dimensional subspace of the m-dimensional space that is determined by the columns in $\mathbf{A}$.

Ex:

$$\mathbf{A} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\text{If} \quad \mathbf{b} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \mathbf{x} = 3.$$

$$If \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{x} =? \quad \text{does not exist !}$$

Least Squares (LS) Solution for Overdetermined Equations:

- Objective of LS solution:

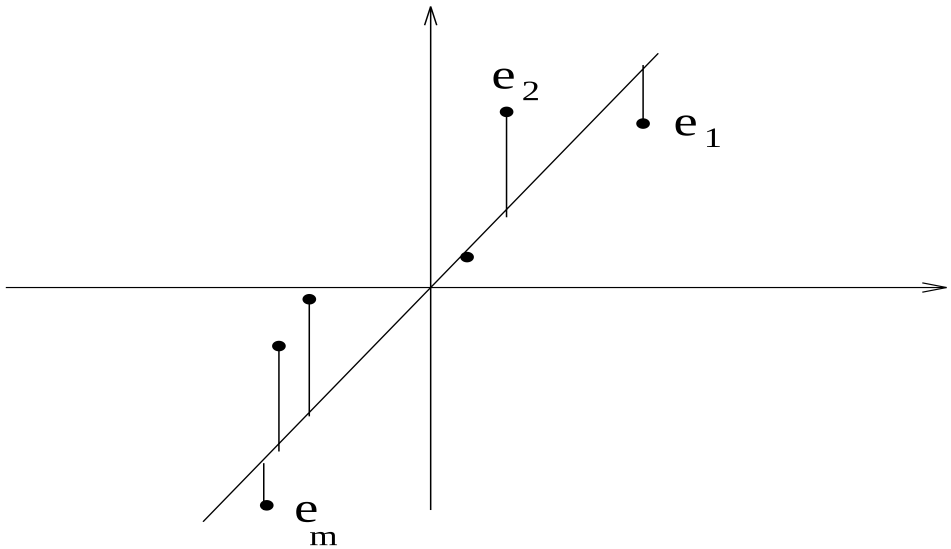
$$\text{Let} \quad \mathbf{e} = \mathbf{A}\mathbf{x} - \mathbf{b}$$

Find $\mathbf{x}_{LS}$ so that $\mathbf{e}^H\mathbf{e}$ is minimized.

$$\text{Let} \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix}$$

$$\text{Euclidean Norm} = \mathbf{e}^H\mathbf{e} = |e_1|^2 + |e_2|^2 + \cdots + |e_m|^2$$

Ex:



Remarks: • $\mathbf{A}\mathbf{x}_{LS} = \mathbf{b} + \mathbf{e}_{LS}$

• We see that $\mathbf{x}_{LS}$ is found by perturbing $\mathbf{b}$ so that a solution exists.

$$
\begin{aligned}
\mathbf{e}^H\mathbf{e} &= (\mathbf{Ax}-\mathbf{b})^H(\mathbf{Ax}-\mathbf{b}) \\
&= \mathbf{x}^H\mathbf{A}^H\mathbf{Ax} - \mathbf{x}^H\mathbf{A}^H\mathbf{b} - \mathbf{b}^H\mathbf{Ax} + \mathbf{b}^H\mathbf{b} \\
&= \left[\mathbf{x} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}\right]^H (\mathbf{A}^H\mathbf{A}) \left[\mathbf{x} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}\right] \\
&+ \left[\mathbf{b}^H\mathbf{b} - \mathbf{b}^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}\right]
\end{aligned}
$$

Remark: • The 2^{nd} term above is independent of $\mathbf{x}$.

• $\mathbf{e}^H\mathbf{e}$ is minimized if

$$\boxed{\mathbf{x} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b}}$$

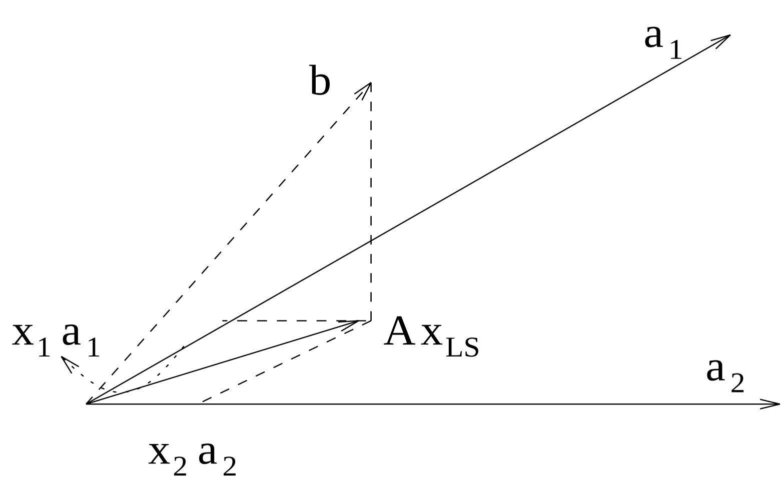
LS Solution

Illustration of LS solution:

Let

$$\mathbf{A} = [\mathbf{a}_1 \;\vdots\; \mathbf{a}_2].$$

$$\mathbf{x}_{LS} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Ex:

$$\mathbf{A} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_{LS} = ?$$

$$\begin{aligned} \mathbf{x}_{LS} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b} \\ &= \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 \end{aligned}$$

$$\mathbf{A}\mathbf{x}_{LS} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} (1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$\mathbf{e}_{LS} = \mathbf{A}\mathbf{x}_{LS} - \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Computational Aspects of LS

- **Solving Normal Equations**

$$\boxed{\left(\mathbf{A}^H\mathbf{A}\right)\mathbf{x}_{LS} = \mathbf{A}^H\mathbf{b}.} \quad (1)$$

This equation is called Normal equation.

Let

$$\mathbf{A}^H\mathbf{A} = \mathbf{C}, \quad \mathbf{A}^H\mathbf{b} = \mathbf{g}.$$

$$\mathbf{C}\mathbf{x}_{LS} = \mathbf{g}, \quad \text{where } \mathbf{C} \text{ is positive definite.}$$

Cholesky Decomposition:

$$\mathbf{C} = \mathbf{L}\mathbf{D}\mathbf{L}^H,$$

where $\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}$ (Lower Triangular Matrix)

$$\mathbf{D} = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}, \quad d_i > 0.$$

Back - Substitution to solve:

$$\mathbf{LDL}^H\mathbf{x}_{LS} = \mathbf{g}$$

Let

$$\mathbf{y} = \mathbf{DL}^H\mathbf{x}_{LS}.$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix}$$

$$\begin{cases} y_1 = g_1 \\ y_2 = g_2 - l_{21}y_1 \\ \vdots \\ y_k = g_k - \sum_{j=1}^{k-1} l_{kj}y_j, \quad k = 3, \cdots, n. \end{cases}$$

$$\begin{bmatrix} 1 & l_{21}^* & \cdots & l_{n1}^* \\ 0 & 1 & \cdots & l_{n2}^* \\ & & \ddots & \\ 0 & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{L}^H \mathbf{x}_{LS} = \mathbf{D}^{-1}\mathbf{y} = \begin{bmatrix} \frac{y_1}{d_1} \\ \vdots \\ \frac{y_n}{d_n} \end{bmatrix}$$

$$\Rightarrow \begin{cases} x_n = \frac{y_n}{d_n} \\ x_k = \frac{y_k}{d_k} - \sum_{j=k+1}^{n} l_{jk}^* x_j, \quad k = n-1, \cdots \end{cases}$$

Remarks:

- Solving Normal equations may be sensitive to numerical errors.

Ex.

$$\begin{bmatrix} 3 & 3-\delta \\ 4 & 4+\delta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbf{Ax} = \mathbf{b}$$

where δ is a small number.

Exact solution:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\delta} \\ \frac{1}{\delta} \end{bmatrix}$$

Assume that due to truncation errors, $\delta^2 = 0$.

$$\mathbf{A}^H\mathbf{A} \doteq \begin{bmatrix} 25 & 25+\delta \\ 25+\delta & 25+2\delta \end{bmatrix}, \mathbf{A}^H\mathbf{b} = \begin{bmatrix} 1 \\ 1+2\delta \end{bmatrix}.$$

Solution to Normal equation (Note the Big Difference!):

$$\mathbf{x} = \left(\mathbf{A}^H \mathbf{A}\right)^{-1} \mathbf{A}^H \mathbf{b} = \begin{bmatrix} \frac{49}{\delta} + 2 \\ -\frac{49}{\delta} \end{bmatrix}.$$

- **QR Method**: (Numerically more robust).

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Using Householder transformation, we can find an orthonormal matrix $\mathbf{Q}$ $\left(i.e, \quad \mathbf{Q}\mathbf{Q}^H = \mathbf{I}\right)$, such that

$$\begin{bmatrix} T \\ .... \\ 0 \end{bmatrix} \mathbf{x} = \mathbf{Q}\mathbf{A}\mathbf{x} = \mathbf{Q}\mathbf{b} = \begin{bmatrix} z_1 \\ \cdots \\ z_2 \end{bmatrix},$$

where $\mathbf{T}$ is a square, upper triangular matrix, and

$$\min \quad \mathbf{e}^H \mathbf{e} = \mathbf{z}_2^H \mathbf{z}_2$$

$$\Rightarrow \mathbf{T}\mathbf{x}_{LS} = \mathbf{z}_1$$

Back Substitution to find $\mathbf{x}_{LS}$

Ex.

$$\begin{bmatrix} 3 & 3-\delta \\ 4 & 4-\delta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

$$\mathbf{Q} = \frac{1}{5} \begin{bmatrix} 3 & 4 \\ 4 & -3 \end{bmatrix}.$$

$$\mathbf{QAx} = \mathbf{Qb} \quad \text{gives} \begin{bmatrix} 5 & 5+\frac{\delta}{5} \\ 0 & -\frac{7\delta}{5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} \\ -\frac{7}{5} \end{bmatrix}.$$

$$\Rightarrow \begin{cases} x_2 = \frac{1}{\delta} \\ x_1 = -\frac{1}{\delta} \quad \text{(same as exact solution)} \end{cases}$$

Remark: For large number of overdetermined equations, QR method needs about <u>twice</u> as much computation as solving Normal equation in (1).

Total Least Squares (TLS) solution to $\mathbf{Ax} = \mathbf{b}$.

- Recall $\mathbf{x}_{LS}$ is obtained by perturbing $\mathbf{b}$ only, i.e,

$$\mathbf{Ax}_{LS} = \mathbf{b} + \mathbf{e}_{LS}. \qquad \mathbf{e}_{LS}^H \mathbf{e}_{LS} = \quad \text{min.}$$

- $\mathbf{x}_{TLS}$ is obtained by perturbing both $\mathbf{A}$ and $\mathbf{b}$, i.e.,

$$(\mathbf{A} + \mathbf{E}_{TLS})\,\mathbf{x}_{TLS} = \mathbf{b} + \mathbf{e}_{TLS},$$

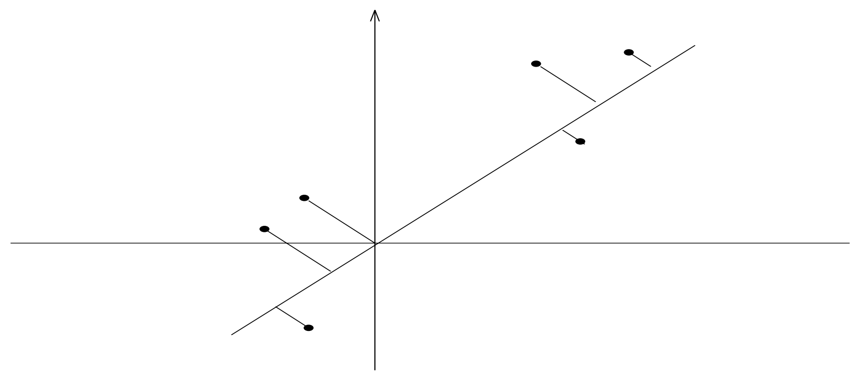
$$||[\mathbf{E}_{TLS} \quad \mathbf{b}_{TLS}]||_F = \quad \text{minimum},$$

where $||.||_F$ is Frobenius matrix norm,

$$||\mathbf{G}||_F = \sum_i \sum_j |g_{ij}|^2,$$

$$g_{ij} = (ij)^{th} \quad \text{element of } \mathbf{G}.$$

Illustration of TLS solution



The straight line is found by minimizing the shortest distance between the line and the points squared

Let $\mathbf{C} = [\mathbf{A} \quad \mathbf{B}]$.

Let the singular value decomposition (SVD) of $\mathbf{C}$ be

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H,$$

Remarks: • The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{C}\mathbf{C}^H$.

Remarks: • The columns in $\mathbf{V}$ are the eigenvectors of $\mathbf{C}^H\mathbf{C}$.

