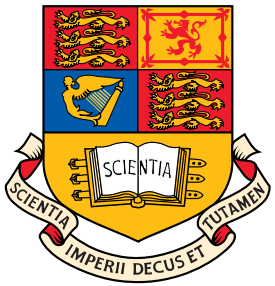
Spectrum Estimation & Adaptive SP
Modern Spectrum Estimation

Danilo Mandic

room 813, ext: 46271



Department of Electrical and Electronic Engineering
Imperial College London, UK

d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

Outline

- Much in spectrum estimation depends on the way we approximate ACF
- Often, we have some knowledge of the processes we want to estimate (comms, biomedical), and are not interested in the whole spectrum
  Indeed, why would we e.g. estimate the high frequency noise, if our interest was only in the frequency of heart beats or respiration rate?
- **Modern Spectrum Estimation answers the natural questions:**
  - Can we reduce the variance by using optimisation? **(MV-SE)**
  - Can we afford to estimate only "important" parts of the spectrum, and simply not care about the rest?
  - Can we extrapolate the autocorrelation function using a model-based approach? **(ARMA spectrum estimation)**
  - Can we extrapolate ACF using entropy? **(Maximum Entropy SE)**
  - Is it possible to guarantee high accuracy of spectral estimates for a set of discrete spectral lines? **(Pisarenko)**
  - Can we separate the signal and noise spaces using eigenanalysis, and estimate spectrum of only the signal of interest? **(MUSIC)**

Summary of nonparametric methods

- **Periodogram:** straightforward and approximates the whole spectrum but has problems with large variance and poor resolution;

- **Limitations:** Relying on DTFT of an estimated autocorrelation sequence, the resolution of these methods is determined by the length of the data record
  - frequency resolution $\sim 1/N$
  - sidelobe supression in the spectrum depend on the type of the window
  - problems when very few data points are present (genomic SP)
  - no matter what modification we take, we cannot bypass the "curse of Fourier"

- Can we go beyond the use of FFT?

Let us first look at two simple methods to help circumvent this problem:

Minimum Variance Estimation and Maximum Entropy Estimation

Filterbank methods: Motivation

The idea behind filterbank methods:

- Parametrise $P_{xx}(\omega)$ by some finite-dimensional parametrisation, e.g. a bank of band-pass filters with center frequencies $\omega_i, i = 0, \ldots, N-1$
- Smoothen the power spectrum in every band $[\omega_i - \Delta, \omega_i + \Delta]$ by assuming that $P_{xx}(\omega_i)$ is constant over the band $[\omega_i - \Delta, \omega_i + \Delta]$

$$\textbf{Recall}: \text{total power} \quad r(0) = E\{|x(n)|^2\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} P(\omega)d\omega$$

Also recall: The quantity $P(\omega)$ is called **power spectral density** as it represents the distribution of signal power over frequencies

Therefore, the power in an infinitesimally small freq. band $d\omega$ is given by

$$P(\omega)\frac{d\omega}{2\pi} \qquad \text{in the band} \quad (\omega - \frac{d\omega}{2}, \omega + \frac{d\omega}{2})$$

☞ **Total power = integral of these infinitesimal contributions over frequencies**

Also recall that $\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{\jmath\omega(k-m)} = \delta_{k,m}$ ↬ Kronecker delta.

Filterbank interpretation of the Periodogram

Recall: Output PSD of a lin. sys. $P_{yy}(f) = |H(f)|^2 P_{xx}(f)$

Consider an FIR filter of length N defined as

$$h_i(n) = \frac{1}{N} e^{\jmath n \omega_i} w_R(n) = \begin{cases} \frac{1}{N} e^{\jmath n \omega_i} & ; \quad 0 \le n \le N \\ 0 & ; \quad \text{otherwise} \end{cases}$$

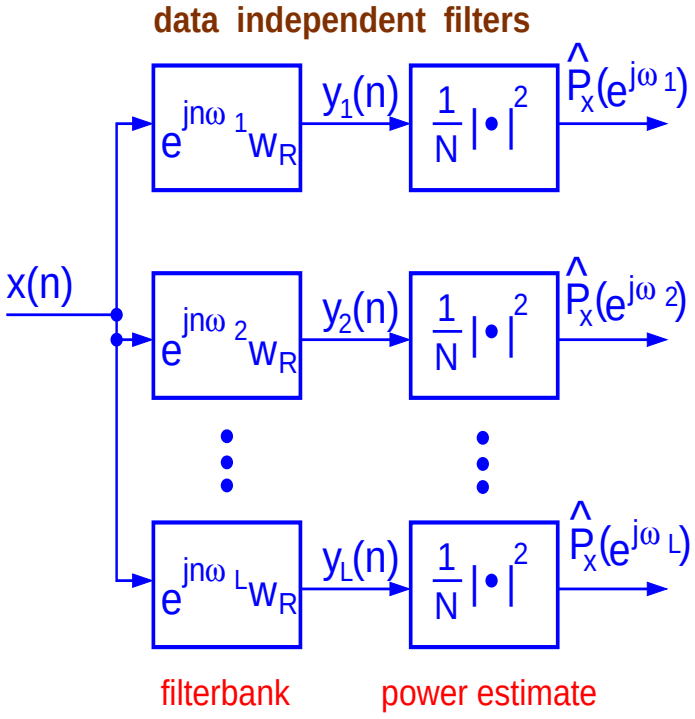
Its frequency response is

$$H_i(e^{\jmath\omega}) = \sum_{n=0}^{N-1} h_i(n) e^{-\jmath n \omega} = e^{-\jmath(\omega-\omega_i)\frac{(N-1)}{2}} \frac{\sin[N(\omega-\omega_i)/2]}{N \sin[(\omega-\omega_i)/2]}$$

- **center freq ω_i, bandpass $\Delta\omega \approx \frac{2\pi}{N}$**
- $y_i = x(n) * h(n) = \frac{1}{N} \sum x(k) e^{\jmath(n-k)\omega}$
  $|H_i(e^{\jmath\omega_i})| = 1 \;\Rightarrow\; P_x(e^{\jmath\omega_i}) = P_y(e^{\jmath\omega_i})$
- $E\{|y_i(n)|^2\} \approx \frac{\Delta\omega}{2\pi} P_x(e^{\jmath\omega_i}) = \frac{1}{N} P_x(e^{\jmath\omega_i})$
  Therefore $P_x(e^{\jmath\omega_i}) \approx N E\{|y_i(n)|^2\}$
- Assume $\hat{E}\{|y_i(n)|^2\} = |y_i(N-1)|^2$, then

$$|y_i(N-1)|^2 = \frac{1}{N^2} \Big| \sum_{k=0}^{N-1} x(k) e^{-\jmath k \omega_i} \Big|^2$$

$$\hat{P}_x(e^{\jmath\omega_i}) = \frac{1}{N} \Big| \sum_{k=0}^{N-1} x(k) e^{-\jmath k \omega_i} \Big|^2$$



Periodogram as a filterbank: Some facts

☞ In the simplest case of the periodogram, $x(n)$ is filtered with a bank of bandpass filters with frequency response $H(\omega) = \frac{\sin[N(\omega-\omega_i)/2]}{N\sin[(\omega-\omega_i)/2]}$.

- The power of each of the filtered signals is estimated using a one–point sample average
$$\hat{E}\left\{|y_i(n)|^2\right\} = |y_i(N-1)|^2$$
- Periodogram is formed by dividing this power estimate by the filter bandwidth $\Delta = 2\pi/N$, so that
$$\hat{P}_x(e^{\jmath\omega_i}) = \frac{E\left\{|y_i(n)|^2\right\}}{\Delta/2\pi}$$
- Each filter in the filter bank of a periodogram is the same $\Rightarrow$ these filters are **data independent**

Consequence: When a random process contains power in frequency bands within the sidelobes of the bandpass filter, leakage through the sidelobes will lead to significant distortion in the power estimates

Minimum Variance Method (MVM) for Spectrum Estimation

The power spectrum is estimated by filtering a process with a **bank of narrowband** bandpass filters.

Motivation: Let us look first at filtering of a WSS random process with a narrowband bandpass filter:

$x(n) \rightarrow$ zero mean WSS random process with a PS $P_x(e^{\jmath\omega})$ and

$g_i(n) \rightarrow$ ideal bandpass filter with a bandwidth Δ and center frequency ω_i

$$|G_i(e^{\jmath\omega})| = \begin{cases} 1, & |\omega - \omega_i| < \Delta/2 \\ 0, & otherwise \end{cases}$$

If $x(n)$ is filtered with $g_i(n)$ then the power of the output process of the filter i, denoted by $y_i(n)$ is

$$E\{|y_i(n)|^2\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_i(e^{\jmath\omega})d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{\jmath\omega})|G_i(e^{\jmath\omega})|^2 d\omega$$

$$= \frac{1}{2\pi}\int_{\omega_i-\Delta/2}^{\omega_i+\Delta/2} P_x(e^{\jmath\omega})d\omega$$

Key: To design the filterbank g_i so that out-of-band signal power is rejected

MVM: Physical meaning

"Ideally" bandwidth Δ is small enough so that $P_x(e^{\jmath\omega}) \approx const.$ for $|\omega - \omega_i| < \Delta/2$. Therefore, the proportion of power contained in $y_i(n)$

$$E\left\{|y_i(n)|^2\right\} = \frac{1}{2\pi}\int_{\omega_i-\Delta/2}^{\omega_i+\Delta/2} P_x(e^{\jmath\omega})d\omega \approx P_x(e^{\jmath\omega_i})\frac{\Delta}{2\pi}$$

- We can now estimate PSD of $x(n)$ at frequency $\omega = \omega_i$ from the filtered process.
- **This is achived by estimating the power in $y_i(n)$ and dividing by the normalised filter bandwidth $\Delta/2\pi$.**
- **Solution:** allow each filter in the filter bank to be **data adaptive** and **optimum** in the sense of rejecting as much out–of–band signal power as possible.
- Set
$$\hat{P}_x(e^{\jmath\omega_i}) = \frac{E\left\{|y_i(n)|^2\right\}}{\Delta/2\pi}$$
that is, the estimated power estimated divided by the filter bandwidth.

MVM: Some supporting mathematics

To estimate PSD of $x(n)$ at frequency ω_i, we use $g_i(n) \hookrightarrow$ a complex–valued bandpass FIR filter of order p.

- To ensure the filter does not later the power in the input process at frequency ω_i, $G_i(\omega)$ will be constrained to have gain one at $\omega = \omega_i$, that is $G_i(\omega) = \sum_{n=0}^{p} g_i(n) e^{-\jmath n \omega_i} = 1$
- For $\mathbf{g}_i = [g_i(0), \ldots, g_i(p)]^T$ vector of filter coefficients and $\mathbf{e}_i = [1, e^{\jmath\omega_i}, \ldots, e^{\jmath p\omega_i}]^T$ vector of complex exponentials, the above constraint results in $\mathbf{g}_i^H \mathbf{e}_i = 1$
- We have the following form for the power $E\left\{|y_i(n)|^2\right\} = \mathbf{g}_i^H \mathbf{R}_{xx} \mathbf{g}_i$
- Optimisation problem becomes: Minimise $E\left\{|y_i(n)|^2\right\}$ s.t. $\mathbf{g}_i^H \mathbf{e}_i = 1$

$$\min_{\mathbf{g}_i} E\left\{|y_i(n)|^2\right\} = \frac{1}{\mathbf{e}_i^H \mathbf{R}_{xx}^{-1} \mathbf{e}_i} \Rightarrow \mathbf{g} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{e}}{\mathbf{e}^H \mathbf{R}_{xx}^{-1}\mathbf{e}} \Rightarrow \hat{\sigma}_x^2(\omega) = \frac{1}{\mathbf{e}^H \mathbf{R}_{xx}^{-1}\mathbf{e}}$$

index i dropped since these estimates are valid for all frequencies

MVM: Practical considerations

Although there are several criteria that may be used to define bandwidth, the simplest is to use the value of Δ that produces the correct PSD for white noise.

After some calculation (*** see previous slide ***) the estimate of power in $x(n)$ is

$$\hat{P}_{MV}(e^{\jmath\omega}) = \frac{p+1}{\mathbf{e}^H \mathbf{R}_x^{-1} \mathbf{e}}$$

where p is the filter order, and $\mathbf{e}_i = [1, e^{\jmath\omega}, \ldots, e^{\jmath p\omega}]^T$.

This is the minimum variance spectral estimate (MVM).

☞ The PSD is **defined in terms of the autocorrelation matrix**

☞ We require its inverse

◦ Since $\mathbf{R}_x$ is Toeplitz, the inverse may be found using either the Levinson recursion or the Cholesky decomposition.

◦ Both these methods we will explain later in the course.

Example 1: PSD of white noise

Task: Estimate power in white noise that has a variance of σ_x^2.

☞ We know that for white noise the autocorrelation matrix $\mathbf{R}_{xx} = \sigma_x^2 \mathbf{I}$

$$\Rightarrow \quad \text{minimum variance BP filter is} \quad \mathbf{g} = \frac{\mathbf{R}_{xx}^{-1}\mathbf{e}}{\mathbf{e}^H \mathbf{R}_{xx}^{-1}\mathbf{e}} = \frac{\sigma_x^{-2}\mathbf{e}}{\sigma_x^{-2}\mathbf{e}^H\mathbf{e}} = \frac{1}{p+1}\mathbf{e}$$

$$\text{while the power estimate at frequency } \omega \quad \rightsquigarrow \quad \hat{\sigma}_x^2(\omega) = \frac{1}{\mathbf{e}^H \mathbf{R}_{xx}^{-1}\mathbf{e}} = \frac{1}{1+p}\sigma_x^2$$

- Power is independent of ω, as the distribution of power is constant
- The estimated power decreases as the filter order p increases!
  (Total power over a freq. band $\Delta\omega$ is $\sigma_x^2 \Delta\omega$. As p increases Δ decreases $\Rightarrow$ less power passes through the filter)
- To obtain an estimate of power spectrum we must divide the power estimate by the bandwidth of the filter. The simplest way is to use the value of Δ that produces the correct PSD for white noise.