• Both $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, i.e,

$$\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}, \quad \mathbf{V}\mathbf{V}^H = \mathbf{V}^H\mathbf{V} = \mathbf{I}.$$

• $\mathbf{\Sigma}$ is diagonal and the diagonal elements are the $\sqrt{\text{eigenvalues}}$ of $\mathbf{C}^H\mathbf{C}$

$$\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{n+1} \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}.$$

• $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+1} \geq 0, \sigma_i$ are real

Let

$$\mathbf{V} = \begin{array}{c} \\ \\ \end{array}\overset{\begin{array}{cc} n & 1 \end{array}}{\begin{bmatrix} \mathbf{V}_{11} & \vdots & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \vdots & \mathbf{V}_{22} \end{bmatrix}} \begin{array}{c} n \\ 1 \end{array}$$

$$\boxed{\mathbf{x}_{TLS} = -\mathbf{V}_{12}\mathbf{V}_{22}{}^{-1}}$$

Remarks:

- At low SNR, TLS may be better than LS.
- At high SNR, TLS and LS yield similar results.

Markov Estimate:

If the statistics of $\mathbf{e} = \mathbf{A}\mathbf{x}$ - $\mathbf{b}$ is known,
the statistics may be used to obtain better solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$.

ARMA Signals:

Two Stage Least Squares Method

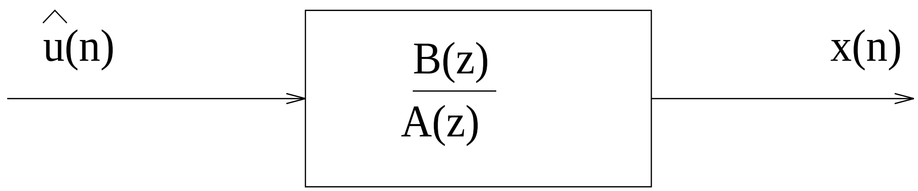
Step 1: Approximate $ARMA(p, q)$ with $AR(L)$ for a large L.

YW Equation may be used to estimate $\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_L$.

$$\hat{u}(n) = x(n) + \hat{a}_1 x(n-1) + \cdots + \hat{a}_L x(n-L).$$

$$\hat{\sigma}^2 = \frac{1}{N-L} \sum_{n=L+1}^{N} \hat{u}^2(n).$$

Step 2: System Identification



$$\text{Let} \quad \mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \quad \hat{u} = \begin{bmatrix} \hat{u}(0) \\ \hat{u}(1) \\ \vdots \\ \hat{u}(N-1) \end{bmatrix}.$$

$$\boldsymbol{\theta} = \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_p \\ b_1 \\ \vdots \\ b_q \end{bmatrix}.$$

$$\mathbf{H} = \begin{bmatrix} x(-1) & \cdots & x(-p) & \hat{u}(-1) & \cdots & \hat{u}(-q) \\ x(0) & \cdots & x(-p+1) & \hat{u}(0) & \cdots & \hat{u}(-q+1) \\ \vdots & & & & & \\ x(N-2) & \cdots & x(N-p-1) & \hat{u}(N-2) & \cdots & \hat{u}(N-q-1) \end{bmatrix}$$

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \hat{\mathbf{u}} \quad \text{(real signals)} .$$

LS Solution $\Rightarrow$

$$\boxed{\hat{\boldsymbol{\theta}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\left(\mathbf{x} - \hat{\mathbf{u}}\right)}$$

Remarks:

- Any elements in $\mathbf{H}$ that are unknown are set to zero.
- QR Method may be used to solve the LS problem.

<u>Step 3:</u>

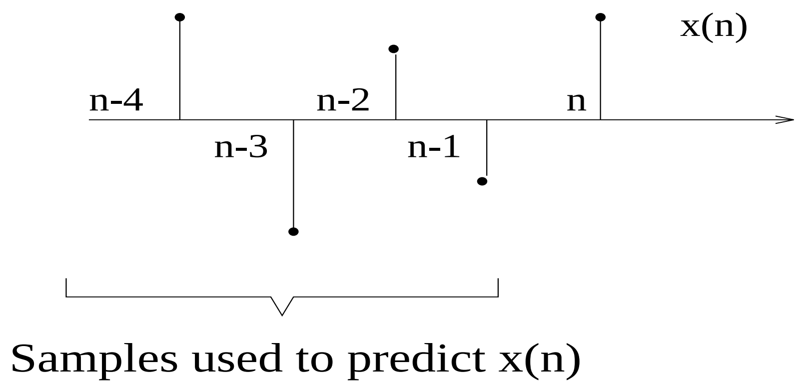
$$\hat{P}(\omega) = \hat{\sigma}^2 \left| \frac{1 + \hat{b}_1 e^{-j\omega} + \cdots + \hat{b}_q e^{-j\omega q}}{1 + \hat{a}_1 e^{-j\omega} + \cdots + \hat{a}_p e^{-j\omega p}} \right|^2$$

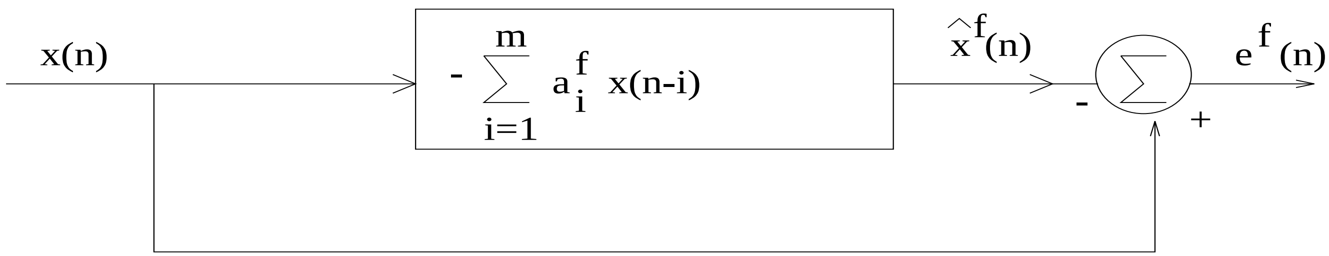
Remark: The difficult case for this method is when ARMA zeroes are near unit circle.

Further Topics on AR Signals:

Linear prediction of AR Processes

- Forward Linear Prediction





$$e^f(n) = x(n) - \hat{x}^f(n).$$

$$\delta^f = E\left[\left(e^f(n)\right)^2\right]$$

Goal: Minimize δ^f

$$\begin{aligned}
\delta^f &= E\left[\left(e^f(n)\right)^2\right] \\
&= E\left[\left(x(n) + \sum_{i=1}^{m} a_i^f x(n-i)\right)^2\right] \\
&= r_{xx}(0) + \sum_{i=1}^{m} a_i^f r_{xx}(i) \\
&+ \sum_{j=1}^{m} a_j^f r_{xx}(j) + \sum_{i=1}^{m}\sum_{j=1}^{m} a_i^f a_j^f r_{xx}(j-i)
\end{aligned}$$

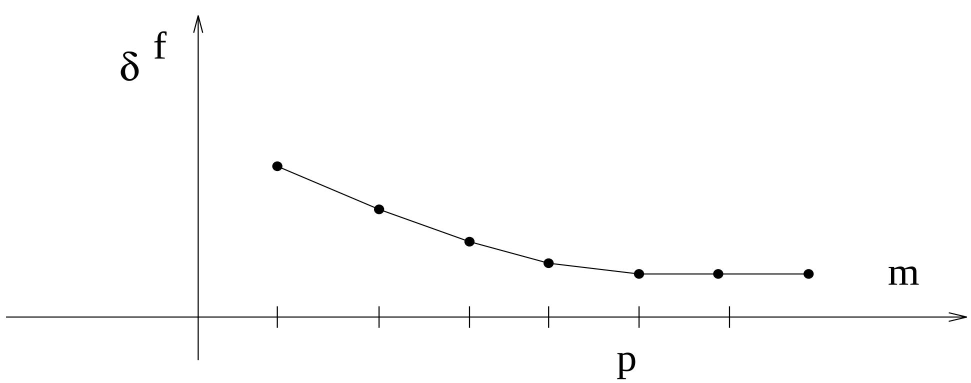
$$\frac{\partial \delta^f}{\partial a_i^f} = 0 \quad \Rightarrow \quad r_{xx}(i) + \sum_{j=1}^{m} a_j^f r_{xx}(j-i) = 0.$$

$\Rightarrow$

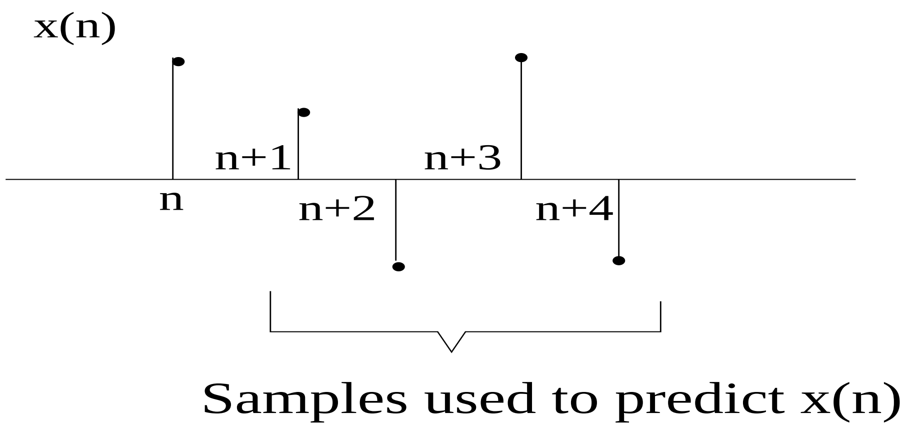
$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(m) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(m-1) \\ \vdots & & & \\ r_{xx}(m) & r_{xx}(m-1) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1^f \\ \vdots \\ a_m^f \end{bmatrix} = \begin{bmatrix} \delta^f \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Remarks: • This is exactly the YW - Equation.

• δ^f decreases as m increases.



Backward Linear prediction



$$\hat{x}^b(n) = -\sum_{i=1}^{m} a_i^b x(n+i).$$

$$e^b(n) = x(n-m) - \hat{x}^b(n-m)$$

$$\delta^b = E\left[\left(e^b(n)\right)^2\right].$$

To minimize δ^b, we obtain

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(m) \\ \vdots & & & \\ r_{xx}(m) & r_{xx}(m-1) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1^b \\ \vdots \\ a_m^b \end{bmatrix} = \begin{bmatrix} \delta^b \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

$$\Rightarrow \quad \boxed{\begin{array}{l} a_i^f = a_i^b, \quad \text{for all} \quad i \\ \delta^f = \delta^b. \end{array}}$$

Consider an AR(p) model and the notation in LDA:

Let $m = 1, 2, \cdots, p$

$$e_m^f(n) = x(n) + \sum_{i=1}^{m} a_{m,i}^f x(n-i)$$

$$= \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-m) \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{\theta}_m \end{bmatrix}.$$

$$e_m^b(n) = x(n-m) + \sum_{i=1}^{m} a_{m,i}^b x(n-m+i)$$

$$= [x(n-m) \quad x(n-m+1) \quad \cdots \quad x(n)] \begin{bmatrix} 1 \\ \boldsymbol{\theta}_m \end{bmatrix}$$

$$= [x(n) \quad \cdots \quad x(n-m+1) \quad x(n-m)] \begin{bmatrix} \tilde{\boldsymbol{\theta}}_m \\ 1 \end{bmatrix}$$

Recall LDA:

$$\boldsymbol{\theta}_m = \begin{bmatrix} \boldsymbol{\theta}_{m-1} \\ 0 \end{bmatrix} + k_m \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{m-1} \\ 1 \end{bmatrix}.$$

$e_m^f(n) =$

$$[x(n) \quad x(n-1) \quad \cdots \quad x(n-m)] \left\{ \begin{bmatrix} 1 \\ \boldsymbol{\theta}_{m-1} \\ 0 \end{bmatrix} + k_m \begin{bmatrix} 0 \\ \tilde{\boldsymbol{\theta}}_{m-1} \\ 1 \end{bmatrix} \right\}$$

$$= [x(n) \quad x(n-1) \quad \cdots \quad x(n-m+1)] \begin{bmatrix} 1 \\ \boldsymbol{\theta}_{m-1} \end{bmatrix}$$