☞ MV estimate of the power in WN is $E\{|y_i(n)|^2\} = \sigma_x^2/(p+1)$ and

$$\hat{P}_{xx}(\omega) = \frac{E\{|y_i(n)|^2\}}{\Delta/2\pi} = \frac{\sigma_x^2}{p+1}\frac{2\pi}{\Delta} \qquad \text{If we set} \quad \Delta = \frac{2\pi}{p+1} \quad \text{then} \quad \hat{P}_{xx}(\omega) = \sigma_x^2$$

Matlab code and practical considerations

The following Matlab code is only a template:

```
x = x(:); % converts vector x into a column vector
R = covar(x,p); % covariance response of LTI models to
                % white noise inputs
[v,d] = eig(R); % [V,D] = EIG(X) produces a diagonal matrix D
                % of eigenvalues and a full matrix V whose columns
                % are the corresponding eigenvectors
                % so that X*V = V*D
U = diag(inv(abs(d)+eps)); % eps added to avoid division by zero
V = abs(fft(v,1024)).^2;
Px = 10*log10(p)-10*log10(V*U);
```

To perform the matrix inversion in an efficient way, in Statistical Signal Processing we use the so called ***Matrix Inversion Lemma*** (Woodbury's indentity).

For several forms of Matrix Inversion Lemma, see Lecture 0 (background)

Minimum Variance Spectrum Estimation: Summary

Despite the obvious advantages, there are also problems associated with the Minimum Variance Method (MVM):

- **Filter order.** The larger the filter order p the better the filter is in rejecting out–of–bound power $\not\mapsto$ we would ideally like $p \to \infty$
- **Practical considerations.** Often not practical, as the accuracy increases with p, and this increases computational complexity, $(p+1) \times (p+1)$ autocorrelation matrix $\mathbf{R}_{xx}$
- **Numerical sensitivity.** An inverse of a large $(p+1) \times (p+1)$ autocorrelation matrix $\mathbf{R}_{xx}$ is required $\to$ very inconvenient for large p
- **Finite data record.** Since for a finite (and often short) data record N, it is only possible to estimate $r_x(k)$ for $k = 0, 1, \ldots, N-1$, the filter order p is limited to $p \leq N$
- **Confidence in data.** In practice $p << N$, due to the large variance of ACF estimate for large k which are close to N

Notes

Notes

Notes

Maximum Entropy Method (MEM)

Motivation: Limitation of the classical approach to spectrum estimation:

Q1: For a data record of length N, the ACF can only be estimated for lags $|k| < N$, that is $\hat{r}_x = 0$ for $|k| \geq N$. However, many signals have ACFs that are nonzero for $|k| \geq N$

Q2: Windowing $\Rightarrow$ limits the resolution and accuracy of estimated spectrum

Solution: Extrapolation of the ACF so as to maximise spectral entropy!

$$P_x(e^{\jmath\omega}) = \sum_{k=-p}^{p} r_x(k)e^{-\jmath\omega} + \sum_{|k|>p} r_e(k)e^{-\jmath\omega} \qquad r_e \quad \text{are the extrapolated} \quad \text{ACF}$$

We must impose some constraints on the extrapolations:

C1: $P_x(e^{\jmath\omega})$ is always positive (an obvious contraint)

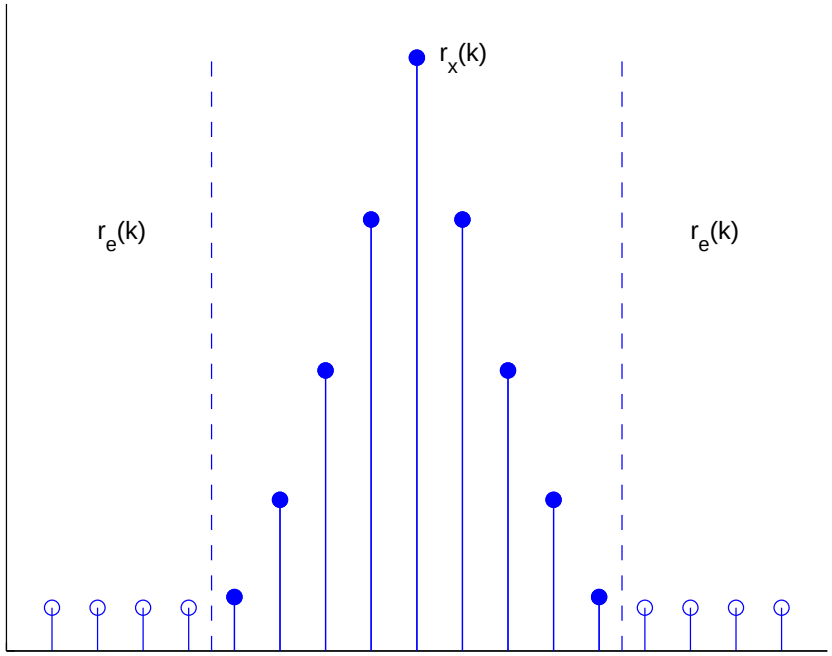
C2: Extrapolation performed so as to maximise the entropy of the process

Notice: Maximum entropy $\Leftrightarrow$ whitening of x $\Leftrightarrow$ "as flat as possible" P_x

☞ MEM imposes an *all pole AR(p) model* on the data

Extrapolating the autocorrelation sequence

Simple extrapolation with $r_e(k) = const$ = last $r_x(k)$ in the window

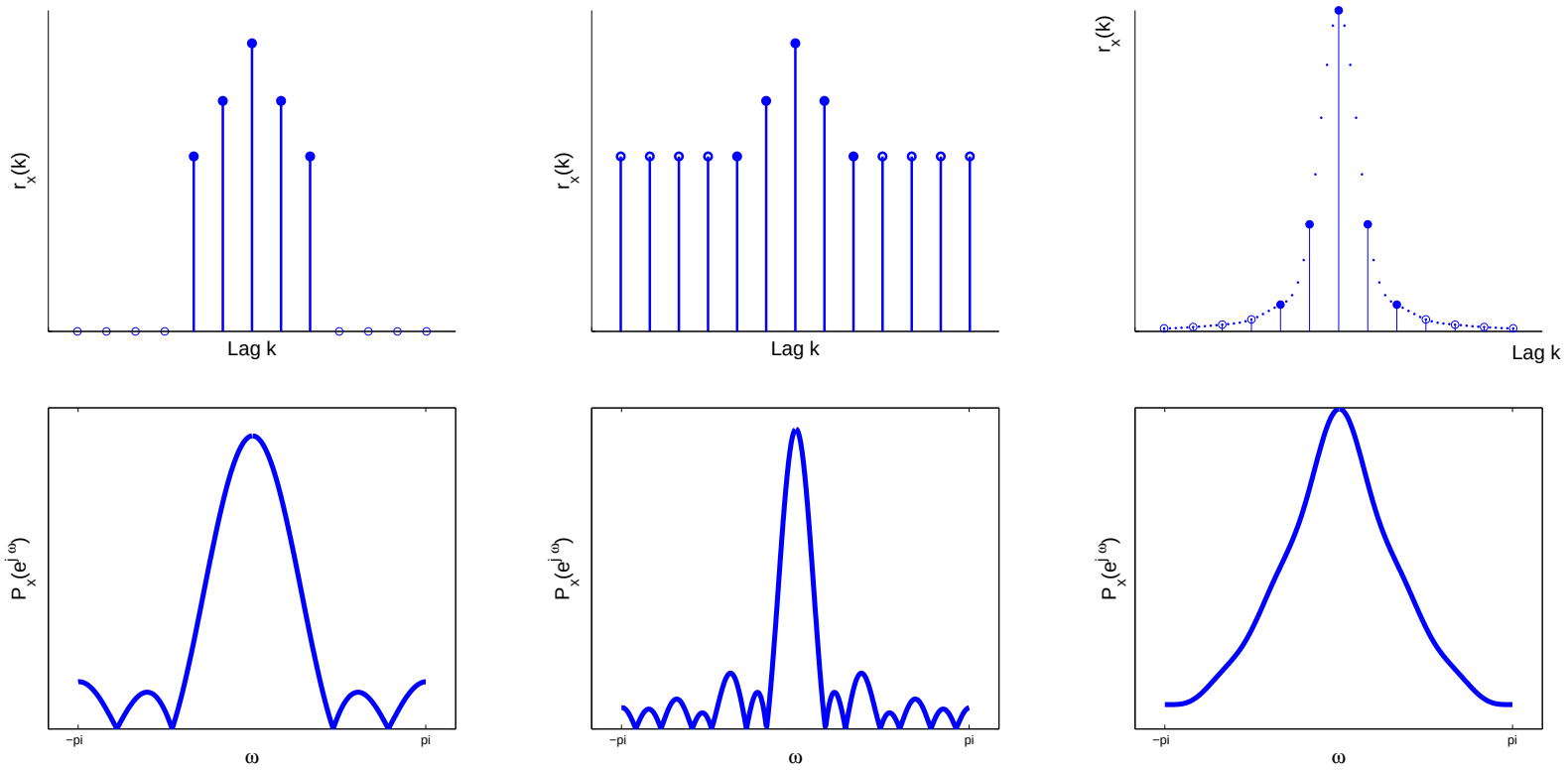


Different extrapolations will give different PSDs!

↪ **Find the extrapolation that produces spectrum as flat as possible**

Different Extrapolations and Associated Spectra

Top panel: autocorrelation sequences



Bottom panel: corresponding PSDs

What is the flattest PSD here?

MEM: Problem setting

Burg proposed to perform extrapolation in such a way so as to **maximise the entropy of the process.**

Entropy = measure of randomness or uncertainty

$\Rightarrow$ Maximum Entropy extrapolation is equivalent to:

find the sequence of autocorrelations, $r_e(k)$

s.t. $x(n)$ be as **white** (random) as possible.

Such constrainst place the **least amount of structure** on $x(n)$.

In terms of the power spectrum, this correspond to the constraint that $P_x(\omega)$ be **as flat as possible**

Recall: For a Gaussian random process with power spectrum $P_{xx}(\omega)$

$$Entropy: \qquad H(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln P_{xx}(\omega) d\omega$$

MEM ↬ derivation

$\Rightarrow$ for a Gaussian process with a given autocorr. sequence $r_x(k)$ for $|k| \leq p$
the Maximum Entropy Power Spectrum minimises entropy $H(x)$

subject to the constraint that the inverse DFT of $P_{xx}(\omega)$ equals the given set of autocorrelations for $|k| \leq p$, that is

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} P_{xx}(\omega)e^{\jmath k\omega}d\omega = r_x(k) \qquad |k| \leq p$$

The values of $r_e(k)$ that maximize the entropy may be found by setting the derivative of $H(x)$ wrt $r_e^*(k)$ equal to zero:

$$\frac{\partial H(x)}{\partial r_e^*(k)} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{P_{xx}(\omega)}\frac{\partial P_{xx}(\omega)}{\partial r_e^*}d\omega = 0 \qquad |k| > p$$

Notice that $\frac{\partial P_{xx}(\omega)}{\partial r_e^*} = e^{\jmath k\omega} \quad \Rightarrow \quad \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1}{P_{xx}(\omega)}e^{\jmath k\omega}d\omega = 0, \quad |k| > p.$

MEM Spectrum

Therefore:

$$Q_{xx}(\omega) = \frac{1}{P_{xx}(\omega)} = \sum_{k=-p}^{p} q_{xx}(k) e^{-\jmath k\omega}$$

$\Rightarrow \hat{P}_{mem}$ is an all–pole spectrum, given by

$$\hat{P}_{mem}(\omega) = \frac{|b(0)|^2}{A_p(\omega) A_p^*(\omega)} = \frac{|b(0)|^2}{|1 + \sum_{k=1}^{p} a_p(k) e^{-\jmath k\omega}|^2}$$

Alternatively

$$\hat{P}_{mem}(\omega) = \frac{|b(0)|^2}{|\mathbf{e}^H \mathbf{a}_p|^2}$$

Coefficients $\mathbf{a}[1, a_p(1), \ldots, a_p(p)]^T$ and $b(0)$ are found from the normal equations (Yule–Walker).

MEM spectum estimation: Calculating coefficients

We have seen that the MEM spectral estimate $\hat{P}_{mem}(\omega)$ produces an ACF sequence that matches the given values of $r_x(k)$ for $|k| \leq p$.

The coefficients $a_p(k)$ are the solution to the autocorrelation normal equations

$$\begin{bmatrix} r_x(0) & r_x^*(1) & \cdots & r_x^*(p) \\ r_x(1) & r_x(0) & \cdots & r_x^*(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p) & r_x(p-1) & \cdots & r_x(0) \end{bmatrix} \times \begin{bmatrix} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \varepsilon_p \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Procedure:

- Solve the autocorrelation normal equations for the all–pole coefficients $\mathbf{a}_p$ and ε
- The MEM spectrum is formed by slotting these coefficients into the equation for $\hat{P}_{mem}$

Summary

Since $\hat{P}_{mem}$ is an all–pole spectrum, the ACF $r_x(k)$ satisfies the Yule–Walker equations

$$r_x(l) = -\sum_{k=1}^{p} a_p(k) r_x(k-l), \quad l > 0$$

(Hint: can be written directly from the AR model of the data)

$\Rightarrow$ the MEM method extrapolates the ACF sequence according to this recursion.

In Matlab:

```
[a,e] = acm(x,p);
Px = 20*(log10(e)-log10(abs(fft(a,1024))));
```

Example 2: Maximum Entropy estimate of a complex exponential in noise

Problem: Find the MEM estimate of the complex exponential in white noise

$$x(n) = A_1 e^{\jmath n\omega_1} + w(n)$$

where $A_1 = |A_1|e^{\jmath\Phi}$, with Φ being a uniformly distributed random variable over $[-\pi, \pi]$, and $var(w(n)) = \sigma_w^2$.