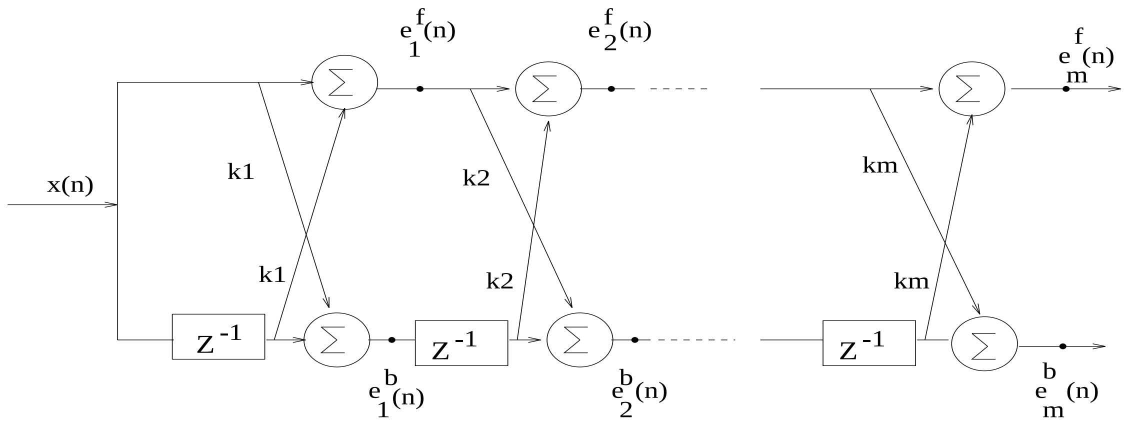
$$+ k_m [x(n-1) \quad x(n-2) \quad \cdots \quad x(n-m)] \begin{bmatrix} \tilde{\boldsymbol{\theta}}_{m-1} \\ 1 \end{bmatrix}$$

Similarly,

$$e_m^f(n) = e_{m-1}^f(n) + k_m e_{m-1}^b(n-1).$$

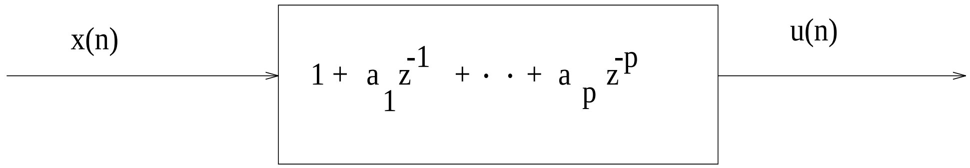
$$e_m^b(n) = e_{m-1}^b(n-1) + k_m e_{m-1}^f(n).$$

<u>Lattice Filter for Linear Prediction Error</u>



Remarks:• The implementation advantage of lattice filters is that they suffer from less round-off noise and are less sensitive to coefficient errors.

• If $x(n)$ is AR(p) and $m = p$, then



Whitening Filter

AR Spectral Estimation Methods

- **Autocorrelation or Yule-Walker method:** Recall that YW-Equation may be obtained by minimizing

$$E\left[e^2(n)\right] = E\left\{\left[x(n) - \hat{x}(n)\right]^2\right\},$$

where

$$\hat{x}(n) = -\sum_{k=1}^{p} a_k x(n-k).$$

The autocorrelation or YW method replaces $r(k)$ in the YW equation with biased $\hat{r}(k)$

$$\begin{bmatrix} \hat{r}(0) & \cdots & \hat{r}(p-1) \\ \vdots & \ddots & \vdots \\ \hat{r}(p-1) & \cdots & \hat{r}(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_p \end{bmatrix} = - \begin{bmatrix} \hat{r}(1) \\ \vdots \\ \hat{r}(p) \end{bmatrix}.$$

- **Covariance or Prony Method**

Consider the AR(p) signal,

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n), \quad n = 0, 1, \cdots, N-1$$

In matrix form,

$$\begin{bmatrix} x(p) \\ x(p+1) \\ \vdots \\ x(N-1) \end{bmatrix} =$$

$$- \begin{bmatrix} x(p-1) & x(p-2) & .. & x(0) \\ x(p) & x(p+1) & .. & x(1) \\ \vdots & & & \\ x(N-2) & & .. & x(N-p-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} + \begin{bmatrix} u(p) \\ u(p+1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

The Prony Method is to find LS solution to the overdetermined equation

$$-\begin{bmatrix} x(p-1) & \cdots & x(0) \\ \vdots & & \\ x(N-2) & \cdots & x(N-p-1) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} \approx \begin{bmatrix} x(p) \\ \vdots \\ x(N-1) \end{bmatrix}.$$

Remarks:

- The Covariance or Prony Method minimizes

$$\hat{\sigma}^2 = \frac{1}{N-p} \sum_{n=p}^{N-1} \hat{u}^2(n) = \frac{1}{N-p} \sum_{n=p}^{N-1} \left[x(n) + \sum_{k=1}^{p} \hat{a}_k x(n-k) \right]^2$$

- The Autocorrelation Method or YW-Method minimizes

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=-\infty}^{\infty} \left[x(n) + \sum_{k=1}^{p} \hat{a}_k x(n-k) \right]^2$$

where those $x(n)$ that are NOT available are set to zero.

- For large N, the YW and Prony methods yield similar results.

- For small N, YW method gives poor performance. The Prony method can give good estimates $\hat{a}_1, \cdots, \hat{a}_p$ for small N. The Prony method gives exact estimates for $x(n)$ =sum of sinusoids.

- Since biased $\hat{r}(k)$ are used in YW method, the estimated poles are inside unit circle. Prony method does not guarantee stability.

Modified Covariance or Forward Backward (F/B) Method

Recall Backward Linear Prediction:

$$x(n) = -\sum_{k=1}^{p} a_k^b x(n+k) + e^b(n).$$

For real data and real AR coefficients,

$$a_k^f = a_k^b = a_k, \quad k = 1, \cdots, p$$

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-p-1) \end{bmatrix} \approx \begin{bmatrix} x(1) & x(2) & \cdots & x(p) \\ x(2) & x(3) & \cdots & x(p+1) \\ \vdots & & & \\ x(N-p) & \cdots & & x(N-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix}$$

In the F/B method, this backward prediction equation is combined with the forward prediction equation and LS solution is found.

$$
-\begin{bmatrix} x(p-1) & \cdots & x(0) \\ \vdots & & \vdots \\ x(N-2) & \cdots & x(N-p-1) \\ x(1) & \cdots & x(p) \\ \vdots & & \vdots \\ x(N-p) & \cdots & x(N-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} x(p) \\ \vdots \\ x(N-1) \\ x(0) \\ \vdots \\ x(N-p-1) \end{bmatrix}
$$

Remarks: • The F/B method does not guarantee poles inside the unit circle. In Practice, the poles are usually inside the unit circle.

- For complex data and complex model,

$$a_k = a_k^f = \left(a_k^b\right)^*, \quad k = 1, \cdots, p$$

Then F/B solves:

$$-\begin{bmatrix} x(p-1) & \cdots & x(0) \\ \vdots & & \vdots \\ x(N-2) & \cdots & x(N-p-1) \\ x^*(1) & \cdots & x^*(p) \\ \vdots & & \vdots \\ x^*(N-p) & \cdots & x^*(N-1) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} x(p) \\ \vdots \\ x(N-1) \\ x^*(0) \\ \vdots \\ x^*(N-p-1) \end{bmatrix}$$

Remarks on $\hat{\sigma}^2$:

- In YW method,

$$\hat{\sigma}^2 = \hat{r}(0) + \sum_{k=1}^{p} \hat{a}_k \hat{r}(k).$$

- In Prony Method,

$$\text{Let} \quad \mathbf{e}_{LS} = \begin{bmatrix} e(p) \\ \vdots \\ e(N-1) \end{bmatrix}$$

$$\sigma^2 = \frac{1}{N-p} \sum_{n=p}^{N-1} |e(n)|^2$$

- In F/B Method,

$$\text{Let} \quad \mathbf{e}_{LS} = \begin{bmatrix} e^f(p) \\ \vdots \\ e^f(N-1) \\ e^b(0) \\ \vdots \\ e^b(N-p-1) \end{bmatrix}$$

$$\hat{\sigma}^2 = \frac{1}{2(N-p)} \left\{ \sum_{n=p}^{N-1} \left| e^f(n) \right|^2 + \sum_{n=0}^{N-p-1} \left| e^b(n) \right|^2 \right\}$$

Burg Method

Consider real data and real model. Recall LDA:

$$\boldsymbol{\theta}_{n+1} = \begin{bmatrix} \boldsymbol{\theta}_n \\ 0 \end{bmatrix} + k_{n+1} \begin{bmatrix} \tilde{\boldsymbol{\theta}}_n \\ 1 \end{bmatrix}$$

Thus, if we know $\boldsymbol{\theta}_n$ and k_{n+1}, we can find $\boldsymbol{\theta}_{n+1}$.

Recall

$$(\ddagger) \qquad \begin{cases} \hat{e}^f_m(n) = \hat{e}^f_{m-1}(n) + k_m \hat{e}^b_{m-1}(n-1) \\ \hat{e}^b_m(n) = \hat{e}^b_{m-1}(n-1) + k_m \hat{e}^f_{m-1}(n), \end{cases}$$

$$\text{where} \quad \hat{e}^f_{m-1}(n) = x(n) + \sum_{k=1}^{m-1} \hat{a}_{m-1,k} x(n-k).$$

$$\hat{e}^b_{m-1}(n) = x(n-m+1) + \sum_{k=1}^{m-1} \hat{a}_{m-1,k} x(n-m+1+k)$$

$\hat{k}_m$ is found by minimizing (for $\boldsymbol{\theta}_{m-1}$ given)

$$\frac{1}{2}\sum_{n=m}^{N-1}\left\{\left[\hat{e}_m^f(n)\right]^2+\left[\hat{e}_m^b(n)\right]^2\right\}.$$

$$\boxed{\hat{k}_m=\frac{-2\sum_{n=m}^{N-1}\hat{e}_{m-1}^f(n)\hat{e}_{m-1}^b(n-1)}{\sum_{n=m}^{N-1}\left\{\left[\hat{e}_{m-1}^f(n)\right]^2+\left[\hat{e}_{m-1}^b(n-1)\right]^2\right\}}.} \qquad (*)$$

Steps in Burg method:

Initialization $\left\{\right.$

- $\hat{r}(0)=\frac{1}{N}\sum_{n=0}^{N-1}x^2(n)$
- $\hat{\delta}_0=\hat{r}(0)$
- $\hat{e}_0^f(n)=x(n),\quad n=1,2,\cdots,N-1$
- $\hat{e}_0^b(n)=x(n),\quad n=0,1,\cdots,N-2.$

For $m = 1, 2, \cdots, p$,

- Calculate $\hat{k}_m$ with (*)
- $\hat{\delta}_m = \hat{\delta}_{m-1}(1 - \hat{k}_m^2)$
- $\hat{\boldsymbol{\theta}}_m = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{m-1} \\ 0 \end{bmatrix} + \hat{k}_m \begin{bmatrix} \tilde{\hat{\boldsymbol{\theta}}}_{m-1} \\ 1 \end{bmatrix}, \left(\quad \hat{\theta}_1 = \hat{k}_1 \right).$
- Update $\hat{e}_m^f(n)$ and $\hat{e}_m^b(n)$ with (‡)

Remarks: • $\hat{\delta}_p = \hat{\sigma}^2$.

- Since $a^2 + b^2 \geq 2ab, \quad \left|\hat{k}_m\right| < 1,$

$\Rightarrow$ Burg Method gives poles that are inside unit circle.

- Different ways of calculating $\hat{k}_m$ are available.

Properties of AR(p) Signals:

- Extension of $r(k)$:

* Given $r(0), r(1), \cdots, r(p)$.

* From YW - Equations we can calculate $a_1, a_2, \cdots, a_p, \sigma^2$

* $r(k) = -\sum_{l=1}^{p} a_l r(k-l), \quad k > p$

- Another point of view:

* Given $r(0), \cdots, r(p)$.

* Calculate $a_1, \cdots, a_p, \sigma^2$.

* Obtain $P(\omega)$

* $r(k) \overset{DTFT}{\longleftrightarrow} P(\omega)$.

Maximum Entropy Spectral Estimation

Given $r(0), \cdots, r(p)$. The remaining $r(p+1), \cdots$ are extrapolated to maximize entropy.

Entropy: Let Sample space for discrete random variable x be $x_1, \cdots, x_N$. The entropy $H(x)$ is

$$H(x) = -\sum_{i=1}^{N} P(x_i) \ln P(x_i),$$

$$P(x_i) = \text{prob}(x = x_i)$$

For continuous random variable,

$$H(x) = -\int_{-\infty}^{\infty} f(x) \ln f(x) dx.$$

$$f(x) = \text{pdf of} \quad x.$$

For Gaussian random variables,

$$\mathbf{x} = \begin{bmatrix} x(0) \\ \vdots \\ x(N-1) \end{bmatrix} \sim \quad N(0, \mathbf{R}_N)$$

$$H_N = \frac{1}{2} \ln(\det \mathbf{R}_N).$$

Since $H_N \to \infty$ as $N \to \infty$, we consider Entropy Rate:

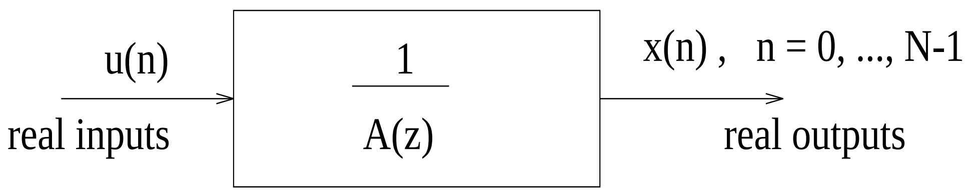
$$h = \lim_{N \to \infty} \frac{H_N}{N+1}$$

h is maximized with respect to $r(p+1), r(p+2), \cdots$.