Solution:

1) Find the AR coefficients $\mathbf{a}_p(k)$, from $\mathbf{R}_{xx} = |A_1|^2\mathbf{e}_1\mathbf{e}_1^H + \sigma_w^2\mathbf{I}$

2) From the formula for MEM spectrum

$$\hat{P}_{mem}(\omega) = \frac{1}{\sigma_w^2}\left[\sigma_x^2 + p|A_1|^2\right]\left[\sigma_x^2 + p|A_1|^2\right]$$

For a large SNR, $|A_1|^2 >> \sigma_w^2 \quad \Rightarrow \quad for \quad \omega = \omega_1$

$$\hat{P}_{mem}(\omega)_{|\omega=\omega_1} \approx p^2\frac{|A_1|^4}{\sigma_w^2}$$

☞ the peak in the MEM spectrum $\Leftrightarrow$ square of the power in the complex exponential.

MEM ↬ pro's and con's

- Works even in the absence of any information or constraints on a process $x(n)$
- Only a given set of ACF values $r_x(0), \ldots, r_x(p)$ needed
- the best way to estimate the power spectrum is to Fourier Transform the autocorrelation sequence formed from the given values together with an extrapolation that imposes the least amount of structure on the data: Maximum Entropy Extrapolation
- This is preferrable to the classical approach where $r_x(k) = 0 \quad |k| > p$
- Since MEM estimation imposes an all–pole model on the data, the estimated spectrum may not be very accurate if the data do not conform to this assumption
- The "goodness" of the MEM estimation as compared to the classical and MV approach depends critically on the all–pole assumption.

Notes

Notes

Modern Spectrum Estimation: Motivation

Limitation of classical methods: Not designed to incorporate information that may be available about the process in hand.

For example, in speech processing, an acoustic tube model for the vocal tract is very well approximated by an AR model, for which the power spectrum estimate is (a peaky autoregressive power spectrum)

$$P_x(e^{\jmath\omega}) = \frac{\sigma_w^2}{\left|1 + \sum_{k=1}^{p} a_p(k) e^{-\jmath k\omega}\right|^2}$$

Problem with periodogram: Its power spectrum is of MA type:

$$\hat{P}_{per}(e^{\jmath\omega}) = \sum_{k=-N}^{N} \hat{r}_x(k) e^{-\jmath k\omega} \quad \text{not ideal for peaky spectra}$$

Solution: Why do not we incorporate a model for the process (e.g. its ARMA model) directly into the spectrum estimation algorithm?

☞ **In this way we arrive at the parametric spectrum estimation.**

More on autocorrelation function (ACF) estimation

When no assuptions are made about the data, the two standard ACF estimators

1) The **standard unbiased** estimate ($r_x(k) \nrightarrow$ true ACF, $\hat{r}_x(k) \nrightarrow$ estim.)

$$E\{\hat{r}_x(k)\} = \frac{1}{N}\sum_{n=0}^{N-1-k} E\{x(n+k)x^*(n)\} = \frac{1}{N-k}\sum_{n=0}^{N-1-k} x(n+k)x(n) = \frac{N-k}{N} r_x(k)$$

- This estimate can take erratic values for large values of k
- It is not guaranteed to give a positive semidefinite sequence $\hat{r}_x(k)$

2) The **standard biased** estimate:

$$\hat{r}_x(k) = \sum_{n=-\infty}^{\infty} x_N(n+k)x_N^*(k) = \frac{1}{N} x_N(k) * x_N(-k) = \frac{1}{N}\sum_{n=0}^{N-1-k} x(n+k)x(n)$$

- This estimate is biased
- It produces positive semidefinite sequence $\hat{r}_x(k)$

For most data the ACF decays rapidly and $r_x(k)$ is small for large k

In that case (small k) the two above ACF estimators are similar.

Some practical covariance (or ACF) estimators

For data coming in real time and data history contained in the vector $\mathbf{x} = [x(0), \ldots, x(N-1)]^T$, some recursive covariance estimators:

$$1) \qquad c(N) = \frac{N-1}{N} c(N-1) + \frac{x^2(N)}{N}$$

$$2) \qquad c(N) = c(N-1) + \frac{x^2(N) - x^2(N-M+1)}{M} \quad M < N$$

$$3) \qquad c(N) = \lambda c(N-1) + (1-\lambda) x^2(n) \quad 0 \leq \lambda \leq 1, \quad \lambda \text{ usually close to } 1$$

It then makes sense to apply similar strategies to the whole covariance (correlation) matrix $\mathbf{C}$, for instance

$$\mathbf{C}(N) = \lambda \mathbf{C}(N-1) + \frac{1-\lambda}{N} \mathbf{x}(N) \mathbf{x}^T(N)$$

or, for a sliding window of size M

$$\mathbf{C}(k) = \mathbf{X}(k)\mathbf{X}^H(k) = \mathbf{C}(k-1) - \frac{\mathbf{x}(k-M)\mathbf{x}^H(k-M)}{M} + \frac{\mathbf{x}(k)\mathbf{x}^H(k)}{M}$$

Parametric spectra vs. line spectra

- Parametric (model-based) methods assume a generating model with known form. Our goal is to estimate parameters of such a model.

  We have two classes of parametric methods:

  1) Autoregressive moving average estimation of **continuous spectra**
  2) Eigen-based methods for the estimation of **line spectra**

$$y(n) = x(n) + w(n) = \sum_{i=1}^{p} A_i e^{-\jmath(\omega_i n + \Phi_i)} + w(n), \qquad w(n) \sim \mathcal{N}(0, \sigma^2)$$

$$r(k) = E\{y(n)y^*(n+k)\} = \sum_{i=1}^{p} A_i^2 e^{\jmath\omega_i k} + \sigma^2\delta(k,0)$$

$$P(\omega) = 2\pi \sum_{i=1}^{p} A_i^2 \delta(\omega - \omega_i) + \sigma^2 \quad \text{↬ becomes a linear regression problem}$$

$\Rightarrow$ The PSD consists of a "noise floor" and narrow lines corresponding to the different sinewaves (since $E\{e^{\jmath\omega_p}e^{\jmath\omega_l}\} = E\{e^{\jmath\omega_p}\}E\{e^{\jmath\omega_l}\} = 0$).

ARMA parametric SE: General procedure

1. Select an appropriate model
   - Autoregressive (AR) (all pole)
   - Moving average (MA) (all zero)
   - Autoregressive moving average (ARMA), i.e. for $w(k) \in \mathcal{N}(0, \sigma_w^2)$

$$y(k) = a_1 y(k-1) + \cdots + a_p y(k-p) + w(k) + b_1 w(k-1) + \cdots + b_q w(k-q) + w(k$$

   - Harmonic (complex exponentials in noise)

2. Estimate the model parameters from the given data

3. Estimate the power spectrum by incorporating the estimated parameters $(\hat{a}_1, \ldots, \hat{a}_p, \hat{b}_1, \ldots, \hat{b}_p)$ into the parametric form for the spectrum, that is

$$\hat{P}_x(e^{\jmath\omega}) = \frac{\left|\sum_{k=0}^{q} \hat{b}_q(k) e^{-\jmath k\omega}\right|^2}{\left|1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-\jmath k\omega}\right|^2}$$

Autoregressive Spectrum Estimation

ARMA spectrum is given by

$$\hat{P}_x(e^{\jmath\omega}) = \frac{\left|\sum_{k=0}^{q} \hat{b}_q(k) e^{-\jmath k\omega}\right|^2}{\left|1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-\jmath k\omega}\right|^2}$$

Advantage: It is possible to significantly improve the spectrum estimate with a parametric model.

Disadvantage: If the model is inappropriate for the process that is being analysed, inaccurate or misleading estimates may be obtained.

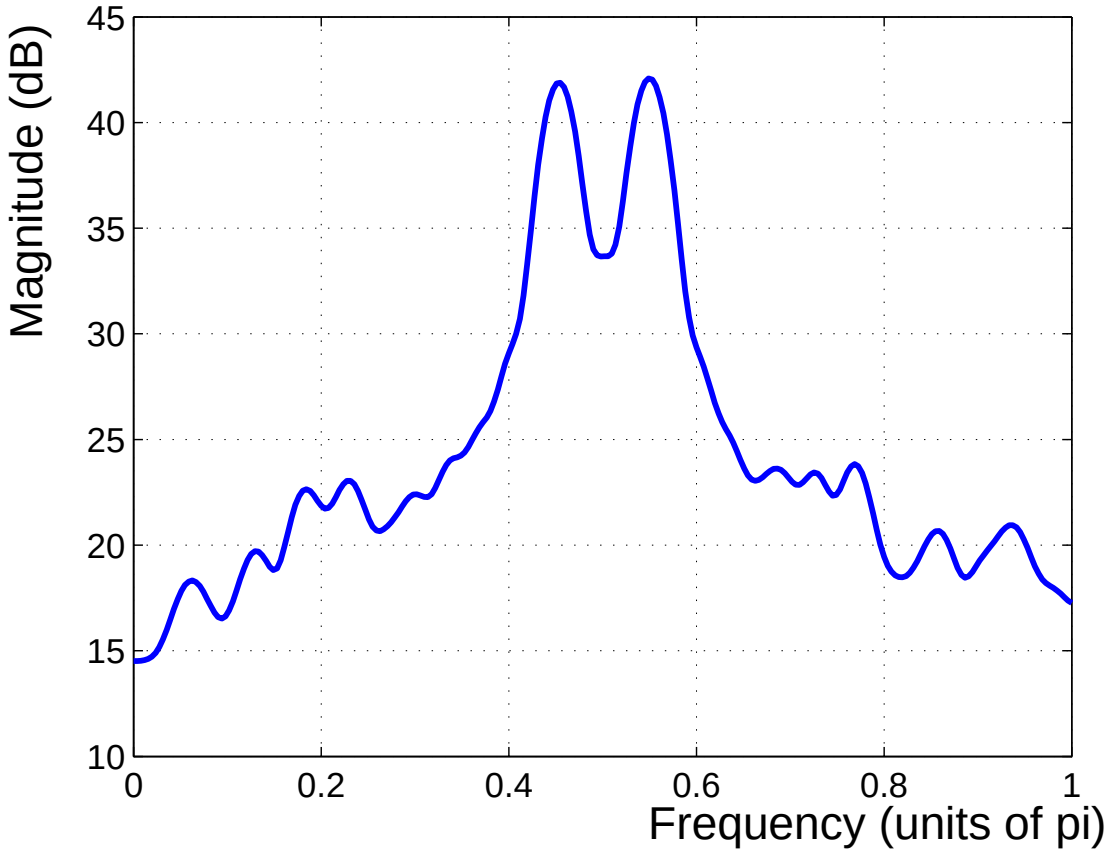
Example 3: Two sinusoids in white noise given by

$$x(n) = 5\sin(0.45\pi n + \Phi_1) + 5\sin(0.55\pi n + \Phi_2) + w(n)$$

Investigate: 1) all–pole model (Figure a)

2) MA model $x(n) = w(n) - w(n-2)$ $\qquad [P_{xx}(\omega) = 2 - 2\cos(\omega)]$

Example: Two sinusoids in WGN



All–pole model capable of estimating two sinewaves in noise!

Blackman-Tukey and MA: Smooth spectra $\not\hookrightarrow$ not suitable here

Example 3: Matlab code

```
clear all,clc
N = 64;   t = 1:N;
ph1 = 2*pi*rand-pi;   ph2 = 2*pi*rand-pi;
fftdatapoints = 512;
w = linspace(0,pi,fftdatapoints/2);
M = 2*N-1;
win = ones(1,M);   win = bartlett(M)';
noise = randn(1,N);

x = 5*sin(0.45*pi*t+ph1)+5*sin(0.55*pi*t+ph2)+noise;
rx = xcorr(x);
rw = rx.*win;

Px = abs(fft(rw,fftdatapoints));

figure(1),
plot(w/pi,10*log10(Px(1:(fftdatapoints/2)))),
xlabel('Frequency (units of pi)'),ylabel('Magnitude (dB)'),grid on
```

Autoregressive Spectrum Estimation

An AR(p) process may be represented as the output of an all–pole filter driven with unit variance white noise. Its spectrum is given by:

$$P_{xx}(\omega) = \frac{|b(0)|^2}{|1 + \sum_{k=1}^{p} a_p(k) e^{-\jmath k\omega}|^2}$$

Since $b(0)$ and $a_p(k)$ can be estimated from the data, an estimate of the power spectrum becomes

$$\hat{P}_{AR}(\omega) = \frac{|\hat{b}(0)|^2}{|1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-\jmath k\omega}|^2}$$

$\Rightarrow$ the accuracy of $\hat{P}_{AR}$ depends on how accurately the model parameters can be estimated, and whether the AR model is consisent with the data generation process.

For accuracy we need to look at methods for estimating all–pole model coefficients.

1. The Autocorrelation Method

The AR coefficients are found by solving the ACF normal equations

$$\begin{bmatrix} r_x(0) & r_x(1) & \cdots & r_x(p) \\ r_x(1) & r_x(0) & \cdots & r_x(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p) & r_x(p-1) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \sigma_w^2 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where

$$r_x(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x(n+k)x(n) \qquad k = 0, \ldots, p$$

and σ_w^2 is the driving noise variance[1]. This is sometimes referred to as the Yule–Walker method.

[1] In the Yule–Walker method $x(n)$ is assumed to be AR process, whereas in the maximum entropy method it is assumed that $x(n)$ is Gaussian.

ACF vs. MEM estimation

There are clear similarities

However, there are differences too:

- The only real difference is in the assumptions imposed on process $x(n)$
  - ME method assumes $x(n)$ is Gaussian

Observe: non–Gaussian distribution $\Leftrightarrow$ Gaussian distribution observed through a static memoryless nonlinear function.