Remark: For Gaussian case, we obtain Yule-Walker equations .... !

Maximum Likelihood Estimators:

- **Exact ML Estimator:**



$u(n)$ is Gaussian white noise with zero-mean.

$$\Rightarrow \begin{cases} E[u(n)] = 0, \\ Var[u(n)] = \sigma^2 \\ E[u(i)u(j)] = 0, i \neq j, \end{cases}$$

The likelihood function is

$$f = f\left[x(0), \cdots, x(N-1)|a_1, \cdots, a_p, \sigma^2\right]$$

The ML estimates of $a_1, \cdots, a_p, \sigma^2$ are found by maximizing f.

$$
\begin{aligned}
f &= f\left[x(p), \cdots, x(N-1) \,|\, x(0), \cdots, x(p-1), a_1, \cdots, a_p, \sigma^2\right] \\
&\quad f\left[x(0), \cdots, x(p-1) | a_1, \cdots, a_p, \sigma^2\right]
\end{aligned}
$$

* Consider first $f_1 = f\left[x(0), \cdots, x(p-1) | a_1, \cdots, a_p, \sigma^2\right]$

$$
f_1 = \frac{1}{(2\pi)^{\frac{p}{2}} \det^{\frac{1}{2}}(\mathbf{R}_p)} \exp\left[-\frac{1}{2}\left(\mathbf{x}_0^T \mathbf{R}_p^{-1} \mathbf{x}_0\right)\right].
$$

$$
\mathbf{x}_0 = \begin{bmatrix} x(0) \\ \vdots \\ x(p-1) \end{bmatrix}, \quad \mathbf{R}_p = \begin{bmatrix} r(0) & \cdots & r(p-1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{bmatrix}.
$$

Remark: $r(0), \cdots, r(p-1)$ are functions of $a_1, \cdots, a_p, \sigma^2$. (see, e.g., the YW system of equations)

* Consider next

$$f_2 = f\left[x(p), \cdots, x(N-1) | x(0), \cdots, x(p-1), a_1, \cdots, a_p, \sigma^2\right]$$

$$x(n) + \sum_{k=1}^{p} a_k x(n-k) = u(n)$$

$$\begin{cases} u(p) = x(p) + a_1 x(p-1) + \cdots + a_p x(0). \\ u(p+1) = x(p+1) + a_1 x(p) + \cdots + a_p x(1) \\ \vdots \\ u(N-1) = x(N-1) + a_1 x(N-2) + \cdots + a_p x(N-p-1). \end{cases}$$

$$
\begin{bmatrix} u(p) \\ \vdots \\ u(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & \cdots & a_p & \cdots & 1 \end{bmatrix} \begin{bmatrix} x(p) \\ x(p+1) \\ \vdots \\ x(N-1) \end{bmatrix}
$$

$$
+ \begin{bmatrix} a_1 x(p-1) + \cdots + a_p x(0) \\ a_2 x(p-1) + \cdots + a_p x(1) \\ \vdots \\ a_p x(p-1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

$$\text{Let} \quad \mathbf{u} = \begin{bmatrix} u(p) \\ \vdots \\ u(N-1) \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x(p) \\ \vdots \\ x(N-1) \end{bmatrix}$$

Given $x(0), \cdots, x(p-1), a_1, \cdots, a_p, \sigma^2$, x and u are related by linear transformation.

The Jacobian of the transformation

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ a_1 & 1 & & & \vdots \\ \vdots & & \ddots & & \\ 0 & \cdots & a_p & \cdots & 1 \end{bmatrix}$$

$$\det(\mathbf{J}) = 1$$

$$\begin{aligned} f(u) &= \frac{1}{(2\pi\sigma^2)^{\frac{N-p}{2}}} \exp\left[-\frac{1}{2\sigma^2}\mathbf{u}^T\mathbf{u}\right] \\ f_2 &= f[u(x)]\,|\det(\mathbf{J})| \\ &= f[u(x)]. \end{aligned}$$

Let $\mathbf{X} = \begin{bmatrix} x(p) & x(p-1) & \cdots & x(0) \\ x(p+1) & x(p) & \cdots & x(1) \\ \vdots & & & \\ x(N-1) & x(N-2) & \cdots & x(N-p-1) \end{bmatrix}$

$$\bar{\mathbf{a}} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}.$$

$$\mathbf{u} = \mathbf{X}\bar{\mathbf{a}}$$

$$\boxed{f_2 = \frac{1}{(2\pi\sigma^2)^{\frac{N-p}{2}}} \exp\left[-\frac{1}{2\sigma^2}\bar{\mathbf{a}}^T\mathbf{X}^T\mathbf{X}\bar{\mathbf{a}}\right].}$$

Remark: Maximizing $f = f_1.f_2$ with respect to $a_1, \cdots, a_p, \sigma^2$ is highly non-linear!

- **An Approximate ML Estimator**

$\hat{a}_1, \cdots, \hat{a}_p, \hat{\sigma}^2$ are found by maximizing f_2.

$\Rightarrow \hat{a}_1, \cdots, \hat{a}_p$ are found by minimizing $\bar{\mathbf{a}}^T \mathbf{X}^T \mathbf{X} \bar{\mathbf{a}} = \mathbf{u}^T \mathbf{u}$

$$\begin{bmatrix} x(p) & \cdots & x(0) \\ x(p+1) & \cdots & x(1) \\ \vdots & & \\ x(N-1) & \cdots & x(N-p-1) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} u(p) \\ u(p+1) \\ \vdots \\ u(N-1) \end{bmatrix}.$$

$\Rightarrow$ This is exactly Prony's Method !

$$\hat{\sigma}^2 = \frac{1}{N-p} \sum_{n=p}^{N-1} \left[x(n) + \sum_{j=1}^{p} \hat{a}_j x(n-j) \right]^2.$$

Again, exactly Prony's Method !

Accuracy of AR PSD Estimators

- Accuracy Analysis is <u>difficult</u>.
- Results for large N are available due to Central Limit Theorem.
- For large N, the variances for $\hat{a}_1, \cdots, \hat{a}_p, \quad \hat{k}_1, \cdots, \hat{k}_p, \quad \sigma^2, \hat{P}(\omega)$ are all proportional to $\frac{1}{N}$. Biases $\propto \frac{1}{N}$.

AR Model Order Selection

Remarks:

- Order too low yields smoothed/biased PSD estimate.
- Order too high yields spurious peaks/large variance in PSD estimate
- Almost all model order estimators are based on the estimate of the power of linear prediction error, denoted $\hat{\delta}_k$, where k is the model order chosen.

Final Prediction Error (FPE) Method

minimizes

$$\boxed{\mathrm{FPE}(k) = \frac{N+k}{N-k}\hat{\delta}_k \ .}$$

Akaike Information Criterion (AIC) Method

minimizes

$$\boxed{\mathrm{AIC}(k) = N \ln \hat{\delta}_k + 2k \ .}$$

Remarks:

- As $N \to \infty$, AIC's probability of error in choosing correct order does <u>NOT</u> $\to 0$.
- As $N \uparrow$, AIC tends to overestimate model order.

Minimum Description Length (MDL) Criterion

minimizes

$$\boxed{\mathrm{MDL}(k) = N \ln \hat{\delta}_k + k \ln N.}$$

Remark: As $N \to \infty$, MDL's probability of error $\to 0$. (consistent!).

Criterion Autoregressive Transfer (CAT) Method

minimizes

$$\mathrm{CAT}(k) = \frac{1}{N} \sum_{i=1}^{k} \frac{1}{\tilde{\delta}_i} - \frac{1}{\tilde{\delta}_k},$$

$$\tilde{\delta}_i = \frac{N}{N-i} \hat{\delta}_i$$

Remarks: • None of the above methods works well for small N

• Use these methods to initially estimate orders. (Practical experience needed).

Noisy AR Processes:

$$\underline{y(n) = x(n) + w(n)}$$

- $x(n) = \mathrm{AR}(p)$ process.
- $w(n) =$ White Gaussian noise with zero-mean and variance σ_w^2
- $x(n)$ and $w(n)$ are Independent of each other.

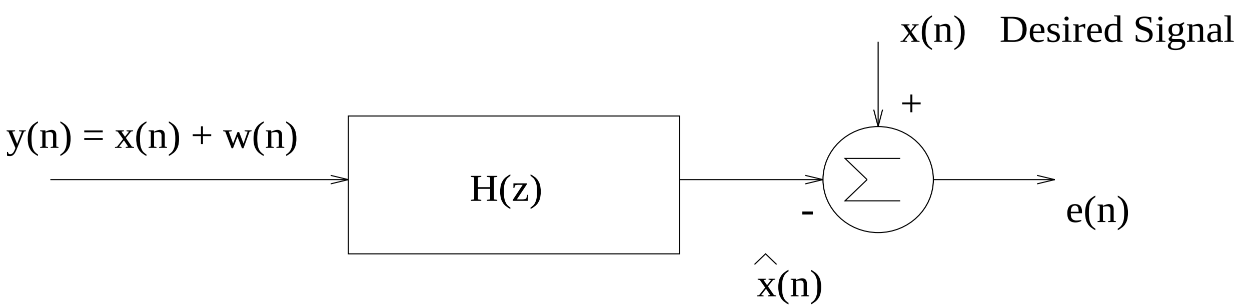
$$\begin{aligned} P_{yy}(\omega) &= P_{xx}(\omega) + P_{ww}(\omega) \\ &= \frac{\sigma^2}{|A(\omega)|^2} + \sigma_w^2 \\ &= \frac{\sigma^2 + \sigma_w^2 |A(\omega)|^2}{|A(\omega)|^2}. \end{aligned}$$

Remarks: • $y(n)$ is an ARMA signal

• $a_1, \cdots, a_p, \quad \sigma^2, \sigma_w^2$ may be estimated by

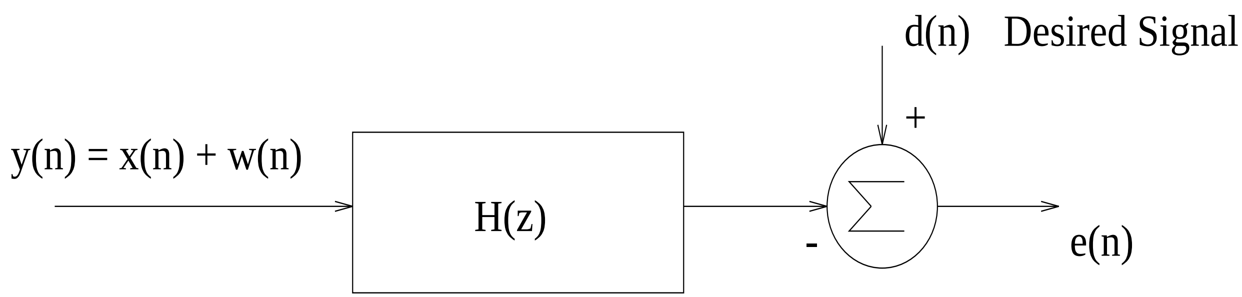
  * ARMA methods.
  * A large order AR approximation.
  * Compensating the effect of $w(n)$.
  * Bootstrap or adaptive filtering and AR methods.

Wiener Filter: (Wiener-Hopf Filter)



- $H(z)$ is found by minimizing $E\left[|e(n)|^2\right]$.
- $H(z)$ depends on knowing $P_{xy}(\omega)$.

General Filtering Problem: (Complex Signals)



Special case of $d(n)$: $d(n) = x(n+m)$:

1.) $m > 0$, m - step ahead prediction.

2.) $m = 0$, filtering problem

3.) $m < 0$, smoothing problem.

Three common filters:

1.) General Non-causal:

$$H(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k}.$$

2.) General Causal:

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k}$$

3.) Finite Impulse Response (FIR):

$$H(z) = \sum_{k=0}^{p} h_k z^{-k}$$

Case 1: Non-causal Filter.

$$E = E\left\{|e(n)|^2\right\}$$

$$= E\left\{\left[d(n) - \sum_{k=-\infty}^{\infty} h_k y(n-k)\right]\left[d(n) - \sum_{l=-\infty}^{\infty} h_l y(n-l)\right]^*\right\}$$

$$= r_{dd}(0) - \sum_{l=-\infty}^{\infty} h_l^* r_{dy}(l) - \sum_{k=-\infty}^{\infty} h_k r_{dy}^*(k)$$

$$+ \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} r_{yy}(l-k) h_k h_l^*$$

Remark: For Causal and FIR filters, only limits of sums differ.

$$\text{Let} \quad h_i = \alpha_i + j\beta_i \qquad \frac{\partial E}{\partial \alpha_i} = 0, \quad \frac{\partial E}{\partial \beta_i} = 0.$$

$$\Rightarrow r_{dy}(i) = \sum_{k=-\infty}^{\infty} h_k^o r_{yy}(i-k), \quad \forall i$$

In Z - domain

$$\boxed{P_{dy}(z) = H^o(z)P_{yy}(z)}$$

which is the optimum Non-causal Wiener Filter.

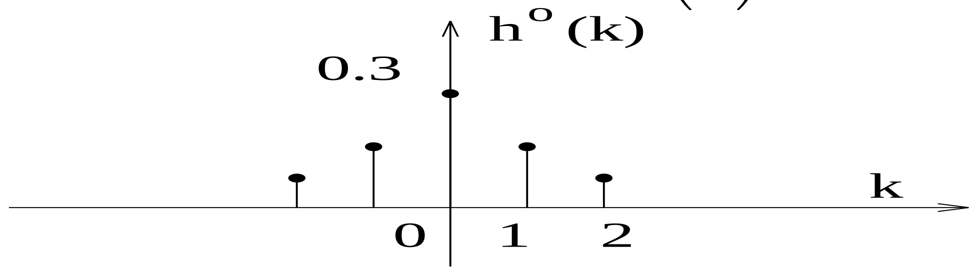
$$\begin{aligned} Ex: \quad d(n) &= x(n), \quad y(n) = x(n) + w(n), \\ P_{xx}(z) &= \frac{0.36}{(1-0.8z^{-1})(1-0.8z)} \\ P_{ww}(z) &= 1. \end{aligned}$$

$x(n)$ and $w(n)$ are uncorrelated.

Optimal filter ?

$$\begin{aligned} P_{yy}(z) &= P_{xx}(z) + P_{ww}(z) \\ &= \frac{0.36}{(1-0.8z^{-1})(1-0.8z)} + 1 \\ &= 1.6\frac{(1-0.5z^{-1})(1-0.5z)}{(1-0.8z^{-1})(1-0.8z)} \end{aligned}$$

$$
\begin{aligned}
r_{dy}(k) &= E\left[d(n+k)y^*(n)\right] \\
&= E\left\{x(n+k)\left[x^*(n)+w(n)\right]\right\} \\
&= r_{xx}(k). \\
P_{dy}(z) &= P_{xx}(z) \\
H^o(z) &= \frac{P_{yy}(z)}{P_{dy}(z)} \\
&= \frac{0.36}{1.6\left(1-0.5z^{-1}\right)\left(1-0.5z\right)} \\
h^o(k) &= 0.3\left(\frac{1}{2}\right)^{|k|}
\end{aligned}
$$



Case 2: Causal Filter.

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k}$$

Through similar derivations as for Case 1, we have

$$\boxed{r_{dy}(i) = \textstyle\sum_{k=0}^{\infty} h_k^o r_{yy}(i-k),}$$

$$h_k^o = ?$$

Let

$$B(z)B^*\left(\frac{1}{z^*}\right) = \frac{1}{P_{yy}(z)}$$

Pick $B(z)$ to be a stable, causal, minimum phase system.

Then

$$P_{dy}(z) = \underbrace{H^o(z)B^{-1}(z)}_{\text{causal}} B^{-*}\left(\frac{1}{z^*}\right)$$

$$\Rightarrow \quad \frac{H^o(z)}{B(z)} = \left[P_{dy}(z)B^*\left(\frac{1}{z^*}\right)\right]_+ ,$$

where

$$[X(z)]_+ = \left[\sum_{k=-\infty}^{\infty} x_k z^{-k}\right]_+ = \sum_{k=0}^{\infty} x_k z^{-k}.$$

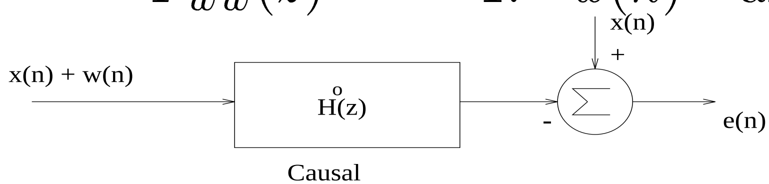
$\Rightarrow$

$$\boxed{H^o(z) = B(z)\left[P_{dy}(z)B^*\left(\tfrac{1}{z^*}\right)\right]_+}$$

Ex. (Same as previous one)

$$P_{xx}(z) = \frac{0.36}{(1-0.8z^{-1})(1-0.8z)},$$

$$P_{ww}(z) = 1. \quad x(n) \quad \text{and} \quad w(n) \quad \text{independent}$$



$$\begin{aligned}
P_{dy}(z) &= P_{xy}(z) = P_{xx}(z) \\
P_{yy}(z) &= \frac{1.6\left(1-0.5z^{-1}\right)(1-0.5z)}{(1-0.8z^{-1})(1-0.8z)}. \\
B(z) &= \frac{1}{\sqrt{1.6}}\frac{1-0.8z^{-1}}{1-0.5z^{-1}} \quad (\text{ stable and causal }) \\
P_{dy}(z)B^*\left(\frac{1}{z^*}\right) &= \frac{0.36}{(1-0.8z^{-1})(1-0.8z)}\frac{1}{\sqrt{1.6}}\frac{1-0.8z}{1-0.5z} \\
&= \frac{0.36}{\sqrt{1.6}}\frac{1}{(1-0.8z^{-1})(1-0.5z)}.
\end{aligned}$$

$$
\begin{aligned}
P_{dy}(z)B^*\left(\frac{1}{z^*}\right) &= \frac{0.36}{\sqrt{1.6}}\left(\frac{\frac{5}{3}}{1-0.8z^{-1}} + \frac{\frac{5}{6}z}{1-0.5z}\right) \\
\left[P_{dy}(z)B^*\left(\frac{1}{z^*}\right)\right]_+ &= \frac{0.36}{\sqrt{1.6}}\frac{\frac{5}{3}}{1-0.8z^{-1}} = G^o(z) \\
H^o(z) &= \frac{0.36}{\sqrt{1.6}}\frac{\frac{5}{3}}{1-0.8z^{-1}}\frac{1}{\sqrt{1.6}}\frac{1-0.8z^{-1}}{1-0.5z^{-1}} \\
&= 0.375\frac{1}{1-0.5z^{-1}} \\
\Rightarrow h^o(k) &= \frac{3}{8}\left(\frac{1}{2}\right)^k, \quad k = 0, 1, 2, \cdots
\end{aligned}
$$

Case 3: FIR Filter:

$$H(z) = \sum_{k=0}^{p} h_k z^{-k}$$

Again, we can show similarly

$$r_{dy}(i) = \sum_{k=0}^{p} h_k^o r_{yy}(i-k).$$

$$\begin{bmatrix} r_{dy}(0) \\ r_{dy}(1) \\ \vdots \\ r_{dy}(p) \end{bmatrix} = \begin{bmatrix} r_{yy}(0) & r_{yy}(-1) & \cdots & r_{yy}(-p) \\ r_{yy}(1) & r_{yy}(0) & \cdots & \\ \vdots & \ddots & & \\ r_{yy}(p) & r_{yy}(p-1) & \cdots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} h_0^o \\ h_1^o \\ \vdots \\ h_p^o \end{bmatrix}$$

Remark: The Minimum error E is the smallest in case (1) and largest in case (3).

Parametric Methods for Line Spectra

$$
\begin{aligned}
y(n) &= x(n) + w(n) \\
x(n) &= \sum_{k=1}^{K} \alpha_k e^{j(\omega_k n + \phi_k)} \\
\phi_k &= \text{Initial phases, independent of each other,} \\
&\quad \text{uniform distribution on } [-\pi, \pi] \\
\alpha_k &= \text{amplitudes, constants, } > 0 \\
\omega_k &= \text{angular frequencies} \\
w(n) &= \text{zero-mean white Gaussian Noise,} \\
&\quad \text{independent of } \phi_1, \cdots, \phi_K
\end{aligned}
$$

Remarks:

- Applications: Radar, Communications, $\cdots$.
- We are <u>mostly interested</u> in estimating $\omega_1, \cdots, \omega_K$.
- Once $\omega_1, \cdots, \omega_K$ are estimated, $\hat{\alpha}_1, \cdots, \hat{\alpha}_K, \quad \hat{\phi}_1, \cdots, \hat{\phi}_K$ can be found readily from $\hat{\omega}_1, \cdots, \hat{\omega}_K$

Let $\quad \alpha_k e^{j\phi_k} = \beta_k$

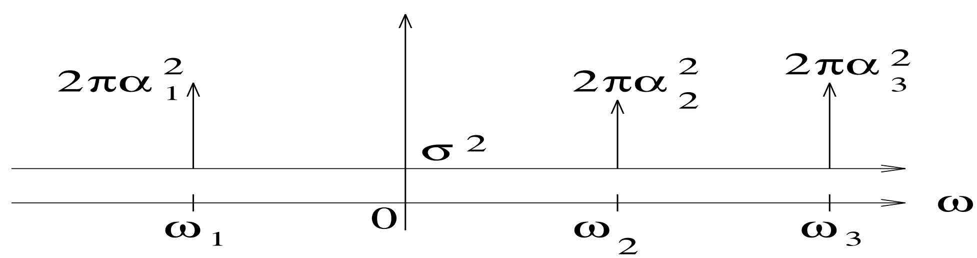
$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} \approx \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j\hat{\omega}_1} & e^{j\hat{\omega}_2} & \cdots & e^{j\hat{\omega}_K} \\ \vdots & & & \vdots \\ e^{j(N-1)\hat{\omega}_1} & \cdots & & e^{j(N-1)\hat{\omega}_K} \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix}$$

The amplitude of $\hat{\beta}_k$ is α_k. The phase of $\hat{\beta}_k$ is ϕ_k.

Remarks:

- $$\begin{aligned} r_{yy}(k) &= E\{y^*(n)y(n+k)\} \\ &= \sum_{i=1}^{K} \alpha_i^2 e^{j\omega_i k} + \sigma^2 \delta(k) \end{aligned}$$

- $$P_{yy}(\omega) = 2\pi \sum_{i=1}^{K} \alpha_i^2 \delta(\omega - \omega_i) + \sigma^2.$$



- Recall that the resolution limit of Periodogram is $\frac{1}{N}$
- The Parametric methods below have resolution better than $\frac{1}{N}$.

(These methods are the so-called High - Resolution or Super - Resolution methods)

Maximum Likelihood Estimator

$w(n)$ is assumed to be zero-mean circularly symmetric complex Gaussian random variable with variance σ^2.

The pdf of $w(n)$ is $N(0, \sigma^2)$

$$f\left(w(n)\right) = \frac{1}{\pi\sigma^2} \exp\left\{ -\frac{\left|w(n)\right|^2}{\sigma^2} \right\}.$$

Remark: • The real and imaginary parts of $w(n)$ are real Gaussian random variables with zero-mean and variance $\frac{\sigma^2}{2}$.

• The two parts are independent of each other.

$$f\left(w(0),\cdots,w(N-1)\right)=\frac{1}{\left(\pi\sigma^2\right)^N}\exp\left\{-\frac{\sum_{n=0}^{N-1}\left|w(n)\right|^2}{\sigma^2}\right\}$$

The likelihood function of $y(0), \quad \cdots, \quad y(N-1)$ is

$$f=f\left(y(0),\cdots,y(N-1)\right)=\frac{1}{\left(\pi\sigma^2\right)^N}\exp\left\{-\frac{\sum_{n=0}^{N-1}\left|y(n)-x(n)\right|^2}{\sigma^2}\right\}$$

Remark: The ML estimates of

$\omega_1,\cdots,\omega_K, \quad \alpha_1,\cdots,\alpha_K, \quad \phi_1,\cdots,\phi_K$ are found by maximizing f with respect to $\omega_1,\cdots,\omega_K,\alpha_1,\cdots,\alpha_K,\phi_1,\cdots,\phi_K$.

Equivalently, we minimize

$$g=\sum_{n=0}^{N-1}\left|y(n)-\sum_{k=1}^{K}\alpha_k e^{j(\omega_k n+\phi_k)}\right|^2$$

Remarks: If $w(n)$ is neither Gaussian nor white, minimizing g is called the <u>non-linear least-squares</u> method, in general.