- Since $\mathbf{R}_{xx}$ in the ACF normal equations is Toeplitz, the Levinson–Durbin algorithm may be used to solve these equations for a_p

- The ACF method effectively applies a rectangular window to the data $\Rightarrow$ the data is effectively extrapolated with zeros.

The Autocorrelation Method

- The ACF is effectively extrapolated with zeros, the ACF method generally produces low resolution estimates than the approaches that do not window the data
- Consequently, this method is generally not used for short data records
- **Spectral Line splitting:** An artifact involves the splitting of a single spectral peak into two separate and distinct peaks.

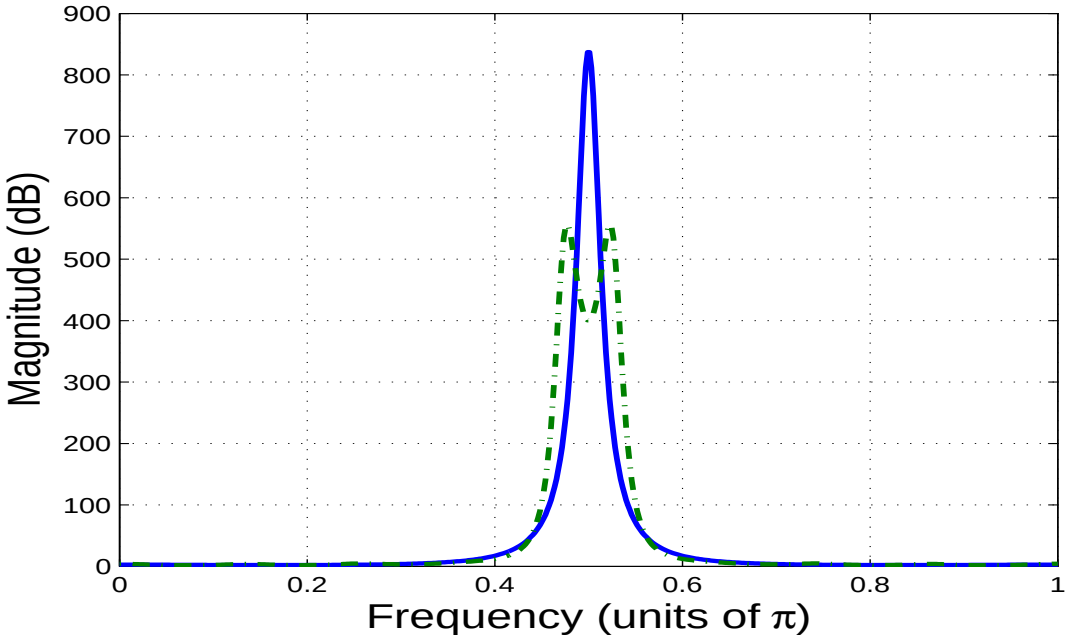
  This particularly occurs when $x(n)$ is overmodelled (p too large).
- Since the ACF estimate $r_x(k) = \frac{1}{N}\sum_{n=0}^{N-1-k} x(n+k)x(n)$ is **biased** we might use an unbiased estimate

$$r_x(k) = \frac{1}{N-k}\sum_{n=0}^{N-1-k} x(n+k)x(n)$$

  However, in this case $\mathbf{R}_{xx}$ is not guaranteed to be positive definite and teh variance in the spectrum tends to become large when $\mathbf{R}_{xx}$ is ill–conditioned.

Effects of overmodelling on spectral estimation: Spectral line splitting

Consider an AR(2) signal $x(n) = -0.9x(n-2) + w(n)$ with $w \sim \mathcal{N}(0,1)$. Consider $N = 64$ data samples, and model orders $p = 4$ and $p = 12$.



Notice that this is an AR(2) model!

Although the true spectrum has a single spectral peak at $\omega - \pi/2$ (blue), when overmodelling using $p = 12$ this peak is split into two peaks (green).

The Autocorrelation Method: Modification

Possible solution: Since the ACF estimate from the above is *biased*, we may opt to use an *unbiased* estimate of the ACF, given by

$$\hat{r}_x(k) = \frac{1}{N-k} \sum_{n=0}^{N-1-k} x(n+k)x(n) \qquad k = 0, 1, \ldots, p$$

Problems with the unbiased estimate:

- $\mathbf{R}_x$ is not guaranteed to be positive definite (singular $\Rightarrow$ cannot find an inverse)
- Large variance of the spectrum estimate
- Biased estimate generally preferred over the unbiased one

Whether or not line splitting occurs depends on the specific white noise process that generates $x(n)$.

2. The Covariance Method

Solve for $\mathbf{a} = [a_p(1), \ldots, a_p(p)]^T$

$$\begin{bmatrix} r_x(1,1) & r_x(2,1) & \cdots & r_x(p,1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(1,p) & r_{2,x}(p-1) & \cdots & r_x(p,p) \end{bmatrix} \begin{bmatrix} a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} r_x(0,1) \\ \vdots \\ r_x(0,p) \end{bmatrix}$$

where

$$r_x(k,l) = \sum_{n=p}^{N-1} x(n-l)x(n-k)$$

Similar to the ACF method, but no windowing of data.

Consequence: for short data records generally produces higher resolution spectrum estimates that the ACF method.

When $N >> p$, the effect of data window becomes small, and the difference between the two approaches becomes negligible.

3. The Modified Covariance Method

Similar to the Covariance method $\rightarrow$

no window is applied to the data

Goal: Find the AR model that minimises the sum of the squares of the forward and backward prediction errors

$$r_x(k,l) = \sum_{n=p}^{N-1} [x(n-l)x(n-k) + x(n-p+l)x(n-p+k)]$$

gives statistically stable spectrum estimates with high resolution

is not subject to spectral line splitting

$\mathbf{R}_{xx}$ is not Toeplitz

This method is also referred to as the *Forward–Backward Method* and the *Least Squares Method*.

4. The Burg Algorithm

Also finds a set of AR parameters that minimises the sum of the squares of the forward and backward prediction errors.

In order to assure that the model is stable, this minimisation is performed sequentially with respect to the reflection coefficients.

does not apply a window to the data

the estimates of AR parameters are more accurate than those obtained with an ACF method

less accurate than the modified covariance method

subject to spectral line splitting

Example: AR(4) process.

Example 4: Spectrum estimation of an AR(4) process

Consider the AR(4) process generated by the difference equation

$$x(n) = 2.7377x(n-1) - 3.7476x(n-2) + 2.6293x(n-3) - 0.9224x(n-4) + w(n)$$

where $w \sim \mathcal{N}(0, 1)$.

This process has a pair of poles at $z = 0.98e^{\pm \jmath 0.2\pi}$ and $z = 0.98e^{\pm \jmath 0.3\pi}$.

Data record length $N = 128$ and an ensemble of 50 spectrum estimates were computed using:

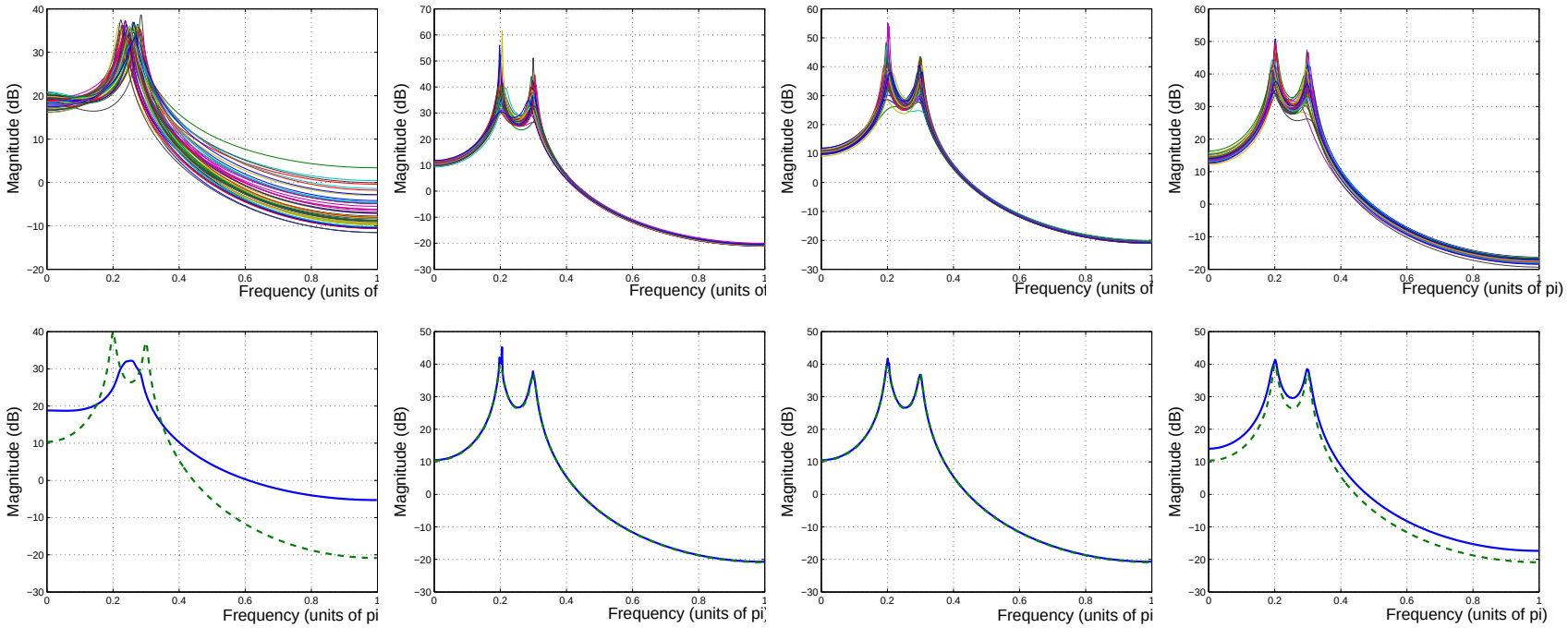
the Yule–Walker method, covariance method, modified covariance method, and Burg's method.

In the Figure onthe next page.

Observe that for this narrowband process, all of the estimates, except the Y-W method appear to be unbiased and to have a comparable variance.

Example 4: Spectrum estimation of AR(4) $\hookrightarrow$ comparison

Top panel: Overlay plot of 50 spectrum estimates



Bottom panel: Average from top, dashed line: true PSD

Methods: Yule Walker Covariance Modified Covariance Burg's method

Moving Average and ARMA spectrum estimation

In the MA spectrum estimation, the signal model is given by

$$x(n) = \sum_{k=0}^{q} b_q(k) w(n-k)$$

The spectrum of this process is given by

$$P_x(e^{\jmath\omega}) = \left| \sum_{k=0}^{q} b_q(k) e^{-\jmath k\omega} \right|^2$$

In ARMA spectrum estimation, spectrum is given as

$$P_x(e^{\jmath\omega}) = \frac{\left|\sum_{k=0}^{q} b_q(k) e^{-\jmath\omega}\right|^2}{\left|1 + \sum_{k=1}^{p} a_p(k) e^{-\jmath k\omega}\right|^2}$$

Moving average spectrum estimation

The spectrum can be estimated in one of two ways:

- Take advantage of the fact that the ACF of an MA process is finite in length

$$r_x(k) = \sum_{k=-q}^{q} b_q(l+k)b_q(k),\ k = 0, 1, \ldots, q \ \Rightarrow \hat{P}_{MA} = \sum_{k=-q}^{q} \hat{r}_x(k)e^{-\jmath k\omega}$$

- Estimate the MA parameters $\mathbf{b}$ first and then substitute these estimates into P

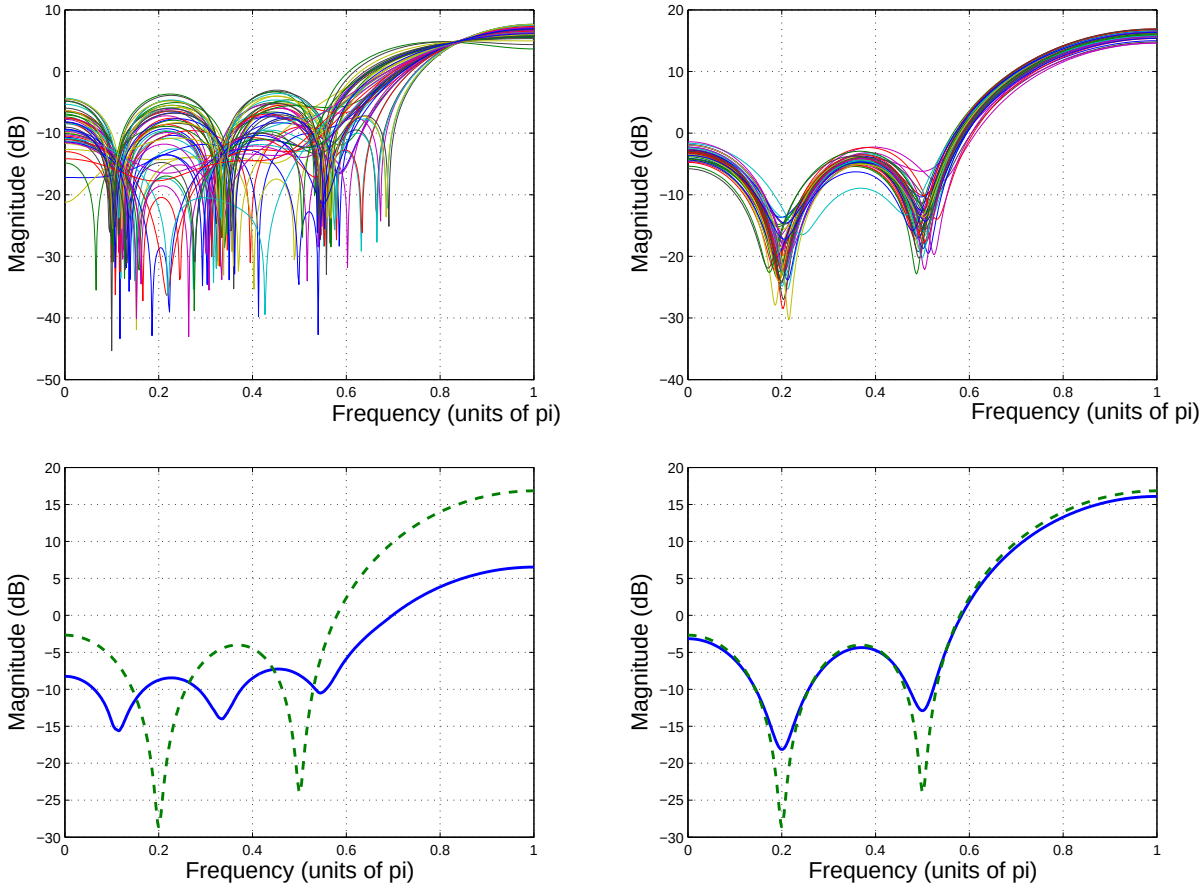
$$\hat{P}_{MA}(e^{\jmath\omega}) = \left| \sum_{k=0}^{q} \hat{b}_q(k)e^{-\jmath k\omega} \right|^2$$

Example: MA(4) model

Durbin's method may be used to estimate the parameters.

Example 5: MA Spectrum of an MA(4) process x[n]= w[n] - 1.5857 w[n-1] + 1.9208 w[n-2] - 1.5229 w[n-3] + 0.9224 w[n-4]

Top: Overlay plots Bottom: Average estimates. Dashed line: true PSD



Blackman–Tukey method **Durbin's method**

MA vs. Blackman–Tukey spectrum estimation

- $\hat{P}_{MA}(\omega) = \hat{P}_{BT}(\omega)$ when using a rectangular window
- The difference is in the assumptions behind these two estimates:
  - $\hat{P}_{MA}$ assumes $x(n)$ is an MA(q) process
  - ACF=0 for $|k| > q$
  - if an unbiased estimate of the ACF is used then $\hat{P}_{MA}(\omega)$ is unbiased
- The BT method makes no assumptions about $x(n)$ and may be applied to any type of process
- Due to the windowing of the ACF, the BT spectrum will be biased (unless $x(n)$ is an MA process)

Intermediate Summary: Methods covered so far

We have seen that there are great advantages in using optimisation to enhance the standard methods:

- The Minimum Variance Method reduces the power in sidelobes by optimizing the coefficients of the subfilters in a filterbank interpretation of the periodogram $\looparrowright$ much reduced spectral masking and bias
- The Maximum Entropy Method extrapolates the short autocorrelation sentence so as to impose least structure on the extrapolated sequence $\looparrowright$ effectively whitening the spectrum
- The AR spectrum estimation extrapolates the autocorrelation function by using a recursive expression for the ACF derived from the AR model fitted to the data ($r(k) = \sum_{i=1}^{p} a_i r(k-i)$)
- The MA spectrum is an excellent estimator when the original data obey an MA model

Notice that the MA type ACF, $r(k) = \sigma_w^2 \sum_{i=0}^{q-k} b_i b_{j+k}$, does not extrapolate

Beyond Parametric SE: Frequency Estimation and Principal Components Spectrum Estimation

By now, we know how to estimate the spectrum for various classes of processes (MA, ARMA, MV, MEM).

Question: How do we estimate the spectrum of a narrow–band signal, or how do we estimate the PSD for only one frequency in the spectrum.

This problem occurs quite frequently, for instance in the EEG research, we consider several frequency bands:

Alpha	8–13 Hz
Beta	14–20 Hz
Delta	0 – 3 Hz
Theta	4–7 Hz
Gamma	24–37 Hz

This is very useful in the design of so–called Brain Computer Interface (BCI).

Frequency estimation

Recall: last time we looked at the estimation of the power spectrum of a WSS random process which is modelled as the output of a linear shift–invariant filter, driven by white noise.

Another, very important model, is complex exponentials in white noise:

$$x(n) = \sum_{i=1}^{p} A_i e^{\jmath n \omega_i} + w(n) \qquad A_i = |A_i| e^{\jmath \Phi_i} \qquad \Phi \sim \mathcal{U}[-\pi, \pi]$$

Problem: $|A_i|$ and ω_i unknown

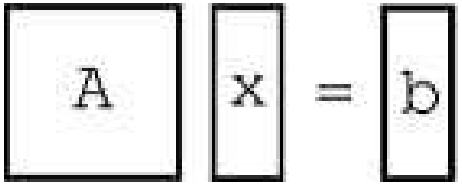
↬ The power spectrum of $x(n)$ consists of p impulses at $\omega_i, \quad i = 1, \ldots, p$ with power $2\pi|A_i|$

↬ **we are more interested in the estimation of the frequencies and amplitudes and not that much in the actual PSD**

These ω_i and $|A_i|$ are **information bearing** (sonar, radar, communications, formant freq. in speech)

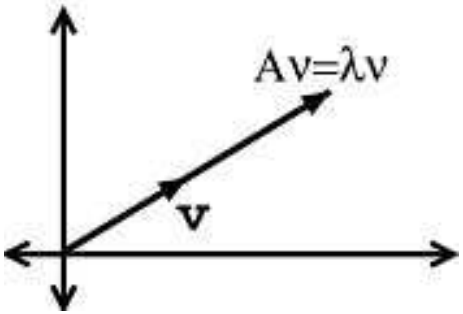
The meaning of eigenanalysis

Let $\mathbf{A}$ be an $n \times n$ matrix, where $\mathbf{A}$ is a linear operator on vectors in $\mathbb{R}^n$, such that $\mathbf{A}\,\mathbf{x} = \mathbf{b}$

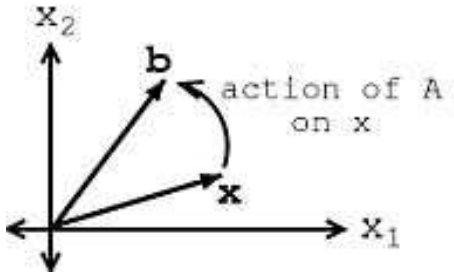


An **eigenvector** of $\mathbf{A}$ is a vector $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{A}\,\mathbf{v} = \lambda\mathbf{v}$, where λ is called the corresponding eigenvalue.

$\mathbf{A}$ only changes the length of $\mathbf{v}$, not its direction!



Equation $\mathbf{A}\,\mathbf{x} = \lambda\mathbf{x}$



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

Main point

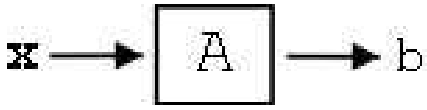
Say the eigenvectors of $\mathbf{A}$, $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ span $\mathbb{R}^n$.

This means $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ are linearly independent and

$$\mathbf{x} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n, \quad \{\alpha_1, \ldots, \alpha_n\} \in \mathbb{R}$$

All that we are doing **is rewriting $\mathbf{x}$ in terms of eigenvectors of $\mathbf{A}$**, then

$$\begin{aligned} \mathbf{A}\,\mathbf{x} &= \mathbf{A}\left(\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n\right) \\ \mathbf{A}\,\mathbf{x} &= \alpha_1 \lambda_1 \mathbf{v}_1 + \alpha_2 \lambda_2 \mathbf{v}_2 + \cdots + \alpha_n \lambda_n \mathbf{v}_n \end{aligned}$$



Therefore $\mathbf{x} = \sum_i^n \alpha_i \mathbf{v}_i$ $\Rightarrow$

By breaking up a vector $\mathbf{x}$ into a combination of eigenvectors, the calculation of $\mathbf{a}\,\mathbf{x}$ is broken into "easy to swallow" pieces

Eigenvalues

For an $n \times n$ matrix $\mathbf{A}$, its **eigenvalues** are found from the n-th order polynomial in λ defined by

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad \Rightarrow \big(\mathbf{A} - \lambda\mathbf{I}\big)\mathbf{x} = \mathbf{0} \quad \Rightarrow \quad det\left(\mathbf{A} - \lambda\mathbf{I}\right) = 0$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

The corresponding n **eigenvectors** satisfy $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ and are generally normalised to have unit norm $\| \mathbf{v} \|_2 = 1$.

For distinct eigenvalues, these eigenvectors are **linearly independent**.

A symmetric matrix is positive definite iff all its eigenvalues are positive

The **Spectral Theorem** allows for a symmetrix matrix to be written as

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

and the

$$\text{Trace}(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i$$

Any connection with signal power?

Eigendecomposition of $\mathbf{R}_{xx}$ (Page 10, Background)

Let us illustrate this on an example:

$$x(n) = A_1 e^{\jmath n\omega_1} + w(n) \qquad A_1 = |A_1| e^{\jmath \Phi_1}, \quad \Phi \sim \mathcal{U}[-\pi, \pi] \quad var(w(n)) = \sigma_w^2$$

The ACF of $x(n)$ is

$$r_x(k) = |A_1|^2 e^{\jmath k\omega_1} + \sigma_w^2 \delta(k) \qquad \text{recall properties of ACF}$$

Since $s \perp n$ (subspaces analysis – later), $\mathbf{R}_{xx}$ becomes ($\mathbf{R}_s$ **is rank one**)

$$\mathbf{R}_{xx} = \mathbf{R}_s + \mathbf{R}_n \qquad \mathbf{R}_s = \begin{bmatrix} 1 & e^{-\jmath\omega_1} & \dots & e^{-\jmath(M-1)\omega_1} \\ e^{\jmath\omega_1} & 1 & \dots & e^{-\jmath(M-2)\omega_1} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\jmath(M-1)\omega_1} & e^{-\jmath(M-2)\omega_1} & \dots & 1 \end{bmatrix}$$

and

$$\mathbf{R}_n = \sigma_w^2 \mathbf{I}$$

What is actually happening here?

For $x(n) = A_1 \exp(\jmath\omega_1 n) + w(n)$, and the frequency of interest ω_1, define

$$\mathbf{e}_1 = \left[1, e^{\jmath\omega_1}, \ldots, e^{\jmath(M-1)\omega_1}\right]^T$$

$\Rightarrow \quad \mathbf{R}_s = |A_1|^2 \mathbf{e}_1 \mathbf{e}_1^H$ **is rank one!** $\Rightarrow$ one nonzero eigenvalue $M|A_1|^2$.

$\mathbf{R}_s$ is Hermitian and the remaining eigenvectors $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_M$ are orthogonal to $\mathbf{e}_1$,

$$\mathbf{e}_1^H \mathbf{v}_i = 0 \qquad i = 2, 3, \ldots, M$$

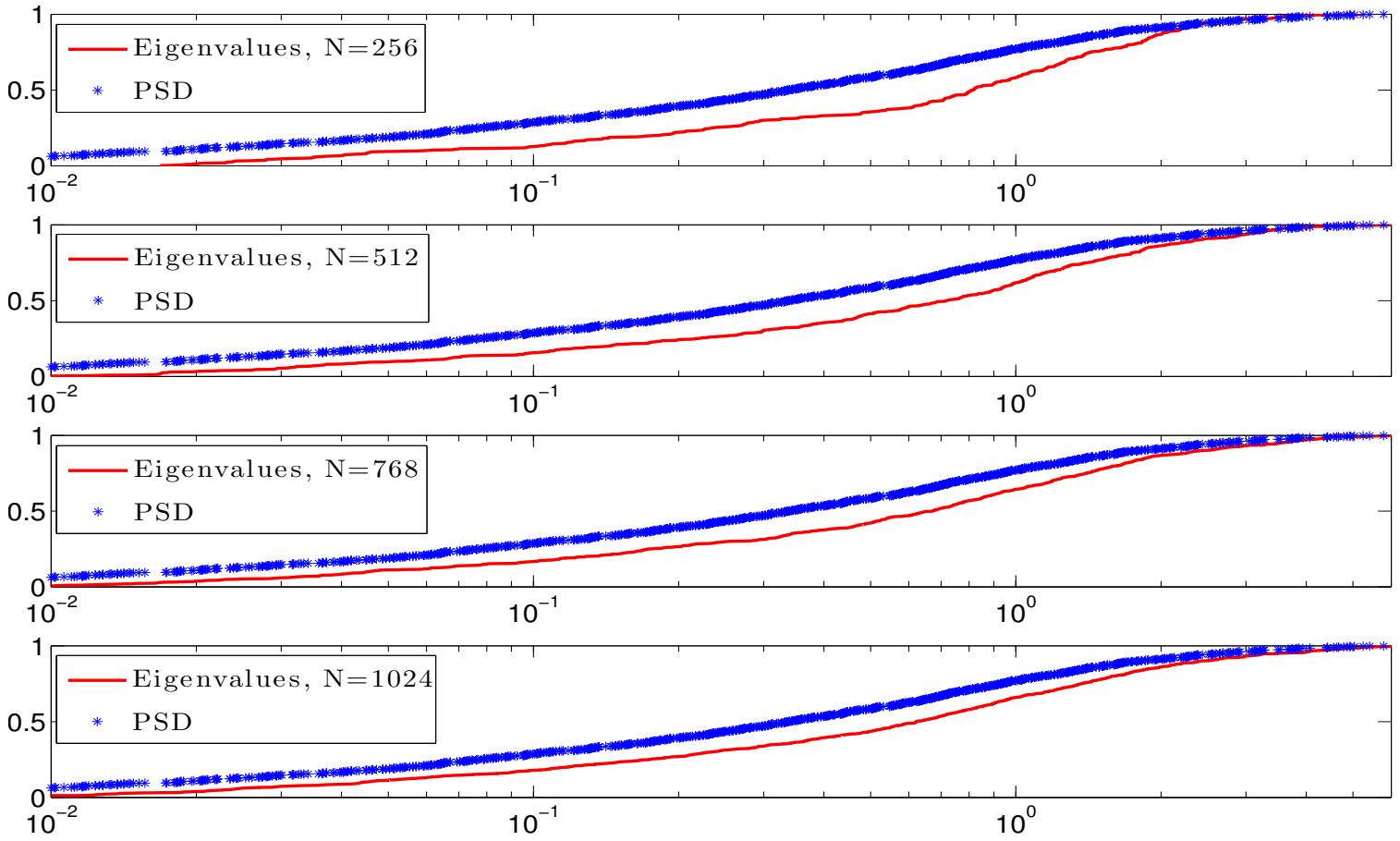
Notice also

$$\mathbf{R}_{xx}\mathbf{v}_i = \left(\mathbf{R}_s + \sigma_w^2 \mathbf{I}\right)\mathbf{v}_i = \left(\lambda_i^s \mathbf{v}_i + \sigma_w^2 \mathbf{v}_i\right) = \left(\lambda_i^s + \sigma_w^2\right)\mathbf{v}_i$$

- $\Rightarrow$ the eigenvectors of $\mathbf{R}_{xx}$ are the same as those of $\mathbf{R}_s$
- The eigenvalues of $\mathbf{R}_{xx}$ are $\lambda_i = \lambda_i^s + \sigma_w^2$
- The largest eigenvalue of $\mathbf{R}_{xx}$ is $\lambda_{max} = M|A_1|^2 + \sigma_w^2$, and the remaining $(M-1)$ eigenvalues are equal to σ_w^2

Link between the power spectrum and the eigen-analysis

Cumulative distribution of PSD and ACS matrix eigenvalues



As $N \to \infty$ the eigenvalues of the ACS matrix converge to the coefficients of the PSD

How to extract parameters of interest from $\mathbf{R}_{xx}$?