- Let $\mathbf{y} = \begin{bmatrix} y(0) \\ \vdots \\ y(N-1) \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_K \end{bmatrix}, \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_K \end{bmatrix}$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j\omega_1} & e^{j\omega_2} & \cdots & e^{j\omega_K} \\ \vdots & & & \vdots \\ e^{j(N-1)\omega_1} & \ldots & & e^{j(N-1)\omega_K} \end{bmatrix}$$

$$
\begin{aligned}
g &= (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^H (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) . \\
&= \left[\boldsymbol{\beta} - \left(\mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{B}^H\mathbf{y}\right]^H \left(\mathbf{B}^H\mathbf{B}\right)\left[\boldsymbol{\beta} - \left(\mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{B}^H\mathbf{y}\right] \\
&+ \mathbf{y}^H\mathbf{y} - \mathbf{y}^H\mathbf{B}\left(\mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{B}^H\mathbf{y}.
\end{aligned}
$$

$$
\Rightarrow \quad \boxed{\begin{aligned}
\hat{\boldsymbol{\omega}} &= \operatorname{argmax}_{\boldsymbol{\omega}} \left[\mathbf{y}^H\mathbf{B}\left(\mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{B}^H\mathbf{y}\right] . \\
\hat{\boldsymbol{\beta}} &= \left(\mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{B}^H\mathbf{y}\Big|_{\boldsymbol{\omega}=\hat{\boldsymbol{\omega}}} .
\end{aligned}}
$$

Remarks: • $\hat{\boldsymbol{\omega}}$ is a consistent estimate of $\boldsymbol{\omega}$

- For large N,

$$
\begin{aligned}
E\left[(\hat{\boldsymbol{\omega}}-\boldsymbol{\omega})(\hat{\boldsymbol{\omega}}-\boldsymbol{\omega})^H\right] &= \frac{6\sigma^2}{N^3}\begin{bmatrix} \frac{1}{\alpha_1^2} & & \\ & \ddots & \\ & & \frac{1}{\alpha_K^2} \end{bmatrix} \\
&= \text{CRB}
\end{aligned}
$$

However,

- The maximization to obtain $\hat{\boldsymbol{\omega}}$ is difficult to implement.

* The search may not find global maximum.

* Computationally expensive.

Special Cases:

1.) $K = 1$

$$\hat{\omega} = \text{argmax}_{\omega} \underbrace{\left[\mathbf{y}^H \mathbf{B} \left(\mathbf{B}^H \mathbf{B} \right)^{-1} \mathbf{B}^H \mathbf{y} \right]}_{\mathbf{g_1}},$$

$$\mathbf{B} = \begin{bmatrix} 1 \\ e^{j\omega} \\ \vdots \\ e^{j(N-1)\omega} \end{bmatrix}, \mathbf{B}^H \mathbf{B} = N.$$

$$\mathbf{B}^H\mathbf{y} = \begin{bmatrix} 1 & e^{-j\omega} & \cdots & e^{-j(N-1)\omega} \end{bmatrix} \begin{bmatrix} y(0) \\ \vdots \\ y(N-1) \end{bmatrix}$$

$$= \sum_{n-0}^{N-1} y(n)e^{-j\omega n}$$

$$\Rightarrow \quad \hat{\omega} = \text{argmax}_{\omega} \frac{1}{N} \left| \sum_{n=0}^{N-1} y(n)e^{-j\omega n} \right|^2$$

$\hat{\omega}$ corresponds to the highest peak of the Periodogram !

2.)

$$\triangle\omega = inf_{i\neq k}\left|\omega_i - \omega_k\right| > \frac{2\pi}{N}.$$

$$\text{Since}\quad Var\left(\hat{\omega}_k - \omega_k\right) \propto \frac{1}{N^3}$$

$$\Rightarrow \quad \hat{\omega}_k - \omega_k \propto \frac{1}{N^{\frac{3}{2}}}.$$

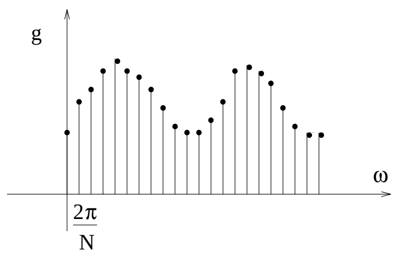
$$\Rightarrow \quad inf_{i\neq k}\left|\hat{\omega}_i - \hat{\omega}_k\right| > \frac{2\pi}{N}.$$

$\Rightarrow$ We can resolve all K sine waves by evaluating g_1 at FFT points:

$$\tilde{\omega}_i = \frac{2\pi}{N} i, \quad i = 0, \cdots, N-1$$

Any K of these $\tilde{\omega}_i$ gives $\mathbf{B}^H\mathbf{B} = N\mathbf{I}$, $\mathbf{I} =$ Identity matrix.

$$\Rightarrow \quad g_1 = \sum_{k=1}^{K} \frac{1}{N}\left|\sum_{n=0}^{N-1} y(n)e^{-j\tilde{\omega}_k n}\right|^2.$$



⇒ The K $\tilde{\omega}_i$ that maximizes g_1 correspond to the largest K peaks of the Periodogram.

Remarks: • $\hat{\omega}_k$ estimates obtained by using the K largest peaks of Periodogram have accuracy $\underline{\hat{\omega}_k - \omega_k \propto \frac{2\pi}{N}}$

• The periodogram is a good frequency estimator. (This was introduced by Schuster a century ago !)

High - Resolution Methods

- Statistical Performance Close to ML estimator (or CRB) .
- Avoid Multidimensional search over parameter space.
- Do not depend on Resolution condition.
- All provide consistent estimates
- All give similar performance, especially for large N.
- Method of choice is a " Matter - of - Taste ".

Higher - Order Yule- Walker (HOYW) Method:

$$\text{Let} \quad x_k(n) = \alpha_k e^{j(\omega_k n + \phi_k)}$$

$$\begin{aligned} \left[1 - e^{j\omega_k} z^{-1}\right] x_k(n) &= x_k(n) - e^{j\omega_k} x_k(n-1) \\ &= \alpha_k e^{j(\omega_k n + \phi_k)} - e^{j\omega_k} \alpha_k e^{j[\omega_k (n-1) + \phi_k]} \\ &= 0 \end{aligned}$$

$\Rightarrow \left[1 - e^{j\omega_k} z^{-1}\right]$ is an Annihilating filter for $x_k(n)$.

$$\text{Let} \quad A(z) = \Pi_{k=1}^{K} \left(1 - e^{j\omega_k} z^{-1}\right)$$

$\Rightarrow$

$$\boxed{A(z)x(n) = 0}$$

$$y(n) = x(n) + w(n)$$

$$\Rightarrow \qquad A(z)y(n) = A(z)w(n) \qquad (*)$$

Remark:

• It is tempting to cancel $A(z)$ from both sides above, but this is wrong since $y(n) \neq w(n)$!

Multiplying both sides of $(*)$ by a polynomial $\bar{A}(z)$ of order $L-K$ gives

$$\left(1+\tilde{a}_1 z^{-1}+\cdots+\tilde{a}_L z^{-L}\right) y(n)=\left(1+\tilde{a}_1 z^{-1}+\cdots+\tilde{a}_L z^{-L}\right) w(n)$$

$$\text{where} \quad 1+\tilde{a}_1 z^{-1}+\cdots+\tilde{a}_L z^{-L}=A(z)\bar{A}(z)$$

$$\Rightarrow [y(n) \quad y(n-1)\cdots \quad y(n-L)]\begin{bmatrix} 1 \\ \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix}=w(n)+\cdots+a_L w(n-L)$$

Multiplying both sides by $\begin{bmatrix} y^*(n-L-1) \\ \vdots \\ y^*(n-L-M) \end{bmatrix}$,

we get
$$\begin{bmatrix} r_{yy}(L+1) & \cdots & r_{yy}(1) \\ \vdots & & \\ r_{yy}(L+M) & \cdots & r_{yy}(M) \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} = 0.$$

$$\Rightarrow \begin{bmatrix} r_{yy}(L) & \cdots & r_{yy}(1) \\ \vdots & & \\ r_{yy}(L+M-1) & \cdots & r_{yy}(M) \end{bmatrix} \begin{bmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} = - \begin{bmatrix} r_{yy}(L+1) \\ \vdots \\ r_{yy}(L+M) \end{bmatrix}$$

$$\Rightarrow \boldsymbol{\Gamma}\tilde{\mathbf{a}} = -\boldsymbol{\gamma}$$

Remarks:

- When $y(0), \cdots, y(N-1)$ are the only data available, we first estimate $r_{yy}(i)$ and replace $r_{yy}(i)$ in above equation with estimate $\hat{r}_{yy}(i)$
- $\{\hat{\omega}_K\}$ are the angular positions of the K roots nearest the unit circle
- Increasing L and M will

  * give <u>better</u> performance due to using the information in higher lags of $\hat{r}(i)$
- Increasing L and M 'too much' will

  * give <u>worse</u> performance due to increased variance in $\hat{r}(i)$ for large i

Γ has rank K, if $M \geq K$ and $L \geq K$

Proof: Let $\tilde{\mathbf{y}}_i(n) = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-i+1) \end{bmatrix}, \quad \tilde{\mathbf{w}}_i(n) = \begin{bmatrix} w(n) \\ w(n-1) \\ \vdots \\ w(n-i+1) \end{bmatrix}$

$$\tilde{\mathbf{x}}(n) = \begin{bmatrix} x_1(n) \\ \vdots \\ x_K(n) \end{bmatrix}, \quad \mathbf{x}_k(n) = \alpha_k e^{j(\omega_k n + \phi_k)}$$

$$\tilde{\mathbf{y}}_i(n) = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\omega_1} & e^{-j\omega_2} & \cdots & e^{-j\omega_K} \\ \vdots & & & \\ e^{-j(i-1)\omega_1} & \cdots & & e^{-j(i-1)\omega_K} \end{bmatrix}}_{\mathbf{A}_i} \tilde{\mathbf{x}}(n) + \tilde{\mathbf{w}}_i(n)$$

$\mathbf{A}_i = i \times K$ Vandermonde matrix.

$\text{rank}(\mathbf{A}_i) = K$ if $i \geq K$ and $\omega_k \neq \omega_l$ for $k \neq l$.

$\Rightarrow \tilde{\mathbf{y}}_i(n) = \mathbf{A}_i \tilde{\mathbf{x}}(n) + \tilde{\mathbf{w}}_i(n)$

Thus $\boldsymbol{\Gamma}^* = E\left\{\begin{bmatrix} y(n-L-1) \\ \vdots \\ y(n-L-M) \end{bmatrix}\begin{bmatrix} y^*(n-1) & \cdots & y^*(n-L) \end{bmatrix}\right\}$

$$\begin{aligned} &= E\left\{\mathbf{A}_M \tilde{\mathbf{x}}(n-L-1)\tilde{\mathbf{x}}^H(n-1)\mathbf{A}_L^H\right\} \\ &\triangleq \mathbf{A}_M \mathbf{P}_{L+1} \mathbf{A}_L^H, \end{aligned}$$

where $\mathbf{P}_{L+1} = E\left\{\tilde{\mathbf{x}}(n-L)\tilde{\mathbf{x}}^H(n)\right\}$

- $$\begin{aligned} E\{x_i(n)\} &= E\left\{\alpha_i e^{j(\omega_i n+\phi_i)}\right\} \\ &= \int_{-\pi}^{\pi} \alpha_i e^{j\omega_i n} e^{j\phi_i} \frac{1}{2\pi} d\phi_i = 0 \end{aligned}$$

- $$\begin{aligned} E\{x_i(n-k)x_i^*(n)\} & \\ &= E\left\{\alpha_i e^{j[\omega_i(n-k)+\phi_i]} \alpha_i e^{-j(\omega_i n+\phi_i)}\right\} \\ &= \alpha_i^2 e^{-j\omega_i k} \end{aligned}$$

- Since $\phi_i's$ are independent of each other,

$$E\left\{x_i(n-k)x_j^*(n)\right\} = 0, \quad i \neq j$$

$$\mathbf{P}_{L+1} = E\left\{\begin{bmatrix} x_1(n-L-1) \\ x_2(n-L-1) \\ \vdots \\ x_K(n-L-1) \end{bmatrix}\begin{bmatrix} x_1^*(n-1) & \cdots & x_K^*(n-1) \end{bmatrix}\right\}$$

$$= \begin{bmatrix} \alpha_1^2 e^{-j\omega_1 L} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_K^2 e^{-j\omega_K L} \end{bmatrix}$$

Remark: For $M \geq K$ and $L \geq K$, $\mathbf{\Gamma}^*$ is of rank K, so is $\mathbf{\Gamma}$.

Consider

$$\begin{bmatrix} \hat{r}_{yy}(L) & \cdots & \hat{r}_{yy}(1) \\ \vdots & & \\ \hat{r}_{yy}(L+M-1) & \cdots & \hat{r}_{yy}(M) \end{bmatrix} \begin{bmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} \approx - \begin{bmatrix} \hat{r}_{yy}(L+1) \\ \vdots \\ \hat{r}_{yy}(L+M) \end{bmatrix}$$

$\Rightarrow \quad \hat{\mathbf{\Gamma}}\tilde{\mathbf{a}} \approx -\hat{\boldsymbol{\gamma}}.$

Remarks: rank $(\hat{\mathbf{\Gamma}}) = \quad \min(M, L)$

almost surely, due to errors in $\hat{r}_{yy}(i)$

• For large N, $\hat{r}_{yy}(i) \approx r_{yy}(i)$ makes $\hat{\mathbf{\Gamma}}$ ill conditioned.

• For large N, LS estimates of $\tilde{a}_1, \cdots, \tilde{a}_L$ give poor estimates of $\omega_1, \cdots, \omega_K$.

Let us use this rank information as follows: Let

$$\begin{aligned} \hat{\boldsymbol{\Gamma}} &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H \\ &= [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \boldsymbol{\Sigma}_1 & 0 \\ 0 & \boldsymbol{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \quad \begin{matrix} K \\ L-K \end{matrix} \end{aligned}$$

denote the singular value decomposition (SVD) of $\hat{\boldsymbol{\Gamma}}$. (Diagonal elements in $\begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 \end{bmatrix}$ arranged from large to small).

Since $\hat{\boldsymbol{\Gamma}}$ is close to rank K, and $\boldsymbol{\Gamma}$ has rank K,

$$\hat{\boldsymbol{\Gamma}}_K = \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^H$$

(The <u>best Rank - K Approximation</u> of $\hat{\Gamma}$ in the Frobenius Norm sense) is generally a better estimate of $\boldsymbol{\Gamma}$ than $\hat{\boldsymbol{\Gamma}}$.

$$\hat{\boldsymbol{\Gamma}}_K \tilde{\mathbf{a}} \approx -\hat{\gamma}, \qquad \hat{\tilde{\mathbf{a}}} = -\mathbf{V}_1\boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^H\hat{\gamma} \qquad (**)$$

Remark:

- Using $\hat{\mathbf{\Gamma}}_K$ to replace $\mathbf{\Gamma}$ gives <u>better</u> frequency estimation.
- This result may be explained by the fact that $\hat{\mathbf{\Gamma}}_K$ is <u>closer</u> to $\mathbf{\Gamma}$ than $\hat{\mathbf{\Gamma}}$.
- The rank approximation step is referred as " noise cleaning ".

Summary of HOYW Frequency Estimator

Step 1: Compute $\hat{r}(k), k = 1, 2, \cdots, L + M$.

Step 2: Compute the SVD of $\hat{\mathbf{\Gamma}}$ and determine $\hat{\tilde{\mathbf{a}}}$ with (**)

Step 3: Compute the roots of

$$1 + \hat{\tilde{a}}_1 z^{-1} + \cdots + \hat{\tilde{a}}_L z^{-L} = 0$$

Pick the K roots that are nearest the unit circle and obtain the frequency estimates as the angular positions (phases) of these roots.

Remarks: • Rule of Thumb for selecting L and M:

$$L \approx M$$

$$L + M \approx \frac{N}{3}$$

• Although one cannot guarantee that the K roots nearest the unit circle give the best frequency estimates, empirical evidence shows that this is true most often .

Some Math Background

Lemma: Let $\mathbf{U}$ be a unitary matrix; i.e., $\mathbf{U}^H\mathbf{U} = \mathbf{I}$.

$$\text{Then} \quad ||\mathbf{U}\mathbf{b}||_2^2 = ||\mathbf{b}||_2^2,$$

$$\text{where} \quad ||\mathbf{x}||_2^2 = \mathbf{x}^H\mathbf{x}.$$

Proof:

$$||\mathbf{U}\mathbf{b}||_2^2 = \mathbf{b}^H\mathbf{U}^H\mathbf{U}\mathbf{b} = \mathbf{b}^H\mathbf{b} = ||\mathbf{b}||^2.$$

$$\text{Consider} \quad \mathbf{A}\mathbf{x} \approx \mathbf{b},$$

where
$\mathbf{A}$ is $M \times L$,
$\mathbf{x}$ is $L \times 1$,
$\mathbf{b}$ is $M \times 1$,
$\mathbf{A}$ is of rank K

SVD of A:

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix}$$

Goal: Find the minimum-norm $\mathbf{x}$ so that $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 = \text{minimum}$.

$$\begin{aligned}
||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 &= ||\mathbf{U}^H\mathbf{A}\mathbf{x} - \mathbf{U}^H\mathbf{b}||_2^2 \\
&= ||\mathbf{U}^H\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H\mathbf{x} - \mathbf{U}^H\mathbf{b}||^2 \\
&= ||\boldsymbol{\Sigma}\underbrace{\mathbf{V}^H\mathbf{x}}_{\mathbf{y}} - \mathbf{U}^H\mathbf{b}||_2^2 \\
&= ||\boldsymbol{\Sigma}\mathbf{y} - \mathbf{U}^H\mathbf{b}||_2^2 \\
&= \left\| \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{U}_1^H\mathbf{b} \\ \mathbf{U}_2^H\mathbf{b} \end{bmatrix} \right\|_2^2 \\
&= ||\boldsymbol{\Sigma}_1\mathbf{y}_1 - \mathbf{U}_1^H\mathbf{b}||_2^2 + ||\mathbf{U}_2^H\mathbf{b}||_2^2
\end{aligned}$$

To minimize $||\mathbf{Ax}-\mathbf{b}||_2^2$, we must have,

$$\mathbf{\Sigma}_1\mathbf{y}_1 = \mathbf{U}_1^H\mathbf{b}$$

$$\Rightarrow \quad \boxed{\mathbf{y}_1 = \mathbf{\Sigma}_1^{-1}\mathbf{U}_1^H\mathbf{b}\ .}$$

Note that $\mathbf{y}_2$ can be anything and $||\mathbf{Ax}-\mathbf{b}||_2^2$ is not affected.

Let $\mathbf{y}_2 = 0$ so that $||\mathbf{y}||_2^2 = ||\mathbf{x}||_2^2 = \text{minimum}$.

$$\Rightarrow \mathbf{V}^H\mathbf{x} = \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ 0 \end{bmatrix}$$

$$\Rightarrow \mathbf{x} = \mathbf{V}\mathbf{y} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix}\begin{bmatrix} \mathbf{y}_1 \\ 0 \end{bmatrix} = \mathbf{V}_1\mathbf{y}_1$$

$$\Rightarrow \quad \boxed{\mathbf{x} = \mathbf{V}_1\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^H\mathbf{b}.}$$

$$||\mathbf{x}||_2^2 = ||\mathbf{y}||_2^2 = \quad \text{minimum}$$

SVD Prony Method

Recall:
$$\left(1+\tilde{a}_1 z^{-1}+\cdots+\tilde{a}_L z^{-L}\right) y(n) = (1+\tilde{a}_1 z^{-1}+\cdots+\tilde{a}_L z^{-L}) w(n). \quad (L \geq K)$$

At not too low SNR,

$$\begin{bmatrix} y(n) & y(n-1) & \cdots & y(n-L) \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} \approx 0$$

$$\begin{bmatrix} y(L) & y(L-1) & \cdots & y(0) \\ y(L+1) & y(L) & \cdots & y(1) \\ \vdots & & & \\ y(N-1) & y(N-2) & \cdots & y(N-L-1) \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} \approx 0 \qquad (*)$$

Remark: • If $w(n) = 0$, Eq $(*)$ holds exactly.

• If $w(n) = 0$, Eq $(*)$ gives <u>EXACT</u> frequency estimates.

<u>Consider next the rank of</u>

$$\mathbf{X} = \begin{bmatrix} x(L-1) & \cdots & x(0) \\ \vdots & & \\ x(N-2) & \cdots & x(N-L-1) \end{bmatrix}$$

Note

$$\begin{bmatrix} x(0) \\ \vdots \\ x(N-L-1) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ e^{j\omega_1} & \cdots & e^{j\omega_K} \\ \vdots & & \vdots \\ e^{j(N-L-1)\omega_1} & \cdots & e^{j(N-L-1)\omega_K} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_K \end{bmatrix}$$

$$\begin{bmatrix} x(1) \\ \vdots \\ x(N-L) \end{bmatrix} = \begin{bmatrix} 1 & .. & 1 \\ e^{j\omega_1} & .. & e^{j\omega_K} \\ \vdots & \vdots & \\ e^{j(N-L-1)\omega_1} & .. & e^{j(N-L-1)\omega_K} \end{bmatrix} \begin{bmatrix} \beta_1 e^{j\omega_1} \\ \vdots \\ \beta_K e^{j\omega_K} \end{bmatrix}$$

$$\Rightarrow \mathbf{X} = \begin{bmatrix} 1 & \cdots & 1 \\ e^{j\omega_1} & \cdots & e^{j\omega_K} \\ \vdots & \vdots & \\ e^{j(N-L-1)\omega_1} & \cdots & e^{j(N-L-1)\omega_K} \end{bmatrix} \begin{bmatrix} \beta_1 & & 0 \\ & \ddots & \\ 0 & & \beta_K \end{bmatrix}$$

$$\begin{bmatrix} e^{j(L-1)\omega_1} & \cdots & e^{j\omega_1} & 1 \\ e^{j(L-1)\omega_2} & \cdots & e^{j\omega_2} & 1 \\ \vdots & \vdots & \vdots & \\ e^{j(L-1)\omega_K} & \cdots & e^{j\omega_K} & 1 \end{bmatrix}$$

Remark: $\underline{\text{If } N-L-1 \geq K \text{ and } L \geq K, \mathbf{X} \text{ is of rank } K.}$

From (*)

$$\underbrace{\begin{bmatrix} y(L-1) & \cdots & y(0) \\ \vdots & & \vdots \\ y(N-2) & \cdots & y(N-L-1) \end{bmatrix}}_{\mathbf{Y}} \begin{bmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_L \end{bmatrix} \approx - \underbrace{\begin{bmatrix} y(L) \\ \vdots \\ y(N-1) \end{bmatrix}}_{\mathbf{y}}$$

Remark: A rank K approximation of $\mathbf{Y}$ has " $\underline{\text{Noise Cleaning}}$ " effect.

$$\text{Let} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \begin{matrix} K \\ L-K \end{matrix}$$

$$\begin{bmatrix} \hat{\tilde{a}}_1 \\ \vdots \\ \hat{\tilde{a}}_L \end{bmatrix} = -\mathbf{V}_1 \mathbf{\Sigma}_1^{-1} \mathbf{U}_1^H \begin{bmatrix} y(L+1) \\ \vdots \\ y(N-1) \end{bmatrix}. \qquad (\dagger)$$

Summary of SVD Prony Estimator.

Step 1. Form $\mathbf{Y}$ and compute SVD of $\mathbf{Y}$

Step 2. Determine $\hat{\tilde{\mathbf{a}}}$ with (†)

Step 3. Compute the roots from $\hat{\tilde{\mathbf{a}}}$. Pick K roots that are nearest the unit circle. Obtain frequency estimates as phases of the roots.

Remark: • Although one cannot guarantee that the K roots nearest the unit circle give the best frequency estimates, empirical results show that this is true most often.

• A more accurate method is obtained by "cleaning" (i.e., rank K approximation of) the matrix $[\mathbf{Y} \;\vdots\; \mathbf{y}]$.

Pisarenko and MUSIC Methods

Remark: Pisarenko method is a special case of MUSIC (Multiple Signal Classification) method.

Recall:

$$\tilde{\mathbf{y}}_M(n) = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-M+1) \end{bmatrix}$$

$$\mathbf{A}_M = \begin{bmatrix} 1 & \cdots & 1 \\ e^{-j\omega_1} & \cdots & e^{-j\omega_K} \\ \vdots & \cdots & \vdots \\ e^{-j(M-1)\omega_1} & \cdots & e^{-j(M-1)\omega_K} \end{bmatrix},$$

$$\tilde{\mathbf{x}}(n) = \begin{bmatrix} x_1(n) \\ \vdots \\ x_K(n) \end{bmatrix},$$

$$\tilde{\mathbf{w}}_M(n) = \begin{bmatrix} w(n) \\ \vdots \\ w(n-M+1) \end{bmatrix}$$

$$\boxed{\tilde{\mathbf{y}}_M(n) = \mathbf{A}_M \tilde{\mathbf{x}}(n) + \tilde{\mathbf{w}}_M(n)}$$

$$\begin{aligned} \text{Let} \quad \mathbf{R} &= E\left\{\tilde{\mathbf{y}}_M(n)\tilde{\mathbf{y}}_M^H(n)\right\} \\ &= E\left\{\mathbf{A}_M \tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^H(n)\mathbf{A}_M^H\right\} \\ &+ E\left\{\tilde{\mathbf{w}}_M(n)\tilde{\mathbf{w}}_M^H(n)\right\} \end{aligned}$$

$$\Rightarrow \quad \boxed{\begin{aligned} \mathbf{R} &= \mathbf{A}_M \mathbf{P} \mathbf{A}_M^H + \sigma^2 \mathbf{I}, \\ \mathbf{P} &= \begin{bmatrix} \alpha_1^2 & & 0 \\ & \ddots & \\ 0 & & \alpha_K^2 \end{bmatrix}. \end{aligned}}$$

Remarks: • rank $(\mathbf{A}_M \mathbf{P} \mathbf{A}_M^H) = K$ if $M \geq K$.

• If $M \geq K, \mathbf{A}_M \mathbf{P} \mathbf{A}_M^H$ has K positive eigenvalues and $M - K$ zero eigenvalues. We shall consider $M \geq K$ below.

• Let the positive eigenvalues of $\mathbf{A}_M \mathbf{P} \mathbf{A}_M^H$ be denoted

$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_K$$

The eigenvalues of $\mathbf{R}$ are:

$$\text{Two groups} \begin{cases} \lambda_k = \tilde{\lambda}_k + \sigma^2, & k = 1, \cdots, K. \\ \lambda_k = \sigma^2, & k = K+1, \cdots, M \end{cases}$$

Let $\mathbf{s}_1, \cdots, \mathbf{s}_K$ be the eigenvectors of $\mathbf{R}$ that correspond to $\lambda_1, \cdots, \lambda_K$.

Let $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_K]$

Let $\mathbf{s}_{K+1}, \cdots, \mathbf{s}_M$ be the eigenvectors of $\mathbf{R}$ that correspond to $\lambda_{K+1}, \cdots, \lambda_M$.

Let $\mathbf{G} = [\mathbf{s}_{K+1}, \cdots, \mathbf{s}_M]$

$$\mathbf{RG} = \mathbf{G} \begin{bmatrix} \sigma^2 & & 0 \\ & \ddots & \\ 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 \mathbf{G}$$

$$\begin{aligned} \mathbf{RG} &= \left(\mathbf{A}_M \mathbf{P} \mathbf{A}_M^H + \sigma^2 \mathbf{I} \right) \mathbf{G} \\ &= \mathbf{A}_M \mathbf{P} \mathbf{A}_M^H \mathbf{G} + \sigma^2 \mathbf{G} \\ &\Rightarrow \mathbf{A}_M \mathbf{P} \mathbf{A}_M^H \mathbf{G} = 0 \quad \Rightarrow \mathbf{A}_M^H \mathbf{G} = 0 \end{aligned}$$

Remark:

Let the linearly independent K columns of $\mathbf{A}_M$ define K -dimensional signal subspace

* Then the eigenvectors of $\mathbf{R}$ that correspond to the $M - K$ smallest eigenvalues are orthogonal to the signal subspace.

* The eigenvectors of $\mathbf{R}$ that correspond to the K largest eigenvalues of $\mathbf{R}$ span the same signal subspace as $\mathbf{A}_M$.

$\Rightarrow \mathbf{A}_M = \mathbf{SC}$ for a $K \times K$ non-singular $\mathbf{C}$.

MUSIC:

The true frequency values $\{\omega_k\}_{k=1}^{K}$ are the only solutions of

$$\mathbf{a}_M^H(\omega)\mathbf{G}\mathbf{G}^H\mathbf{a}_M(\omega) = 0.$$

$$\mathbf{a}_M(\omega) = \begin{bmatrix} 1 \\ e^{-j\omega} \\ \vdots \\ e^{-j\omega(M-1)} \end{bmatrix}.$$

Steps in MUSIC:

Step 1: Compute $\hat{\mathbf{R}} = \frac{1}{N}\sum_{n=M}^{N} \tilde{\mathbf{y}}_M(n)\tilde{\mathbf{y}}_M^H(n)$, and its eigendecomposition.

Form $\hat{\mathbf{G}}$ whose columns are the eigenvectors of $\hat{\mathbf{R}}$ that correspond to the $M-K$ smallest eigenvalues of $\hat{\mathbf{R}}$.

Step 2a (Spectral MUSIC): Determine the frequency estimates as the locations of the K highest peaks of the MUSIC spectrum

$$\frac{1}{\mathbf{a}_M^H(\omega)\hat{\mathbf{G}}\hat{\mathbf{G}}^H\mathbf{a}_M(\omega)}, \quad \omega \in [-\pi, \pi]$$

Step 2b (Root MUSIC): Determine the frequency estimates as angular positions (phases) of K (pairs of reciprocal) roots of equation

$$\mathbf{a}_M^H\left(z^{-1}\right)\hat{\mathbf{G}}\hat{\mathbf{G}}^H\mathbf{a}_M(z) = 0$$

that are closest to the unit circle

$$\mathbf{a}_M(z) = \begin{bmatrix}1 & z^{-1} & \cdots & z^{-M+1}\end{bmatrix}^T, \textit{i.e.}, \quad \mathbf{a}_M(z)|_{z=e^{j\omega}} = \mathbf{a}_M(\omega)$$

Pisarenko Method = (MUSIC with $M = K + 1$)

Remarks:

- Pisarenko method is not as good as MUSIC.
- M in MUSIC should not be too large due to poor accuracy of $\hat{r}(k)$ for large k.

ESPRIT Method

(Estimation of Signal Parameters by Rotational Invariance Techniques)

$$\mathbf{A}_M = \begin{bmatrix} 1 & \cdots & 1 \\ e^{-j\omega_1} & \cdots & e^{-j\omega_K} \\ \vdots & & \\ e^{-j(M-1)\omega_1} & \cdots & e^{-j(M-1)\omega_K} \end{bmatrix}$$

Let $\mathbf{B}_1 =$ first $M-1$ rows of $\mathbf{A}_M$, $\mathbf{B}_2 =$ last $M-1$ rows of $\mathbf{A}_M$.

$$\mathbf{B}_2\mathbf{D} = \mathbf{B}_1,$$

$$\mathbf{D} = \begin{bmatrix} e^{j\omega_1} & & 0 \\ & \ddots & \\ 0 & & e^{j\omega_K} \end{bmatrix}$$

Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be formed from $\mathbf{S}$ the same way as $\mathbf{B}_1$ and $\mathbf{B}_2$ from $\mathbf{A}_M$

Recall: $\mathbf{S} = \mathbf{A}_M \mathbf{C}$

$$\Rightarrow \begin{cases} \mathbf{S}_1 = \mathbf{B}_1 \mathbf{C} = \mathbf{B}_2 \mathbf{D} \mathbf{C}. \\ \mathbf{S}_2 = \mathbf{B}_2 \mathbf{C} \end{cases}$$

$$\mathbf{S}_2 \mathbf{C}^{-1} = \mathbf{B}_2$$

$$\Rightarrow \quad \mathbf{S}_1 = \mathbf{S}_2 \mathbf{C}^{-1} \mathbf{D} \mathbf{C} \stackrel{\triangle}{=} \mathbf{S}_2 \boldsymbol{\Psi}.$$

$\Rightarrow$

$$\boxed{\boldsymbol{\Psi} = \left(\mathbf{S}_2^H \mathbf{S}_2\right)^{-1} \mathbf{S}_2^H \mathbf{S}_1.}$$

The diagonal elements of $\mathbf{D}$ are the eigenvalues of $\boldsymbol{\Psi}$.

Steps of ESPRIT: Step 1: $\hat{\boldsymbol{\Psi}} = \left(\hat{\mathbf{S}}_2^H \hat{\mathbf{S}}_2\right)^{-1} \hat{\mathbf{S}}_2^H \hat{\mathbf{S}}_1$

Step 2: Frequency estimates are angular positions of the eigenvalues of $\hat{\boldsymbol{\Psi}}$.

Remarks:

- $\hat{\mathbf{S}}_2 \boldsymbol{\Psi} \approx \hat{\mathbf{S}}_1$

can also be solved with <u>Total Least Squares Method</u>

- Since $\boldsymbol{\Psi}$ is $K \times K$ matrix, we <u>do not</u> need to pick K roots nearest the unit circle, which could be wrong roots.

- ESPRIT does not require the search over parameter space, as required by Spectral MUSIC.

All of these remarks make ESPRIT a recommended method !

Sinusoidal Parameter Estimation in the Presence of Colored Noise via RELAX

$$y(n) = \sum_{k=1}^{K} \beta_k e^{j\omega_k n} + e(n)$$

- β_k = Complex amplitudes, unknown.
- ω_k = Unknown frequencies.
- $e(n)$ = Unknown AR or ARMA noise.

Consider the Non-linear least-squares (NLS) method.

$$g = \sum_{n=0}^{N-1} \left| y(n) - \sum_{k=1}^{K} \beta_k e^{j\omega_k n} \right|^2$$

Remarks:

- $\hat{\beta}_k$ and $\hat{\omega}_k, \quad k = 1, \cdots, K$ are found by minimizing g .
- When $e(n)$ is zero mean Gaussian white noise, this NLS method is the ML method.
- When $e(n)$ is non-white noise, NLS method gives asymptotically ($N \to \infty$) statistically efficient estimates of $\hat{\omega}_k$ and $\hat{\beta}_k$ despite the fact that NLS is not an ML method for this case.
- The non-linear minimization is a difficult problem.

Remarks:

- Concentrating out $\{\beta_k\}$ gives

$$\hat{\boldsymbol{\omega}} = \quad \text{argmax}_{\omega} \left[\mathbf{y}^H \mathbf{B} \left(\mathbf{B}^H \mathbf{B} \right)^{-1} \mathbf{B}^H \mathbf{y} \right]$$

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{B}^H \mathbf{B} \right)^{-1} \mathbf{B}^H \mathbf{y} \Big|_{\omega = \hat{\omega}} .$$

- Concentrating out $\{\beta_k\}$, instead of simplifying the problem, actually complicates the problem.

- The RELAX algorithm is a <u>relaxation - based</u> optimization approach.

- RELAX is both <u>computationally and conceptually simple</u>.

Preparation:

$$\text{Let} \quad y_k(n) = y(n) - \sum_{i=1, i\neq k}^{K} \hat{\beta}_i e^{j\hat{\omega}_i n}$$

* $\hat{\beta}_i$ and $\hat{\omega}_i$, $i \neq k$, are assumed given, known, or estimated.

$$\text{Let} \quad g_k = \sum_{n=0}^{N-1} \left| y_k(n) - \beta_k e^{j\omega_k n} \right|^2.$$

* Minimizing g_k gives:

$$\hat{\omega}_k = \quad \text{argmax}_{\omega_k} \left| \sum_{n=0}^{N-1} y_k(n) e^{-j\omega_k n} \right|^2.$$

$$\hat{\beta}_k = \frac{1}{N} \sum_{n=0}^{N-1} y_k(n) e^{-j\omega_k n} \Big|_{\omega_k = \hat{\omega}_k}.$$

Remarks:

$$\sum_{n=0}^{N-1} y_k(n)e^{-j\omega_k n} \quad \text{is the DTFT of } y_k(n)!$$

(can be computed via FFT and zero-padding.)

- $\hat{\omega}_k$ corresponds to the peak of the Periodogram!
- $\hat{\beta}_k$ is the peak height (complex number!) of the DTFT of $y_k(n)$ (at $\hat{\omega}_k$) divided by N.

The RELAX Algorithm

Step 1: Assume K =1. Obtain $\hat{\omega}_1$ and $\hat{\beta}_1$ from $y(n)$.

Step 2: Obtain $y_2(n)$ by assuming K=2 and using $\hat{\omega}_1$ and $\hat{\beta}_1$ obtained from Step 1.

$$\text{Iterate until converg.} \left\{ \begin{array}{l} \text{Obtain } \hat{\omega}_2 \text{ and } \hat{\beta}_2 \text{ from } y_2(n) \\ \text{Obtain } y_1(n) \text{ by using } \hat{\omega}_2 \text{ and } \hat{\beta}_2 \\ \text{and reestimate } \hat{\omega}_1 \text{ and } \hat{\beta}_1 \text{ from } y_1(n) \end{array} \right.$$

Step 3: Assume $K = 3$.

Obtain $y_3(n)$ from $\hat{\omega}_1$,$\hat{\beta}_1$, $\hat{\omega}_2, \hat{\beta}_2$. Obtain $\hat{\omega}_3$ and $\hat{\beta}_3$ from $y_3(n)$.

Obtain $y_1(n)$ from $\hat{\omega}_2, \hat{\beta}_2, \hat{\omega}_3, \hat{\beta}_3$. Reestimate $\hat{\omega}_1$ and $\hat{\beta}_1$ from $y_1(n)$.

Obtain $y_2(n)$ from $\hat{\omega}_1, \hat{\beta}_1, \hat{\omega}_3, \hat{\beta}_3$. Reestimate $\hat{\omega}_2$ and $\hat{\beta}_2$ from $y_2(n)$.

Iterate until g does not decrease “significantly” anymore !

Step 4: Assume $K = 4, \cdots$

$$\vdots$$

Continue until K is large enough!

Remark:

- RELAX is found to perform better than existing high-resolution algorithms, especially in obtaining better $\hat{\beta}_k$, $k = 1, \cdots, K$
- RELAX is more robust to the choice of K and the data model errors.