1. Perform eigendecomposition of $\mathbf{R}_{xx}$
   - the largest eigenvalue will be equal to $\lambda_{max} = M|A_1|^2 + \sigma_w^2$
   - the remaining eigenvalues will be equal to σ_w^2

2. Use the eigenvalues of $\mathbf{R}_{xx}$ to solve for the power $|A_1|^2$ and noise variance

$$\begin{aligned} \sigma_w^2 &= \lambda_{min} \\ |A_1|^2 &= \frac{1}{M}(\lambda_{max} - \lambda_{min}) \end{aligned}$$

3. Determine frequency ω_1 from the eigenvector $\mathbf{v}_{max}$ that is associated with the largest eigenvalue using, for example, the second coefficient of $\mathbf{v}_{max}$

$$\omega_i = arg\{v_{max}(1)\}$$

Example: Complex exponential in white noise

Consider

$$x(n) = A_1 e^{\jmath n\omega_1} + w(n)$$

with a correlation matrix given by

$$\mathbf{R}_{xx} = \begin{bmatrix} 3 & 2(1-\jmath) \\ 2(1+\jmath) & 3 \end{bmatrix}$$

Then

$$\lambda_{1,2} = 3 \pm 2\sqrt{2}, \qquad \mathbf{v}_{1,2} = \left[1, \pm\frac{\sqrt{2}}{2}(1+\jmath)\right]^T$$

and

$$\sigma_w^2 = \lambda_{min} = 3 - 2\sqrt{2}, \qquad |A_1|^2 = \frac{1}{M}\left(\lambda_{max} - \lambda_{min}\right) = 2\sqrt{2}$$

and finally

$$\omega_1 = arg\left\{\frac{\sqrt{2}}{2}(1+\jmath)\right\} = \frac{\pi}{4}$$

Some practical considerations

This approach works well only if we know ACF exactly

$\Rightarrow$ we can use estimates of ACF, but the eigenvalues are very sensitive to the perturbations in the matrix values.

Remedy: Why do not we use our old tricks with averaging?

Thinking loudly: if $\mathbf{v}_i$ is a noise eigenvector of $\mathbf{R}_{xx}$ ($\lambda_i = \sigma_w^2$) then

$$V_i(\omega) = \sum_{k=0}^{M-1} v_i(k) e^{-\jmath k\omega} = \mathbf{e}^H \mathbf{v}_i \underbrace{\Longrightarrow}_{Since\ s \perp n} \text{for } \omega = \omega_1 \text{ we have } V_i(\omega_1) = 0$$

$\Rightarrow$ a makeshift estimator could be (PSD large at ω_1)

$$\hat{P}_i(\omega) = \frac{1}{\left|\sum_{k=0}^{M-1} v_i(k) e^{-\jmath k\omega}\right|^2} = \frac{1}{\left|\mathbf{e}^H \mathbf{v}_i\right|^2}$$

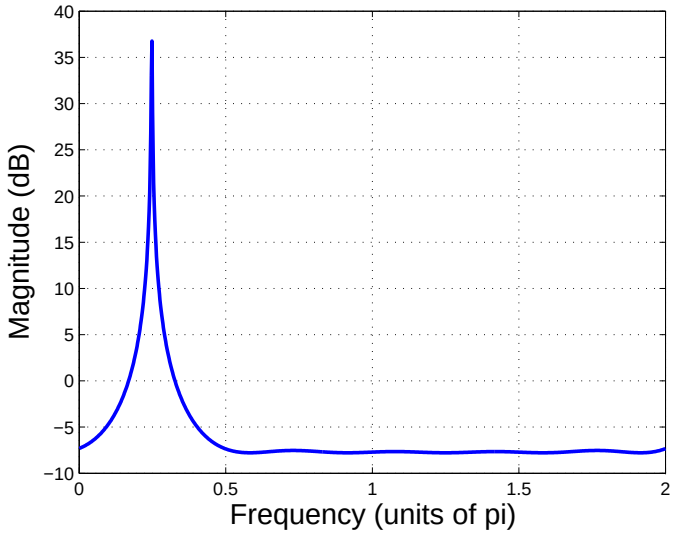
For robustness, average over all noise eigenvectors

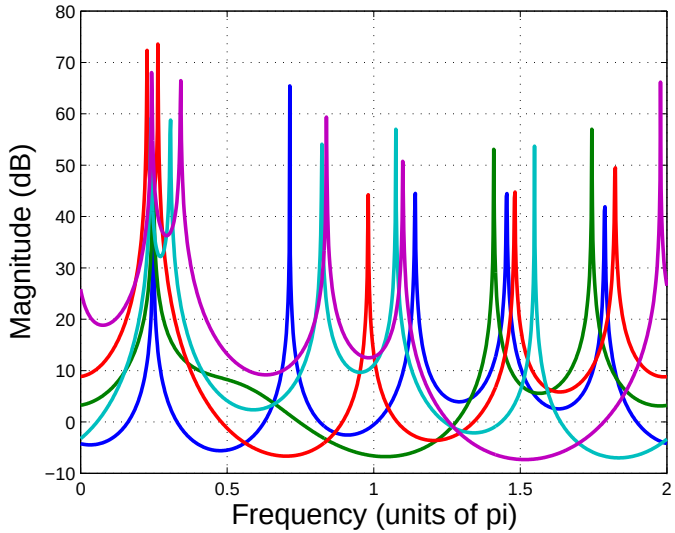
$$\hat{P}(\omega) = \frac{1}{\sum_{i=2}^{M} \alpha_i \left|\mathbf{e}^H \mathbf{v}_i\right|^2} \qquad \alpha_i \ \rightarrow \text{chosen constants}$$

Example 6: Complex exponential in WGN

$$x(n) = 4e^{\jmath n\pi/4+\Phi} + w(n), \quad \Phi \sim \mathcal{U}(0,1)$$

Left: PSD estimate with $\alpha_i = 1$ Right: overlay plot of $V_i(e^{\jmath\omega})$





Using $N = 64$, a 6×6 correlation matrix can be estimated, then we can perform eigendecomposition. The peak of PSD is at $\omega = 0.2539\pi$.

The minimim eigenvalue is $\lambda_{min} = 1$ (close to noise variance $\sigma_w^2 = 1$).

Using only a single noise eigenvector gives spurious peaks.

Example 6: Matlab code

```
clear all,clc;
N = 64;    t = 1:N;
phi = 2*pi*rand-pi;   fftdatapoints = 1024;
w = linspace(0,2*pi,fftdatapoints);
x = 4*exp(j*(pi*t/4+phi)) + randn(1,64);
rx = xcorr(x,'biased');
M = 6; % size of autoorrelation matrix
R = toeplitz(rx(N:N+M-1));
[v d] = eig(R);
d = diag(d);    [d ind] = sort(d,'descend');
v = v(:,ind);
eHv2 = abs(fft(v,fftdatapoints)).^2;
Phat = 1./sum(eHv2(:,2:end),2);
%
figure(1);    plot(w/pi, 10*log10(flipud(Phat))),
xlabel('Frequency (units of pi)'),ylabel('Magnitude (dB)'),grid on
%
figure(2);    plot(w/pi, 10*log10(flipud(1./eHv2(:,2:end)))),
xlabel('Frequency (units of pi)'),ylabel('Magnitude (dB)'),grid on
```

Two complex exponentials in white noise

Some maths:

$$
\begin{aligned}
x(n) &= A_1 e^{\jmath n\omega_1} + A_2 e^{\jmath n\omega_2} + w(n), \qquad A_i = |A_i| e^{\jmath \Phi_i} \\
r_x(k) &= P_1 e^{\jmath k\omega_1} + P_2 e^{\jmath k\omega_2} + \sigma_w^2 \delta(k), \qquad P_i = |A_i|^2 \\
\mathbf{R}_{xx} &= P_1 \mathbf{e}_1 \mathbf{e}_1^H + P_2 \mathbf{e}_2 \mathbf{e}_2^H + \sigma_w^2 \mathbf{I} \\
\mathbf{R}_s &= P_1 \mathbf{e}_1 \mathbf{e}_1^H + P_2 \mathbf{e}_2 \mathbf{e}_2^H \qquad \mathbf{R}_n = \sigma_w^2 \mathbf{I} \\
or \qquad & \mathbf{R}_{xx} = \mathbf{E}\mathbf{P}\mathbf{E}^H + \sigma_w^2 \mathbf{I} \qquad \mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2] \quad \mathbf{P} = diag\{P_1, P_2\}
\end{aligned}
$$

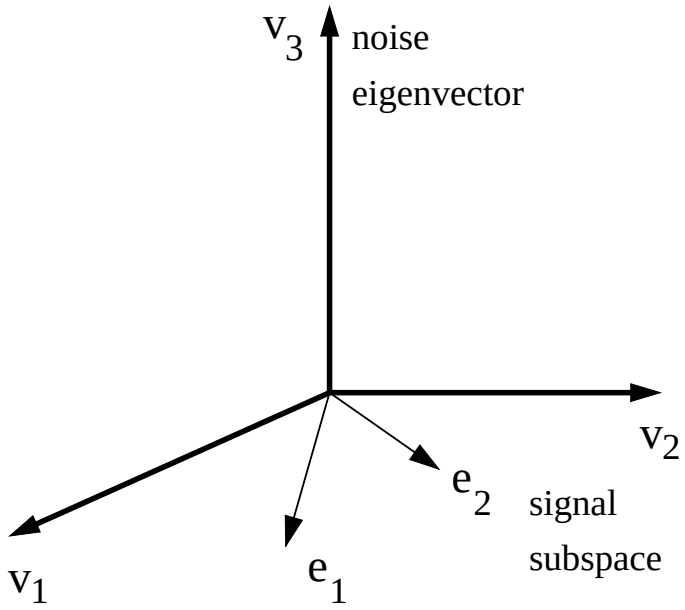
Since $\mathbf{R}_{xx} = \mathbf{R}_s + \sigma_w^2 \mathbf{I}$ we have $\lambda_i = \lambda_i^s + \sigma_w^2$

- Since $\mathbf{R}_s$ has rank two, it has only two non–zero eigenvalues and both are > 0 (positive semidefinite $\mathbf{R}_s$)
- $\Rightarrow$ The first two eigenvalues of $\mathbf{R}_{xx}$ are greater that σ_w^2 (**signal eigenvectors**) and the remaining eigenvalues are equal to σ_w^2 (**noise eigenvectors**)
- The noise eigenvectors span an (M-2)–dimensional **noise subspace**

Geometric interpretation

Due to $\mathbf{R}_{xx}$ Hermitian, the eigenvectors form an **orthonormal set**

$\Rightarrow$ **the signal and noise subspaces are orthogonal.**



Notice that the signal eigenvectors are generally not equal to $\mathbf{e}_1$ and $\mathbf{e}_2$, but $\mathbf{e}_1$ and $\mathbf{e}_2$ do lie in the signal subspace that is spanned by signal eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$

A bit more maths:

Since we have:

$$\mathbf{e}_1^H \mathbf{v}_i = 0 \quad ; \quad i = 3, 4, \ldots, M$$
$$\mathbf{e}_2^H \mathbf{v}_i = 0 \quad ; \quad i = 3, 4, \ldots, M$$

we may perform PSD estimation as:

$$\hat{P}(\omega) = \frac{1}{\sum_{i=3}^{M} \alpha_i \left| \mathbf{e}^H \mathbf{v}_i \right|^2}$$

Now for p distinct complex exponentials in white noise ($P_i = |A_i|^2$)

$$r_x(k) = \sum_{i=1}^{p} P_i e^{\jmath k \omega_i} + \sigma_w^2 \omega(k), \qquad \mathbf{R}_x = \mathbf{R}_s + \mathbf{R}_n = \sum_{i=1}^{p} P_i \mathbf{e}_i \mathbf{e}_i^H + \sigma_w^2 \mathbf{I}$$

Extension to a p–dimensional case

Notice that:

$$\mathbf{R}_{xx} = \mathbf{E}\mathbf{P}\mathbf{E}^H + \sigma_w^2\mathbf{I}, \qquad \mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_p], \quad \mathbf{P} = diag(P_1, \ldots, P_p)$$

and

$$\mathbf{P}_s = \mathbf{V}_s\mathbf{V}_s^H \quad ; \quad \mathbf{P}_n = \mathbf{V}_n\mathbf{V}_n^H$$

Since $\mathbf{e}_1, \ldots, \mathbf{e}_p$ are in the signal subspace, use the orthogonality with $\mathbf{v}_k$

$$\mathbf{e}_i^H\mathbf{v}_k = 0; \begin{cases} i = 2, \ldots, p \\ k = p+1, p+2, \ldots, M \end{cases}$$

Therefore, the frequency estimation function can be expressed as

$$\hat{P}(\omega) = \frac{1}{\sum_{i=p+1}^{M} \alpha_i \left|\mathbf{e}^H\mathbf{v}_i\right|^2}$$

Several frequency estimation algorithms are based on the above expression! The Pisarenko one uses $M = p+1$ and $\alpha_M = 1$.

Pisarenko harmonic decomposition

Assumptions:

- Signal $x(n)$ is a sum of p complex exponentials in white noise
- p is known
- $(p+1)$ values of ACF are known or estimated

Consequence:

- Since $[\mathbf{R}_{xx}]_{(p+1)\times(p+1)}$
  - the dimension of the noise subspace $= 1$
  - noise subspace spanned by the eigenvector corresponding to the minimum eigenvalue $\lambda_{min} = \sigma_w^2$.
- $\Rightarrow$ the noise eigenvector $\mathbf{v}_{min} \perp \mathbf{e}_i, \quad i = 1, 2, \ldots, p$
- $\Rightarrow V_{min}(\jmath\omega) = \sum_{k=0}^{p} v_{min}(k) e^{-\jmath k\omega} = 0 \quad at \quad \omega = \omega_i, \quad i = 1, \ldots, p$

Pisarenko frequency estimation function

From the eigenanalysis (a few slides back), we have:

$$\hat{P}_{PHD}(\omega) = \frac{1}{\left|\mathbf{e}^H \mathbf{v}_{min}\right|^2}$$

Some observations:

- $\hat{P}_{PHD}$ is a special case of the eigendecomposition approach for $M = p + 1$ and $\alpha_{p+1} = 1$
- $\hat{P}_{PHD}$ will be large at the frequencies of the complex exponentials
- The locations of peaks in $\hat{P}_{PHD}$ can be used as frequency estimates
- $\hat{P}_{PHD}$ does not contain any informatoin about the power within the signal $\Rightarrow$ often called **pseudospectrum**.

How do we detect the powers at complex exponentials?

Assume the signal subspace vectors are normalised

$$\mathbf{v}_i^H \mathbf{v}_i = 1, \quad i = 1, \ldots, p.$$

Multiply the eigenvalue–eigenevector expression by $\mathbf{v}_i^H$, to give

$$\mathbf{v}_i^H \mathbf{R}_{xx} \mathbf{v}_i = \lambda_i \mathbf{v}_i^H \mathbf{v}_i = \lambda_i, \quad i = 1, 2, \ldots, p$$

Substituting the expresion of $\mathbf{R}_{xx}$ we have

$$\mathbf{v}_i^H \mathbf{R}_{xx} \mathbf{v}_i = \mathbf{v}_i^H \left\{ \sum_{k=1}^{p} P_k \mathbf{e}_k \mathbf{e}_k^H + \sigma_w^2 \mathbf{I} \right\} \mathbf{v}_i = \lambda_i$$

with

$$\sum_{k=1}^{p} P_k \left| \mathbf{e}_k^H \mathbf{v}_i \right|^2 = \lambda_i - \sigma_w^2, \quad i = 1, 2, \ldots, p$$

Finally: PSD

Notice:

$$\left|\mathbf{e}_k^H \mathbf{v}_i\right|^2 = \left|V_i(\omega)\right|^2$$

and

$$V_i(\omega) = \sum_{l=0}^{p} v_i(l) e^{-\jmath l\omega}$$

$$\Rightarrow \qquad \sum_{k=1}^{p} P_k \left|\mathbf{e}_k^H \mathbf{v}_i\right|^2 = \sum_{k=1}^{p} P_k \left|V_i(\omega)\right|^2 = \lambda_i - \sigma_w^2, \quad i = 1, 2, \ldots, p$$

Notice: p linear equations with p unknowns P_k.

Therefore, we can solve for the powers of the complex exponentials.

In MATLAB

```
x=x(:);
R=covar(x,p+1);
[v,d]=eig(R);
sigma=min(diag(d));
index=find(diag(d)==sigma);
vmin=v(:,index);
```

```
COVAR
    P = COVAR(SYS,W) computes the output response covariance
    P = E[yy'] when the LTI model SYS is driven by WGN inputs

[V,D] = EIG(X) produces a diagonal matrix D of eigenvalues and
    a full matrix V whose columns are the corresponding
    eigenvectors sothat X*V = V*D

DIAG Diagonal matrices and diagonals of a matrix
```

Pisarenko's Method for One Sinusoid

$$x(n) = A\sin(n\omega_0 + \Phi) + w(n) \quad and \quad r_x(0) = 2.2,\ r_x(1) = 1.3,\ r_x(2) = 0.8$$

Hence $[\mathbf{R}_{xx}]_{3\times 3} = toep\{2.2, 1.3, 0.8\}$ and its eigenvalues are

$$\lambda_1 = 4.4815, \quad \lambda_2 = 1.4, \quad \lambda_3 = 0.7185$$

The white noise power is then $\sigma_w^2 = \lambda_{min} = 0.7185$, and the eigenvectors

$$\mathbf{v}_1 = \begin{bmatrix} 0.5506 \\ 0.6275 \\ 0.5506 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} -0.7071 \\ 0 \\ 0.7071 \end{bmatrix} \qquad \mathbf{v}_3 = \begin{bmatrix} -0.4437 \\ -0.7787 \\ 0.4437 \end{bmatrix}$$

To estimate the frequency ω_0, find the roots of the eigenfilter
$\mathcal{Z}(\mathbf{v}_3) = 0.4437(1 - 1.755z^{-1} + z^{-2}) \quad \Rightarrow \quad \omega_0 = 0.159\pi$

To find the signal power, look at the ACF $r_x(k) = \frac{1}{2}A^2\cos(k\omega_0) + \sigma_w^2\delta(k)$

For $k = 0$, we have $r_x(0) = \frac{1}{2}A^2 + \sigma_w^2 \quad \Rightarrow \quad A^2 = 2.963$

Pro's and con's of Pisarenko's method

- Not commonly used in practice
- Requires the number of complex exponentials to be known
- Assumes the additive noise is white
- If the additive noise is not white, the frequency estimates are biased
- Computationally consuming for higher–order problems
- However, it is possible to find the eigenvalues and eigenvector efficiently using iterative algorithms (Levinson–Durbin)

MUltiple SIgnal Classification method (MUSIC)

Assume again signal model

$$x(n) = \sum_{i=1}^{p} A_i e^{\jmath n \omega_i} + w(n), \qquad var(w(n)) = \sigma_w^2$$

and the $\mathbf{R}_{xx}$ is an $M \times M$ autocorrelation matarix, with $M > p + 1$.[2]

We desire:

- p largest eigenvalues $\lambda_1, \ldots, \lambda_p$ to correspond to the signal subspace
- the remaining $M - p$ eigenvalues to correspond to the noise subspace

Recall that we estimate the frequencies $\omega_i,\ i = 1, 2, \ldots, p$ based on the eigenvectors corresponding to noise!

Idea leading to MUSIC algorithm: look at noise eigenfilters.

[2] For $M = p + 1$, MUSIC $\equiv$ Pisarenko

Looking at the noise subspace

$$V_i(z) = \sum_{k=0}^{M-1} v_i(k) z^{-k}, \;\; i = p+1, \ldots, M \;\; \Leftrightarrow \;\; |V_i(\omega)|^2 = \frac{1}{\left| \sum_{k=0}^{M-1} v_i(k) e^{-\jmath k \omega} \right|^2}$$

Ideally p of the roots of the above polynomials (corresponding to the complex exponentials), will lie on the unit circle in the $\mathcal{Z}$ plane.

Problem:

- the remaining $(M-p-1)$ roots may lie anywhere, even close to the unit circle $\Rightarrow$ spurios peaks in the eigenspectrum
- With estimated ACF, the zeros of $V_i(z)$ which should be on the unit circle might move somewhere else
- $\Rightarrow$ when only noise eigenvectors are used, there is an ambiguity in distinguishing the desired peaks from the undesired ones.

The MUSIC algorithm

The idea is simple: **reduce the effects of spurios peaks by averaging.**

The frequency estimation function becomes

$$\hat{P}_{MU}(\omega) = \frac{1}{\sum_{i=p+1}^{M} |\mathbf{e}^H \mathbf{v}_i|^2}$$

The desired frequencies $\leftrightarrow$ locations of p largest peaks in $\hat{P}_{MU}(\omega)$.

Alternatively, look at the angles of the p roots of $\sum_{i=p+1}^{M} V_i(z) V_i^*(1/z^*)$

In MATLAB

```
x=x(:); R= covar(x,M); [v,d]=eig(R); [y,i]=sort(diag(d)); Px=0;
for j=1:M-p
   Px=Px+abs(fft(v(:,i(j)),1024));
end
```

Other eigenvector methods

- **The EigenVector (EV) method** (produces fewer spurios peaks)

$$\hat{P}_{EV}(\omega) = \frac{1}{\sum_{i=p+1}^{M} \frac{1}{\lambda_i} \left|\mathbf{e}^H \mathbf{v}_i\right|^2}$$

- **The Minimum Norm Algorithm (MN)**

$$\hat{P}_{MN}(\omega) = \frac{1}{\left|\mathbf{e}^H \mathbf{a}\right|^2} \quad s.t. \quad \mathbf{a} \in \text{noise subspace}$$

In addition:

- $\mathbf{a}$ has minimum norm (to ensure that p roots are on the unit circle)
- The first element of $\mathbf{a}$ is unity (to ensure spurious roots lie inside the unit circle)
- $\mathbf{a}$ is found from the projection onto the noise subspace

Example:

Consider $x(n)$ which consists of four complex exponentials in white noise

$$x(n) = \sum_{k=1}^{4} A_k e^{\jmath(n\omega_k + \Phi_k)} + w(n), \quad A_i = 1, i = 1, \ldots, 4$$

and

$$\omega_k \in \{0.2\pi, 0.3\pi, 0.8\pi, 1.2\pi\} \qquad \Phi \sim \mathcal{U}(0, 2\pi), \qquad \sigma_w^2 = 0.5$$

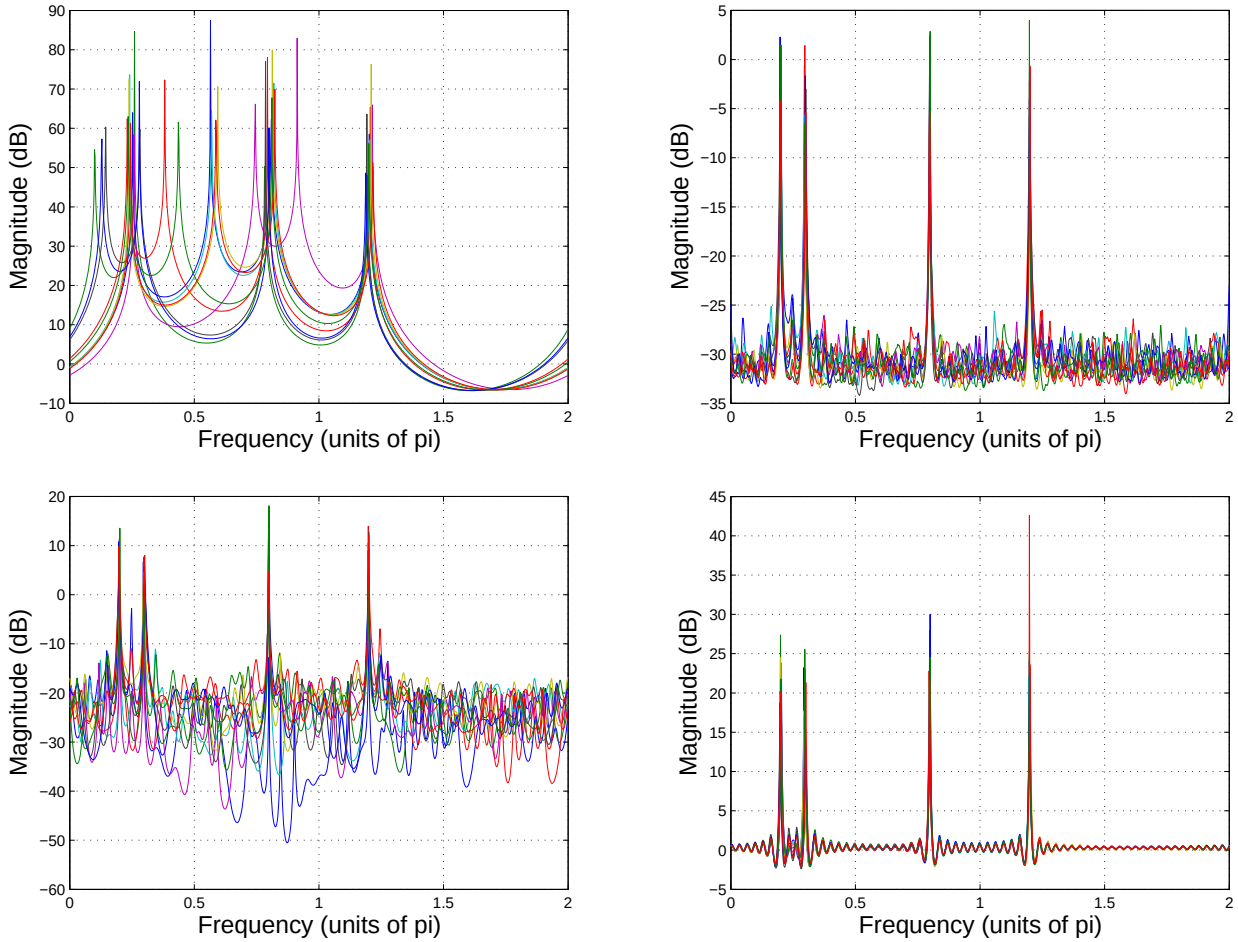
Use ten different realisation of $x(n)$ with $N = 64$ and compare Pisarenko's method, MUSIC, the eigenvector method and the minimimum norm method.

For Pisarenko's method, use 5×5 autocorrelation matrix

For other methods, use 64×64 autocorrelation matrix

PSD Est. Comparison: 4 complex sinewaves in WGN

Clockwise from top left: Pisarenko, MUSIC, minimum norm, eigenvector



Except for Pisarenko's method, all other estimates are correct!

Principal components spectrum estimation

The previous methods are often referred to as **noise subspace methods**.

Here, we look at methods that use vectors that **lie in the signal subspace.**

signal subspace methods are based on **principal component analysis**

Mathematical background:

Let $\mathbf{R}_{xx}$ be an $M \times M$ ACM of a signal consisting of p complex exponentials in white noise. The **eigendecomposition** of $\mathbf{R}_{xx}$ is

$$\mathbf{R}_{xx} = \sum_{i=1}^{M} \lambda_i \mathbf{v}_i \mathbf{v}_i^H = \underbrace{\sum_{i=1}^{p} \lambda_i \mathbf{v}_i \mathbf{v}_i^H}_{signal} + \underbrace{\sum_{i=p+1}^{M} \lambda_i \mathbf{v}_i \mathbf{v}_i^H}_{noise}$$

The eigenvalues are arranged in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$

Question: Why do not we look at a reduced rank ACM $\mathbf{R}_s$?

Let us do it!

$$\hat{R}_s = \sum_{i=1}^{p} \lambda_i \mathbf{v}_i \mathbf{v}_i^H$$

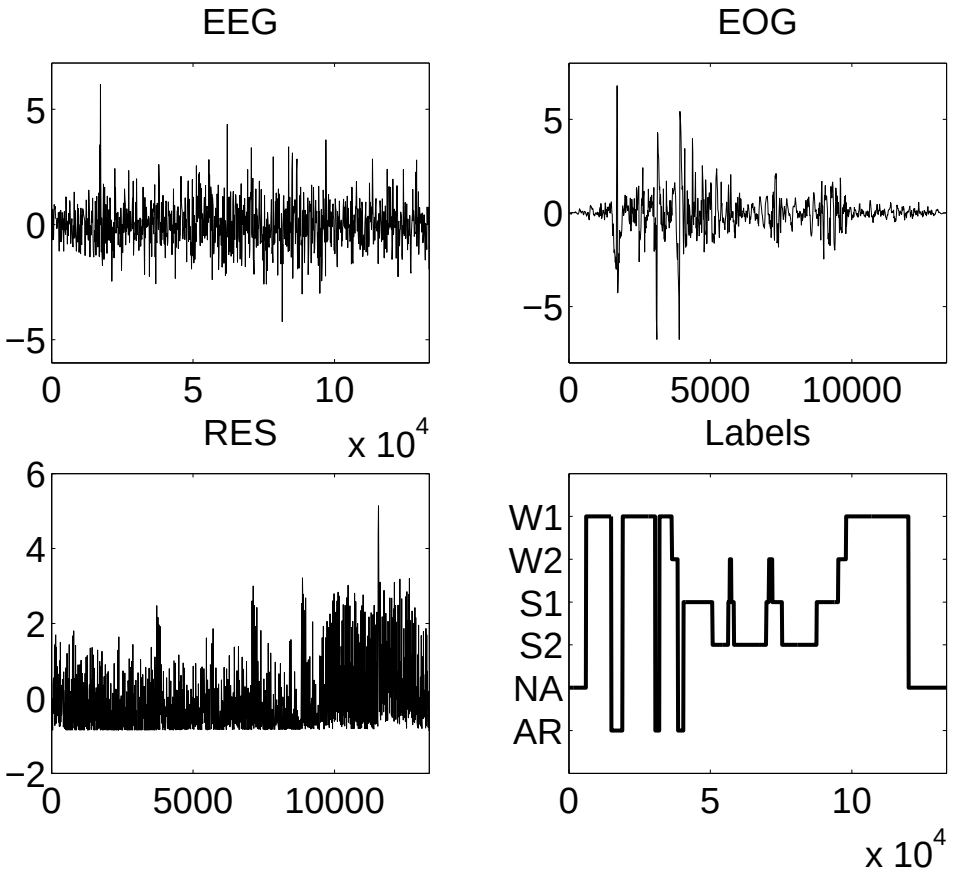
- we effectively filter out a portion of the noise
- this way, the estimate of the spectral component due to signal is enhanced
- in terms of maths, we impose a **rank p constraint** on $\mathbf{R}_{xx}$
- We can use principal component (PCA) analysis in conjunction with
  - Blackman–Tukey frequency estimation
  - minimum variance method
  - maximum entropy method
  - autoregressive spectrum estimation

Some Observations

1. **The white noise assumption is not too restrictive**, as if the noise is correlated and the correlation time is less than half the period of the fastest sinewave of interest, then a sampling period larger than the noise correlation time results in white noise.
2. For line spectra methods, the PSD consists of a **noise floor** of a constant level σ_w^2 and p vertical lines at frequencies $\omega_i, i = 1, \ldots, p$. We may use some thresholding method.
3. Line methods for frequency estimation are **high resolution** (or even super-resolutions) due to their ability to resolve spectral lines by less than $1/N$ cycles per sampling interval.
4. Periodogram based methods **do not assume any knowledge** about the data, line spectra methods do.
5. Frequency estimation of line spectra is a **linear regression** problem:

$$x(n) = \sum_{i=1}^{p} A_i e^{\jmath \omega_i n} + w(n)$$ and can be estimated using e.g. least squares

Opportunities: Mode Segmentation in Sleep Analysis



Time series plots of the EEG, EOG, RES-signals and the manual segmentation, performed by a medical expert.

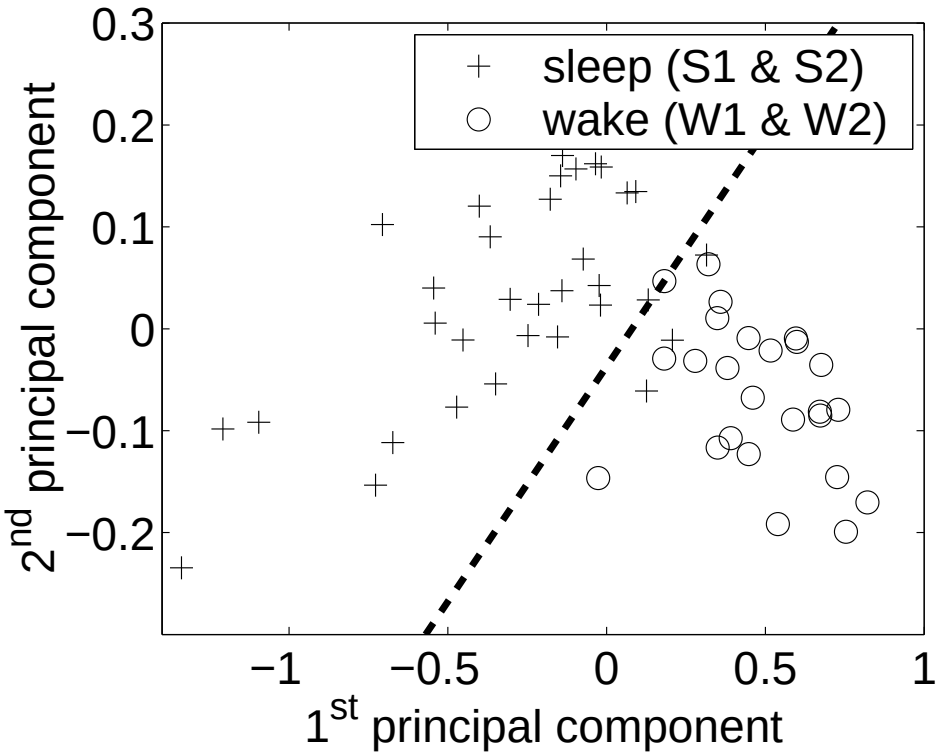
EEG = ElectroEncephaloGram

EOG= ElectroOculoGram

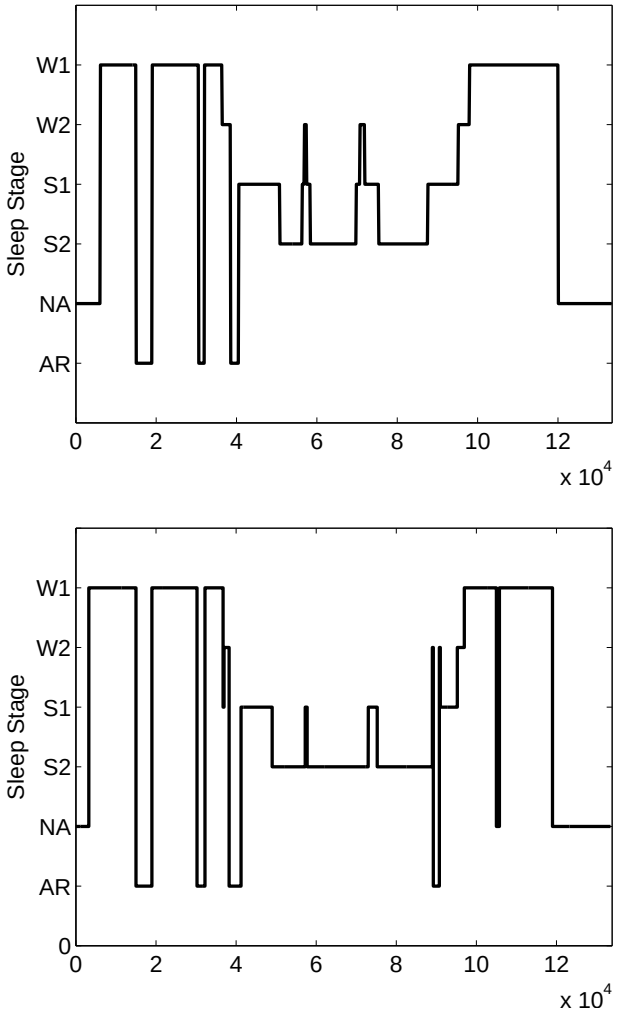
RES = Respiration signal

A medic makes assessment of sleep stages by visually inspecting these signals

Opportunities: Mode Segmentation – PCA Analysis



Correct labelling (medic)



PCA based labelling

Discrimination ability of eigenanalysis:

- Scatter plot for the 2nd order eigenspace
- Clear separation between 'sleep' and 'awake'
- More detailed separation - a larger eigenspace

Lecture Summary

- Periodogram
  - derived from the DFT, not consistent
  - to improve statistical properties: windowing, averaging, smoothing
  - limited performance for short data records, and low resolution
  - limitation for closely spaced narrowband processes
  - Advantage: no assumptions or constraints $\Rightarrow$ applicable to any class of processes

- Minimum Variance Method
  - a data–adaptive modification of periodogram
  - consists of a filter bank of bandpass filters
  - power spectrum estimated by dividing the power estimate with the bandwidth of the filter
  - higher resolution than the Blackman–Tukey methods

- Maximum Entropy method
  - derived in order to overcome the limitation of a windowed ACF
  - spectrum estimated using a maximum entropy extrapolation of ACF
  - equivalent to an all–pole model

Lecture Summary

- Parametric methods
  - need to select an appropriate model for the process
  - use of a priori knowledge about process generation
  - boils down to estimating model parameters from the data
  - estimate power spectrum by incorporating the estimated model parameters into the expression for true spectrum
  - works well if the process is consistent with the model used

- Techniques that assume a harmonic model of the process
  - goal – estimate frequencies and possibly determine the powers
  - one approach to take advantage of the orthogonality of the signal and noise subspaces
  - another approach – use PCA to produce a reduce–rank approximation and then combine with other methods

Some background: Second order data modelling

We start from: $y(n) = a_1(n)x(n-1) + a_2(n)x(n-2) + \cdots a_p(n)x(n-p)$

Teaching signal: $d(n)$, **Output error:** $e(n) = d(n) - y(n)$

Fixed coeff. $\mathbf{a}$ & $x(n) = y(n)$

Autoregressive modelling

$$\begin{aligned} r_{xx}(1) &= a_1 r_{xx}(0) + \cdots + a_p r_{xx}(p-1) \\ r_{xx}(2) &= a_1 r_{xx}(1) + \cdots + a_p r_{xx}(p-2) \\ \vdots &= \vdots \\ r_{xx}(p) &= a_1 r_{xx}(p-1) + \cdots + a_p r_{xx}(0) \\ \cdots & \quad \cdots \\ \mathbf{r}_{xx} &= \mathbf{R}_{xx}\mathbf{a} \end{aligned}$$

Solution: $\mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}$

Yule–Walker equation

Fixed optimal coeff. $\mathbf{w}_o = \mathbf{a}_{opt}$

$$J = E\{\frac{1}{2}e^2(n)\} = \sigma_d^2 - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}$$

is quadratic in $\mathbf{w}$ and for a full rank $\mathbf{R}$, it has **one unique minimum.**

Now:

$$\frac{\partial J}{\partial \mathbf{w}} = -\mathbf{p} + \mathbf{R}\cdot\mathbf{w} = \mathbf{0}$$

Solution: $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$

Wiener–Hopf equation

Notes:

Notes: