Stationary and Non-stationary Spectral Analysis

Spring 2014

Andreas Jakobsson & Maria Sandsten

Course outline

• Basic definitions, AR/MA/ARMA, Line spectra
• Parametric techniques, Subspace techniques
• Periodogram, multi-window techniques
• Data-adaptive techniques
• Spatial spectral analysis
• Spectrogram, Wigner-Ville distribution, Wigner spectrum, Cross-terms.
• Ambiguity function and kernels. Concentration measures. Quadratic distributions, Cohen’s class.
• Project

Administration

Literature:
• Maria Sandsten, lecture notes
• Research papers

Examination:
• Four hand-ins (theoretical and computer exercises) [10 marks each].
  Deadline: 31/1, 7/2, 26/2, 7/3. The deadline is strict!
• Project presentation and report [30 marks].

Grades:
• Undergrad. students: U: 0-35, G: 36-50, VG: 51-70
• Grad. students: U: 0-45, G: 46-70

Spectral analysis?

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 is . . . an Art”

Petre Stoica
Are signals stationary?

Basic definitions
Stoica & Moses, Chapter 1

- Fundamentals.
- First definition of PSD.
- Second definition of PSD.
- Properties

Fundamentals
Let \( \{y(t)\}_{t=-\infty}^{\infty} \) be a discrete-time deterministic data sequence. If
\[
\sum_{t=-\infty}^{\infty} |y(t)|^2 < \infty,
\]
then both the Discrete-Time Fourier Transform (DFT)
\[
Y(\omega) = \sum_{t=-\infty}^{\infty} y(t)e^{-j\omega t}, \quad \omega \in [-\pi, \pi]
\]
and the inverse DFT
\[
y(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\omega)e^{j\omega t} d\omega
\]
exists.

Fundamentals
Define \( S(\omega) \triangleq |Y(\omega)|^2 \). Then
\[
\sum_{t=-\infty}^{\infty} |y(t)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega,
\]
which is Parseval’s equality. Thus, \( S(\omega) \) is the distribution of energy as a function of frequency, i.e., \( S(\omega) \) is the energy spectral density.
A first definition

Instead, let \( \{y(t)\}_{t=-\infty}^{\infty} \) be a random sequence. Then, the sequence is not absolute square summable. However, the average power is usually finite, such that \( E \{ |y(t)|^2 \} < \infty \).

The power spectral density (PSD) is defined as the DFT of the ACS,

\[
\phi(\omega) \triangleq \sum_{k=-\infty}^{\infty} r(k)e^{-j\omega k}
\]

Thus,

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\omega) e^{j\omega \tau} d\omega = \sum_{p=-\infty}^{\infty} r(p) \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{j\omega(k-p)} d\omega \right] = r(k)
\]

yielding that

\[
r(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi(\omega) d\omega
\]

As \( r(0) = E \{ |y(t)|^2 \} \) measures the (average) power, \( \phi(\omega) \) is the distribution of signal power over frequency, i.e., the PSD.

Properties

P1: \( \phi(\omega) = \phi(\omega + 2\pi f) \) for all \( \omega \). Let \( \omega = 2\pi f \), where \( f \in [-\frac{1}{2}, \frac{1}{2}] \).

P2: \( \phi(\omega) \) is real-valued and non-negative, i.e., \( \phi(\omega) \geq 0, \forall \omega \).

P3: If \( y(t) \) is real-valued, \( \phi(\omega) = \phi(-\omega) \). Otherwise, \( \phi(\omega) \neq \phi(-\omega) \).

P4: If \( y(t) = e^{j\omega t} x(t) \), then \( \phi_\omega(\omega) = \phi_\omega (\omega - \omega_0) \).

P5: If \( y(t) = \sum_{n=-\infty}^{\infty} b_n x(t-n) \), then \( \phi_\omega(\omega) = |H(\omega)|^2 \phi_\omega (\omega) \).

P6: For a zero-mean stationary random process,

\[
\min_\omega \phi(\omega) \leq \lambda_k \leq \max_\omega \phi(\omega) \quad \text{for } k = 1, \ldots, m
\]

Problem: Find an estimate of \( \phi(\omega), \omega \in [-\pi, \pi] \) given \( \{y(t)\}_{t=-N}^{N} \).

A second definition

Alternatively,

\[
\phi(\omega) \triangleq \lim_{N \to \infty} E \left\{ \frac{1}{N} \left| \sum_{t=1}^{N} y(t)e^{-j\omega t} \right|^2 \right\} = \lim_{N \to \infty} E \left\{ \frac{1}{N} \left| Y_N (\omega) \right|^2 \right\}
\]

where

\[
Y_N (\omega) = \sum_{t=1}^{N} y(t)e^{-j\omega t}
\]

Under the assumption that \( r(k) \) decays sufficiently rapidly, i.e.,

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

the two definitions are equivalent.

Parametric methods for rational spectra

Stoica & Moses, Chapter 3

- Introduction
- AR signals
  - Yule-Walker
  - Least-squares
  - The Burg algorithm
  - Modified covariance method
  - A combined approach
- MA signals
- ARMA signals
  - Modified Yule-Walker
  - Two-stage least-squares
Parametric vs non-parametric methods

With accurate model knowledge, a parametric method will generally be preferable. A non-parametric method makes no assumption on the model and are therefore more robust, but often less precise, than the parametric methods.

Recently, there has also been notable work on semi-parametric methods.

Parametric methods for rational spectra

Consider a zero-mean white sequence, $u(n)$, filtered by a linear filter,

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

where

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

$$
B(\omega) = 1 + b_1 e^{-i\omega} + \cdots + b_q e^{-iq\omega}
$$

Remarks:

(i) According to Weierstrass theorem, $\phi(\omega)$ can approximate arbitrarily well any continuous PSD, provided $p$ and $q$ are large enough.

(ii) The choice of $p$ and $q$ is not simple.

(iii) Some PSD are not continuous.

It is often convenient to work in the $Z$-domain, i.e., $z = e^{i\omega}$, to describe $\phi(\omega)$ using polynomials. Then,

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

We speak of AR ($q = 0$), MA ($p = 0$) and ARMA models.

Autoregressive signals

Often, one is interested in working with autoregressive (AR) signals,

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

as

(i) the AR spectrum is “peaky” which well models many signals.

(ii) very efficient and accurate methods exist.

In the following, we will examine the four main techniques to estimate the AR coefficients, i.e., the Yule-Walker, least-squares, Burg and modified covariance methods.
The Yule-Walker method

Consider an AR signal,

\[ u(n) = \sum_{k=0}^{p} a_k x(n-k) \]


where

\[ \Psi = \begin{bmatrix} 1 & \theta^T \end{bmatrix} = \begin{bmatrix} 1 & a_1 & \cdots & a_p \end{bmatrix}^T \]

Multiply with \( x'(n-k) \) and take \( E \{ \cdot \} \),

\[ \begin{bmatrix} r(k) & r(k-1) & \cdots & r(k-p) \end{bmatrix} \Psi = \sigma_n \delta(k) \]

In matrix form,

\[ \begin{bmatrix} r(0) & r(-1) & \cdots & r(-p) \\ r(1) & r(0) & \cdots & r(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix} = \sigma_n^2 \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \]

The least-squares method

Recalling that \( x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n) \), we define the estimation error, \( e(t) \), as

\[ e(t) = x(t) - \hat{x}(t) = x(t) + \sum_{k=1}^{p} a_k x(n-k), \]

and determine \( \theta \) such that the mean squared error is minimized,

\[ \hat{\theta}_{LS} = \arg \min_{\theta} f(\theta) = \arg \min_{\theta} \sum_{t=1}^{N_k} |e(t)|^2 \]

for some appropriate \( N_1 \) and \( N_2 \). Rewrite \( f(\theta) \) as

\[ f(\theta) = \left\| \begin{bmatrix} x(N_1) \\ \vdots \\ x(N_2) \end{bmatrix} + \begin{bmatrix} x(N_1-1) & \cdots & x(N_1-p) \\ \vdots & \ddots & \vdots \\ x(N_2-1) & \cdots & x(N_2-p) \end{bmatrix} \theta \right\|_F^2 = \| Q + X \theta \|_F^2 \]

where \( \| Q \|_F = \text{tr} (Q^* Q) \) denotes the Frobenius norm.

The Yule-Walker method

Equivalently, the so-called Yule-Walker (YW) equations, can be expressed as

\[ \mathbf{R}_p \theta = -\mathbf{r}_p, \]

where

\[ \mathbf{R}_p = \begin{bmatrix} r(0) & \cdots & r(-p+1) \\ \vdots & \ddots & \vdots \\ r(p-1) & \cdots & r(0) \end{bmatrix}, \quad \theta = \begin{bmatrix} r(1) \\ \vdots \\ r(p) \end{bmatrix}, \quad \mathbf{r}_p = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \]

Thus, one may replace \( r(k) \) with \( \hat{r}(k) \), solve \( \mathbf{R}_p \hat{\theta} = -\mathbf{r}_p \) as \( \hat{\theta}_{YW} = -\mathbf{R}_p^{-1} \mathbf{r}_p \), to obtain the Yule-Walker PSD estimate

\[ \hat{\phi}_{YW}(\omega) = \frac{\hat{\sigma}_n^2}{\mathbf{A}(\omega)} \]

Remarks

(i) The Yule-Walker estimate, \( \hat{\theta}_{YW} \), is stable.
(ii) \( \mathbf{R}_p \) is Toeplitz, Hermitian and positive semidefinite.
(iii) Different approaches to estimate \( \mathbf{R}_p \) yields different \( \hat{\phi}_{YW}(\omega) \).

The least-squares method

Thus,

\[ \hat{\theta}_{LS} = -(X^* X)^{-1} X^* x \]

Depending on the choice of \( N_1 \) and \( N_2 \), we obtain different estimates. The most common ones are

(i) The autocorrelation method, with \( N_1 = 1 \) and \( N_2 = N + p \).
(ii) The covariance method, with \( N_1 = p + 1 \) and \( N_2 = N \).

Remarks

(i) The autocorrelation method is equivalent to the Yule-Walker method.
(ii) The covariance method use unbiased estimates of \( \mathbf{R}_p \) and \( \mathbf{r}_p \) but may be unstable.
(iii) The covariance method yields more accurate spectral estimates.
(iv) For the Yule-Walker estimate, \( X^* X \) is Toeplitz, for the covariance method, it is not.
The Levinson-Durbin algorithm

By exploiting the Toeplitz structure of $R_p$, one can efficiently compute the Yule-Walker estimate recursively. Note that as $r(k) = r^{*}(-k)$,

$$ R_{n+1} = R_{n+1}^* = \begin{bmatrix} r(0) & \cdots & r(-n) \\ \vdots & \ddots & \vdots \\ r(n) & \cdots & r(0) \end{bmatrix} $$

Then, letting $\rho_n$ denote either $r(n)$ or $\bar{r}(n)$,

$$ R_{n+2} = \begin{bmatrix} R_{n+1} & \rho_{n+1}^* \\ \rho_{n+1} & \rho_0 \end{bmatrix} $$

where $\rho_n = [\rho_n \ldots \rho_1]$.

21

The Levinson-Durbin algorithm

Therefore,

$$ k_n = -\alpha_n/\sigma_n^2 $$
$$ \theta_{n+1} = \begin{bmatrix} \theta_n \\ 0 \end{bmatrix} + k_n \begin{bmatrix} \theta_n \\ 1 \end{bmatrix} $$
$$ \sigma_n^2 = \sigma_n^2 + k_n \alpha_n^* = \sigma_n^2 (1 - |k_n|^2) $$

This is the Levinson-Durbin algorithm (LDA), which computes the AR coefficients in $O(p^2)$ operations.

22

The Levinson-Durbin algorithm

Thus,

$$ R_{n+2} \begin{bmatrix} 1 \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \end{bmatrix} $$

where $\alpha_n = \rho_{n+1} + \bar{r}_n \theta_n$. But as conjugating and transposing

$$ R_{n+2} \begin{bmatrix} 1 \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \end{bmatrix} $$

yields

$$ R_{n+2} \begin{bmatrix} 0 \\ \bar{\theta}_n \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \end{bmatrix} $$

where $\bar{x}$ denotes the reversing of $x$, we can write

$$ R_{n+2} \begin{bmatrix} 1 \\ \theta_n \end{bmatrix} + k_n \begin{bmatrix} 0 \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sigma_n^2 + k_n \alpha_n^* \\ 0 \end{bmatrix} $$
$$ 0 + k_n \alpha_n^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} $$

23

The Levinson-Durbin algorithm

Therefore,

$$ k_n = -\alpha_n/\sigma_n^2 $$
$$ \theta_{n+1} = \begin{bmatrix} \theta_n \\ 0 \end{bmatrix} + k_n \begin{bmatrix} \theta_n \\ 1 \end{bmatrix} $$
$$ \sigma_n^2 = \sigma_n^2 + k_n \alpha_n^* = \sigma_n^2 (1 - |k_n|^2) $$

This is the Levinson-Durbin algorithm (LDA), which computes the AR coefficients in $O(p^2)$ operations.

The Schur recursion A central step of the LDA is the computation of the reflection coefficients,

$$ k_n = -\alpha_n/\sigma_n^2 $$

which are used to compute the AR coefficients. As an alternative, the Schur recursion instead just update the reflection coefficients, possibly computing the AR as a final step if they are needed. The Schur recursion

(i) is numerically robust compared to the LDA,
(ii) can be efficiently implemented on parallel processors, and
(iii) is only marginally costlier (if the ARs are needed).

24
The Burg algorithm

A popular alternative to the Yule-Walker or covariance methods, is the Burg algorithm which computes the AR coefficients from data without first computing \( r(k) \). It exploits the fact that as

\[
\theta_{n+1} = \left[ \begin{array}{c} \theta_n \\ 0 \end{array} \right] + k_{n+1} \left[ \begin{array}{c} \hat{\theta}_n \\ 1 \end{array} \right] \tag{1}
\]

we only need to know \( \theta_n \) and \( k_{n+1} \) to compute \( \theta_{n+1} \). How can we compute the reflection coefficients, \( k_n \), from the data? For this purpose, we form the forward and backward prediction errors for the \( p \)-th order model as \( (t = p, \ldots , N) \)

\[
\hat{e}_{f,p}(t) = x(t) + \sum_{k=1}^{p} \hat{a}_{f,p,k} x(t-k), \\
\hat{e}_{b,p}(t) = x(t-p) + \sum_{k=1}^{p} \hat{a}_{b,p,k} x(t-p+k)
\]

Using (1), these can be rewritten as

\[
\hat{e}_{f,p}(t) = \hat{e}_{f,p-1}(t) + k_p \hat{e}_{b,p-1}(t-1) \\
\hat{e}_{b,p}(t) = \hat{e}_{b,p-1}(t-1) + k_p \hat{e}_{f,p-1}(t)
\]

Remarks

(i) The Burg algorithm guarantees a stable model, and can be computed efficiently.

(ii) The obtained spectrum, \( \hat{\phi}_{\text{burg}}(\omega) \), usually behaves somewhere in between the LS and the YW methods.

(iii) Peak locations in \( \hat{\phi}_{\text{burg}}(\omega) \) is highly dependent on initial phase.

(iv) \( \hat{\phi}_{\text{burg}}(\omega) \) may suffer line-splitting.

(v) Frequency bias for estimates of sinusoids in noise.

(vi) The method is suboptimal as it decouples the \( n \)-dimensional minimization problem into \( n \) one-dimensional minimization problems.

The Burg algorithm

Burg’s algorithm finds \( k_p \) such that

\[
k_p = \arg \min_{k_p} \left\{ \sum_{t=p+1}^{N} |\hat{e}_{f,p}(t)|^2 + \sum_{t=p+1}^{N} |\hat{e}_{b,p}(t)|^2 \right\}
\]

\[
= -2 \sum_{t=p+1}^{N} \hat{e}_{f,p-1}(t) \hat{e}_{b,p-1}(t-1) + 2 \sum_{t=p+1}^{N} |\hat{e}_{f,p-1}(t)|^2 + |\hat{e}_{b,p-1}(t-1)|^2
\]

The solution can be seen as a lattice filter.

The modified covariance method

Instead, estimate \( \theta_p \) by minimizing over all dimensions at once, i.e.,

\[
\hat{\theta}_p = \arg \min_{\theta_p} \left\{ \sum_{t=p+1}^{N} |\hat{e}_{f,p}(t)|^2 + |\hat{e}_{b,p}(t)|^2 \right\}
\]

\[
= \arg \min_{\theta_p} \left\{ \hat{e}_{f,p}^T \hat{e}_{f,p} + \hat{e}_{b,p}^T \hat{e}_{b,p} \right\}
\]

Recall that, for a given \( t \),

\[
\hat{e}_{f,p}(t) = x(t) - \hat{x}(t) = x(t) + \sum_{k=0}^{p} a_k x(t-k) = \sum_{k=0}^{p} a_k x(t-k)
\]

\[
= \begin{bmatrix} x(t) & x(t-1) & \ldots & x(t-p) \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix}
\]

Thus, for \( t = p+1, \ldots , N \),

\[
\hat{e}_{f,p} = \begin{bmatrix} x(p+1) & \ldots & x(1) \\ \vdots & \ddots & \vdots \\ x(N) & \ldots & x(N-p) \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix}
\]

\[
= X \begin{bmatrix} 1 \\ \theta \end{bmatrix}
\]
In matrix form, multiply with:

Consider an AR signal, 

\[
\mathbf{r} = \mathbf{X} \begin{bmatrix} \theta^b \\theta \end{bmatrix}
\]

where \( \mathbf{J} \) is the exchange matrix, and \( \theta \) and \( \theta^b \) are related as

\[
\theta^b = \theta + 3\theta^*.
\]

Combining the equations yields

\[
\begin{bmatrix} e_{f,p} \\ e_{p,\hat{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \mathbf{J} \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix}
\]

The modified covariance method (MCM) minimize

\[
E^{fb} = \hat{e}_{f,p}^T \hat{e}_{f,p} + \hat{e}_{p,\hat{b}}^T \hat{e}_{p,\hat{b}}
\]

or equivalently,

\[
\begin{bmatrix} \mathbf{X}^T \mathbf{J} & \mathbf{X} \mathbf{J} \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix} = \begin{bmatrix} E^{fb} \\ R_{ef} \end{bmatrix}
\]

\[
= \begin{bmatrix} \mathbf{X}^T \mathbf{X} + \mathbf{J}^T \mathbf{X}^T \mathbf{J} \end{bmatrix} \begin{bmatrix} 1 \\ \theta \end{bmatrix}
\]

The modified covariance method

**Remarks**

(i) The obtained spectrum, \( \hat{\phi}_{MCM}(\omega) \), has excellent performance.

(ii) There exists efficient algorithms to compute \( \hat{\phi}_{MCM}(\omega) \).

(iii) MCM may produce unstable models (although very rare in practice).

(iv) The peak locations in \( \hat{\phi}_{MCM}(\omega) \) depends slightly on initial phase.

(v) \( \hat{\phi}_{MCM}(\omega) \) does not suffer from line-splitting.

(vi) Only minor frequency bias for estimates of sinusoids in noise.

A combined approach

Often, one can obtain very accurate estimates by combining parametric pre-whitening with non-parametric PSD estimation.

(i) Estimate the AR coefficients using Burg or MCM.

(ii) Compute the residual,

\[
e(t) = x(t) - \hat{x}(t) = x(t) + \sum_{k=1}^\infty a_k x(n-k),
\]

(iii) Compute \( \hat{\phi}_e(\omega) \) using a non-parametric PSD method.

(iv) Compute \( \hat{\phi}_e(\omega) \) by postcoloring,

\[
\hat{\phi}_e(\omega) = \frac{\hat{\phi}_e(\omega)}{|\hat{A}(\omega)|^2}
\]

Moving average signals

The moving average (MA) signals are good at modeling spectra characterized by broad peaks and sharp nulls. However, there is a limited interest in MA modeling as

(i) Such spectra are encountered less frequently in practice.

(ii) The MA parameter estimation problem is nonlinear, and is much harder than the AR parameter estimation problem.
Moving average signals

Consider
\[ y(t) = \sum_{k=0}^{q} b_k e(t-k) = c(t) + b_1 e(t-1) + \ldots + b_q e(t-q) \]

Multiplying by \( y^*(t-k) \) and taking \( E\{\cdot\} \), yields
\[ r(k) = 0 \quad \text{for } |k| > q, \]
and thus,
\[ \phi(\omega) = |B(\omega)|^2 \sigma_e^2 = \sum_{k=-q}^{q} r(k)e^{-ik\omega} \]

Two approaches:
(i) Estimate \( \{b_k\} \) and \( \sigma_e^2 \) and compute \( \phi(\omega) \). Non-linear estimation problem, but \( \phi(\omega) \geq 0 \).
(ii) Estimate \( \{r(k)\} \) and compute \( \phi(\omega) \). This is \( \hat{\phi}_RT(\omega) \) with a rectangular lag window. However, \( \phi(\omega) \) is not guaranteed to be \( \geq 0 \).

ARMA signals

ARMA models can represent spectra with both peaks and valleys. Let \( a_0 = b_0 = 1 \),
\[ \sum_{i=0}^{n} a_i y(t-i) = \sum_{j=0}^{n} b_j e(t-j) \]
or equivalently, as \( H(q) \) is asymptotically stable and causal,
\[ y(t) = \frac{B(q)}{A(q)} e(t) = H(q) e(t) = \sum_{k=0}^{\infty} b_k e(t-k) \]

Multiplying with \( y^*(t-k) \) and taking \( E\{\cdot\} \) yields
\[ \sum_{i=0}^{n} a_i r(k-l) = \sum_{j=0}^{n} b_j E\{e(t-j)e^*(t-k)\} = \sigma_e^2 \sum_{j=0}^{\infty} b_j b_{j-k}^* \]

Note that
\[ \sum_{i=0}^{n} a_i r(k-l) = 0, \quad \text{for } k > q \]

ARMA signals

Thus,
\[
\begin{bmatrix}
  r(q) & \ldots & r(q-p+1) \\
  \vdots & \ddots & \vdots \\
  r(q-p+1) & \ldots & r(q)
\end{bmatrix}
\begin{bmatrix}
a_1 \\
\vdots \\
 a_p
\end{bmatrix}
= \begin{bmatrix}
r(q+1) \\
\vdots \\
 r(q+p)
\end{bmatrix}
\]

These are the modified Yule-Walker (MYW) equations. In general,
(i) \( R \) is nonsingular under mild conditions.
(ii) \( R \) is Toeplitz, but not Hermitian.
(iii) Levinson-type fast algorithms exist (about 2 \times \text{LDA}).
**ARMA signals**

But,

\[ \gamma_k = E \{ u(t)u(t-k) \} \]
\[ = E \left\{ \sum_{l=0}^{p} a_l y(t-l) \sum_{j=0}^{r} a_j^* y^*(t-j-k) \right\} \]
\[ = \sum_{l=0}^{p} \sum_{j=0}^{r} a_l a_j^* \kappa(k + j - l). \]

As \( \hat{a}_1, \ldots, \hat{a}_p \) can be found as the solution to the MYW equations,

\[ \begin{cases} \hat{\gamma}_k = \sum_{l=0}^{p} \sum_{j=0}^{r} \hat{a}_l \hat{a}_j^* \kappa(k + j - l), & k = 0, \ldots, q \\ \hat{\gamma}_{-k} = \hat{\gamma}_k, & k = -1, \ldots, -q \end{cases} \]

**Overdetermined systems**

Note that as \( r(k) \) is slowly decaying for narrowband signals, one can make use of \( M > p \) equations, yielding

\[
\begin{bmatrix}
 r(q) & \cdots & r(q - p + 1) \\
 \vdots & \ddots & \vdots \\
 r(q + p - 1) & \cdots & r(q) \\
 \vdots & \ddots & \vdots \\
 r(q + M - 1) & \cdots & r(q + M - p)
\end{bmatrix}
\begin{bmatrix}
 a_1 \\
 \vdots \\
 a_p \\
 0 \\
 \vdots \\
 0 \\
 \vdots \\
 a_p \\
 0 \\
 \vdots \\
 0 \\
 \vdots \\
 0
\end{bmatrix}
\approx
\begin{bmatrix}
 r(q + 1) \\
 \vdots \\
 r(q + p) \\
 \vdots \\
 r(q + M)
\end{bmatrix}
\]

The **overdetermined** system is typically solved using either

(i) least squares, or

(ii) total least squares

**The modified Yule-Walker estimator**

The modified YW spectral estimator is found as

\[ \hat{\phi}_{MYW}(\omega) = \frac{\sum_{k=-\infty}^{\infty} \hat{\gamma}_k e^{-i k \omega}}{|A(\omega)|^2} \]

**Remarks**

(i) Due to the MA part, \( \hat{\phi}_{MYW}(\omega) \) is not guaranteed to be \( \geq 0, \forall \omega \).

(ii) The AR estimates \( \hat{a}_1, \ldots, \hat{a}_p \) have **reasonable** accuracy if the ARMA poles and zeroes are well inside the unit circle, but are **very poor** when the poles and zeroes are closely-spaced and nearby the unit circle (the narrowband signal case).
The total least squares (TLS) solution is found by instead minimizing the shortest distance (squared) between the line and the points.

Let \( C = [ A \mid b ] = U \Sigma V^\ast \). The TLS solution is found as

\[
x_{\text{TLS}} = -V_{12} V_{22}^{-1}
\]

Remark: at low SNR, TLS may be better than LS. At high, similar.

### Overdetermined systems

The two-stage least squares method

If the noise sequence was known, the ARMA estimation problem would be one of system identification, solvable using least-squares. Assume that the ARMA model is invertible, i.e.,

\[
e(t) = A(q) y(t) = y(t) + \alpha_1 y(t-1) + \alpha_2 y(t-2) + \ldots
\]

(i) Approximate as AR(\( K \)) using some large \( K \),

\[
e(t) \approx y(t) + \hat{\alpha}_1 y(t-1) + \ldots + \hat{\alpha}_K y(t-K)
\]

Estimate \( \hat{\alpha}_1, \ldots, \hat{\alpha}_K \) as well as the residual

\[
e(t) = y(t) + \hat{\alpha}_1 y(t-1) + \ldots + \hat{\alpha}_K y(t-K)
\]

(ii) Replace \( \{e(t)\} \) with \( \{\epsilon(t)\} \) in \( A(q) y(t) \approx B(q) \epsilon(t) \), and estimate \( \{a_j, b_j\} \) using least-squares.

### The amplitude spectrum

Consider a signal consisting of a sum of sinusoids,

\[
y(t) = x(t) + e(t) = \sum_{k=1}^{\infty} \alpha_k e^{i \omega_k t + i \phi_k} + e(t),
\]

where the additive noise, \( e(t) \), is assumed to be complex-valued circular white Gaussian noise with variance \( \sigma^2_e \). Here, \( \alpha_k > 0, \omega \in [-\pi, \pi] \). Model \( \phi_k \) as independent r.v. uniformly distributed on \( [-\pi, \pi] \). The ACS for such a signal is

\[
r(k) = E\{y(t) y^*(t-k)\} = \sum_{p=1}^{\infty} \alpha_p^2 \delta_{p+k} + \sigma^2_e \delta_k
\]

where \( \delta_k \) is the Kronecker delta. Normally, we are primarily interested in estimating \( \omega_k \) (\( \omega_k \) and \( \phi_k \) are easily solved for using LS). Thus,

\[
\phi_k(\omega) = \phi_k(\omega) + \phi_k(\omega) = 2\pi \sum_{p=1}^{\infty} \alpha_p^2 \delta(\omega - \omega_p) + \sigma^2_e,
\]

where \( \delta(\omega) \) is the Diarc delta. Note that \( \phi(\omega) = \infty \) at each \( \omega_p \).
The amplitude spectrum

It is often more convenient to represent the PSD

\[ \phi_y(\omega) = \phi_x(\omega) + \phi_c(\omega) = 2\pi \sum_{p=1}^{N} \alpha_p^2 \delta(\omega - \omega_p) + \sigma_e^2, \]

using the amplitude spectrum, i.e.,

\[ \phi_c^y(\omega) = \begin{cases} \alpha_p, & \omega = \omega_p \\ 0, & \omega \neq \omega_p \end{cases} \]

The non-linear least-squares estimator

For white noise, the ML minimization simplifies to finding

\[
\min f(\omega, \alpha, \varphi) = \min \sum_{i=1}^{N} y_i(t) - \sum_{k=1}^{n} \alpha_k e^{\omega_k t + \varphi_k} \right|^2,
\]

normally termed the NLS estimate. Let \( \beta_k = \alpha_k e^{\varphi_k} \), and

\[
\beta = \begin{bmatrix} \beta_1 \ldots \beta_n \end{bmatrix}^T, y = \begin{bmatrix} y(1) \ldots y(N) \end{bmatrix}^T, B = \begin{bmatrix} e^{\omega_1 t} \ldots e^{\omega_n t} \\ e^{\omega_1 N} \ldots e^{\omega_n N} \end{bmatrix}.
\]

Then,

\[
f(\omega, \alpha, \varphi) = (y - B\beta)^T(y - B\beta) = \|B^T y - \beta\|^2 = \|y - B^T B y\|^2 + \gamma^2 - y^T B^T (y - B\beta),
\]

The maximum likelihood estimator

The ML estimator is found as the estimate that maximizes the log-likelihood function of the measured signal,

\[
\hat{\theta} = \arg \max_{\theta} \left\{ \ln p_y(\theta) \right\},
\]

where \( p_y(\theta) \) is the pdf of the vector \( y \) described by the parameters \( \theta \). For a \( N \)-sample complex Gaussian vector, \( y \sim CN(\mu_y, C_y) \),

\[
p_y(\theta) = \frac{1}{\pi^N \det[C_y]} e^{-\left(y - \mu_y\right)^T C_y^{-1} \left(y - \mu_y\right)}
\]

where \( C_y = E(yy^T) \) and \( \mu_y = E(y) \); the ML estimate is obtained as

\[
\hat{\theta} = \arg \min_{\theta} \left\{ \ln \det[C_y] + \left(y - \mu_y\right)^T C_y^{-1} \left(y - \mu_y\right) \right\}
\]

In general, the ML estimate is computationally prohibitive.

The non-linear least-squares estimator

The minimum of \( f(\omega, \alpha, \varphi) \) is thus found for

\[
\hat{\beta} = (B^T B)^{-1} B^T y
\]

and as a result

\[
\hat{\omega} = \arg \max_{\omega} \left\{ y^T (B^T B)^{-1} B^T y \right\} = \arg \max_{\omega} \operatorname{tr}(B^T B y y^T) = \arg \max_{\omega} \operatorname{tr}(\hat{B}_B y y^T),
\]

where \( B_0 = B(B^T B)^{-1} B^T \) is the projection onto \( R(B) \).

Remarks:

(i) The NLS has excellent accuracy, with \( \operatorname{Var}(\hat{\omega}) = \frac{4\pi^2}{N^2} \), for \( N \gg 1 \)
(ii) The cost function is multimodal and requires an \( n \)-dimensional search.
(iii) If \( n = 1 \),

\[
y^T (B^T B)^{-1} B^T y = \left\| y/\sqrt{N} \right\|^2 = \hat{\omega}_1^2, \]

The NLS estimate is found as the peak of the Periodogram.
The Cramér-Rao lower bound

Let \( p_y(\theta) \) denote the likelihood function of a data vector, \( y \), described by the parameters \( \theta \), and let

\[
P = E\left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^\top \right\}
\]

denote the covariance matrix of the estimation error \( \hat{\theta} - \theta \). The Cramér-Rao lower bound (CRB) states that \( P \geq P_{CRB} \), for any unbiased estimate of \( \theta \), where

\[
P^{-1}_{CRB}(\theta) = E\left\{ \frac{\partial \ln p_y(\theta)}{\partial \theta} \left( \frac{\partial \ln p_y(\theta)}{\partial \theta} \right)^\top \right\}
\]

The matrix \( P^{-1}_{CRB}(\theta) \) is termed the Fisher information matrix.

49

The Cramér-Rao lower bound

For a sum of sinusoids in white complex Gaussian noise,

\[
y(t) = x(t, \theta) + e(t) = \sum_{k=1}^{N} \alpha_k e^{j\omega_k t + j\phi_k} + e(t),
\]

or, in vector form, \( y = x(\theta) + e \), where

\[
x(\theta) = \begin{bmatrix} x(0, \theta) & \ldots & x(N-1, \theta) \end{bmatrix}^\top,
\]

with

\[
\theta = \begin{bmatrix} \alpha_1 & \phi_1 & \omega_1 & \ldots & \alpha_n & \phi_n & \omega_n \end{bmatrix}^\top
\]

the Slepian-Bangs formula asymptotically yields

\[
P_{CRB}(\theta) = \begin{bmatrix} \Psi_1 & 0 & \cdots & 0 \\ \cdots & \Psi_n & \cdots & 0 \\ 0 & \cdots & 0 & \Psi_n \end{bmatrix},
\]

\[
\Psi_k = \begin{bmatrix} \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} & 0 & \cdots & 0 \\ 0 & \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} \end{bmatrix}
\]

50

The Cramér-Rao lower bound

For a sum of sinusoids in colored Gaussian noise, \( y = x(\theta) + e \), the log-likelihood function can be written as

\[
\ln p_y(e|\theta) = -\ln |Q(\mu)| - |y - x(\theta)|^2 Q^{-1}(\mu) (y - x(\theta))
\]

Then, the CRB matrix is given by the Slepian-Bangs formula,

\[
P^{-1}_{CRB}(\theta) = \begin{bmatrix} \Psi \end{bmatrix} 2 \Re \left( \frac{\partial x^\top(\theta)}{\partial \eta_k} Q^{-1}(\mu) \frac{\partial x(\theta)}{\partial \eta_k} \right)
\]

where \( Q(\mu) = E(ee^\top) \), and

\[
\eta = \begin{bmatrix} \theta \\ \mu \end{bmatrix}
\]

51

The Cramér-Rao lower bound

For a sum of sinusoids in colored Gaussian noise, the CRB expression is very involved. A more convenient bound is the asymptotic CRB,

\[
P_{ACRB}(\theta) = \lim_{N \to \infty} K_N P_{CRB}(\theta) K_N^\top
\]

where \( K_N \) is block-diagonal with \( K_N \) along the diagonal. Here,

\[
P_{ACRB}(\theta) = \begin{bmatrix} \Psi_1 & 0 & \cdots & 0 \\ 0 & \Psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Psi_n \end{bmatrix},
\]

\[
K_N = \begin{bmatrix} N^{1/2} & 0 & 0 \\ 0 & N^{1/2} & 0 \\ 0 & 0 & N^{1/2} \end{bmatrix}
\]

and, with \( \phi_k(\omega) \) being the PSD of the noise at frequency \( \omega \),

\[
\Psi_k = \begin{bmatrix} \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} & 0 & \cdots & 0 \\ 0 & \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\phi_k(\omega_k)}{\phi_k(\omega_k) - \phi_k(\omega_k)^2} \end{bmatrix}
\]

(From: Slepian, W.; Carlson, A. B. and Smith, C. A. Proceedings, IEEE, 1974). 52
The Cramér-Rao lower bound

Remarks

(i) For a sum of sinusoids in white Gaussian noise,

$$P_{\text{CRB}}(\hat{\omega}_k) = \frac{6\sigma_n^2}{N^2\omega_k^4}$$

Thus, the NLS estimator is efficient. The NLS estimator is equivalent to the ML estimator.

(ii) For a sum of sinusoids in colored Gaussian noise, the NLS is asymptotically efficient. The NLS estimator is different from the ML estimator.

(iii) For $n = 1$, the Periodogram is the ML estimator in the white noise case. In the colored noise case, it asymptotically achieves the same performance.

(iv) In practice, the NLS estimator is too complex.

The covariance matrix model

Consider $y(t) = x(t) + e(t) = \sum_{k=1}^{n} a_k e^{i\omega_k t + \phi_k} + e(t)$ and let

$$a[\omega] = \begin{bmatrix} 1 & e^{-i\omega_1} & \ldots & e^{-i\omega_{n-1}} \end{bmatrix}^T$$
$$A = \begin{bmatrix} a[\omega_1] & \cdots & a[\omega_n] \end{bmatrix}$$

Here, $A$ is $m \times n$ Vandermonde matrix, i.e., rank($A$) = $n$. Define

$$y_t = \begin{bmatrix} y(t) & \ldots & y(t-m+1) \end{bmatrix}^T = Ax_t + e_t,$$

where

$$x_t = \begin{bmatrix} a_1 e^{i\omega_1 t + \phi_1} & \ldots & a_n e^{i\omega_n t + \phi_n} \end{bmatrix}^T$$
$$e_t = \begin{bmatrix} e(t) & \ldots & e(t-m+1) \end{bmatrix}^T$$

Then,

$$R_y = E\{y_t^*y_t\} = APA^* + \sigma_n^2 I$$

with

$$P = E\{x_t^*x_t\} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

The covariance matrix model

Thus,$$
A = \begin{bmatrix} \lambda_1 - \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \lambda_2 - \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n - \sigma_n^2 \end{bmatrix}$$

is non-singular. Further, as $G^*S = 0$,

$$R_yS = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} = APA^*S + \sigma_n^2 S$$

Also,$$
S = APA^*SA^{-1} = AC$$

and $R(S) \subset R(A)$.
The MUSIC algorithm

Let $\hat{R}_y = \frac{1}{N} \sum_{t=0}^{N-1} y_t y_t^*$ and form $\mathbf{S}$ and $\mathbf{G}$ from the eigenvectors of $\hat{R}_y$. The (spectral) Multiple Signal Classification (MUSIC) estimates are found as the $n$ highest peaks of

$$\varphi(\omega) = \frac{1}{\hat{a}^*(\omega) \mathbf{G} \hat{G}^* \hat{a}(\omega)}, \quad \omega \in [-\pi, \pi]$$

where $\varphi(\omega)$ is normally denoted the MUSIC "pseudo-spectrum".

As an alternative, the (root) MUSIC estimates are found as the $n$ roots of

$$\hat{a}(z) = 0$$

that are closest to the unit circle. Here, $z = e^{j\omega}$, and

$$\hat{a}(z) = \begin{bmatrix} 1 & z^{-1} & \ldots & z^{-m+1} \end{bmatrix}^T$$

Remarks:

(i) Spectral MUSIC requires a grid search, with resolution limited by the grid; root-MUSIC does not suffer this restriction.

(ii) Spectral MUSIC has lower resolution than root-MUSIC.

(iii) Both the spectral- and root-MUSIC methods may produce spurious frequency estimates. Solved using modified MUSIC.

(iv) For $m = n + 1$ (the minimum value), $\hat{G} = \hat{g}_1$, i.e., $\hat{g}^* \hat{a}(z) = 0$. The so-called Pisarenko method does not suffer spurious frequency estimates, is computationally simple, but is (much) less accurate than MUSIC with $m \gg n + 1$.

(v) As $I = \mathbf{S}^* + \mathbf{G}^*$, one can use $I - \mathbf{S}^*$ instead of $\mathbf{G}^*$. 
The ESPRIT algorithm

Let $A_1 = [I_{m-1} \ 0 \ | \ A]$ and $A_2 = [0 \ | \ I_{m-1} \ | \ A]$. Then,

$$A_2 = A_1 D,$$

where

$$D = \begin{bmatrix} e^{-j\omega_1} & & \cdots \ & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & \cdots & & e^{-j\omega_n} \end{bmatrix}$$

Similarly, let $S_1 = [I_{m-1} \ 0 \ | \ S]$ and $S_2 = [0 \ | \ I_{m-1} \ | \ S]$. Recall that $S = AC$. Thus,

$$S_2 = A_2 C = A_1 D C = S_1 C^{-1} D C = S_i \Psi,$$

yielding the least-squares estimate

$$\hat{\Psi} = (S_i^* S_i)^{-1} S_i^* S_2.$$

The subspace fitting algorithms

Recall that

$$R_p = APA^* + \sigma^2 I = SA_S^* + \sigma^2 GG^*$$

As $GG^* = I - SS^*$,

$$APA^* + \sigma^2 SS^* = SA_S^*$$

Post-multiply with $S$,

$$APA^* S = SA_S - \sigma^2 S \quad \Rightarrow \quad S = APA^* S (A_S - \sigma^2 I)^{-1} = AT$$

The signal subspace fitting (SSF) estimate is obtained as

$$\hat{\omega, \hat{T}} = \arg \min_{\omega, T} \| S - AT \|^2_F$$

where $\| X \|^2_F = \text{tr}(X^*X)$ denotes the Frobenius norm.

The ESPRIT algorithm

Note that

$$\Psi = C^{-1} DC$$

is a similarity transformation, i.e., $\Psi$ and $D$ share the same eigenvalues. This as

$$\text{det} [\Psi - \lambda I] = \text{det} [C^{-1} (D - \lambda I) C] = \text{det} [C^{-1}] \text{det} [D - \lambda I] \text{det} [C] = \text{det} [D - \lambda I]$$

However, $\lambda(D) = \lambda(\Psi) = (e^{-j\omega_k})_{k=1}^n$, and thus $\omega_k = -\arg(\lambda_k)$. The ESPRIT (Estimation of Signal Parameters by Rotational Invariance Techniques) estimates are found as the angular position of the eigenvalues of $\Psi$.

Remarks

(i) ESPRIT is computationally simple, and does not produce spurious estimates.

(ii) It is often preferable to use the TLS estimate of $\Psi$ instead.

The subspace fitting algorithms

As (for a fixed unknown $A$),

$$\hat{T} = A^* S = (A^* A)^{-1} A^* S,$$

the SSF estimate can be rewritten as

$$\hat{\omega} = \arg \min_{\omega} \text{tr} \left( \Pi^* \hat{S} \hat{A}^* \hat{S}^* \right)$$

Since the eigenvectors are estimated with different quality, it is natural to include a weighting $W$ of the eigenvectors, i.e.,

$$\hat{\omega} = \arg \min_{\omega} \text{tr} \left( \Pi^* W \hat{S} \hat{W}^* \hat{S}^* \right)$$

It is possible to show that to minimize the estimation error variance, the weighting should be chosen as

$$W = (\hat{A}_S - \sigma^2 I)^2 \hat{A}_S^{-1},$$

yielding the so-called weighted subspace fitting (WSF) estimate.
The subspace fitting algorithms

Remarks:

(i) One can instead use $A^*G = 0$, obtaining the noise subspace fitting (NSF) algorithm,

$$\omega = \arg\min_{\omega} \text{tr} \left( A^*GG^*A^* \right),$$

where $V$ is a weighting matrix. If chosen as $V = A^*\hat{S}W\hat{S}^*A^*$, the NSF asymptotically coincide with the WSF. The MUSIC estimate is obtained if $V = I$.

(ii) The WSF estimate can be efficiently computed using MODE.

(iii) The WSF is very accurate, being a strong candidate for “best” approach.

[Good overview in Krim & Viberg, IEEE SP M, 1996]

The MODE algorithm

Therefore, the MODE (also, root-WSF) estimate can be obtained as:

1. Solve the quadratic problem

$$\hat{b} = \arg\min_{\hat{b}} \text{tr} \left( BB^*\hat{S}W\hat{S}^* \right)$$

2. Form $\hat{B}$ using $\hat{b}$. Solve the quadratic problem

$$\hat{b} = \arg\min_{\hat{b}} \text{tr} \left( B(B^*B)^{-1}B^*\hat{S}W\hat{S}^* \right)$$

3. Obtain the frequency estimates as the angle of the roots of $\hat{b}$.

The second step will yield estimates having the best possible asymptotic accuracy. Aside from the eigendecomposition and the polynomial rooting, the algorithm is essentially in closed form.

The RELAX algorithm

Recall that the NLS is asymptotically efficient in the colored noise case, but that the minimization is difficult. One solution is to use a relaxation approach. Consider all but the $k$th sinusoid as known (or estimated), and let

$$y_k(t) = y(t) - \sum_{j=1, j \neq k}^{N} \hat{a}_j e^{j\hat{\omega}_j t + j\hat{\phi}_j}$$

For this case, minimizing $f_k(\omega, \alpha, \varphi)$ yields

$$\hat{\omega}_k = \arg\max_{\omega_k} \hat{\phi}_k(\omega)$$

$$\hat{\eta}_k = \hat{\gamma}_k(\hat{\omega}_k)$$

Both estimates are obtained from the Periodogram of $y_k(t)$. 

The RELAX algorithm

Steps:
1. Assume $n = 1$. Obtain $\hat{\omega}_1$ and $\hat{\beta}_1$ from $y(t)$.
2. Assume $n = 2$. Obtain $y_2(t)$ using $\hat{\omega}_1$ and $\hat{\beta}_1$ from Step 1. Iterate until $f(\omega, \alpha, \varphi)$ does not decrease “significantly” anymore:
   - Obtain $\hat{\omega}_2$ and $\hat{\beta}_2$ from $y_2(t)$.
   - Obtain $y_1(t)$ by using $\hat{\omega}_2$ and $\hat{\beta}_2$.
   - Estimate $\hat{\omega}_1$ and $\hat{\beta}_1$ from $y_1(t)$.
3. Assume $n = 3$. Obtain $y_3(t)$ using $\hat{\omega}_1, \hat{\beta}_1, \hat{\omega}_2, \hat{\beta}_2$ from Step 2. Iterate until $f(\omega, \alpha, \varphi)$ does not decrease “significantly” anymore:
   - Obtain $\hat{\omega}_3$ and $\hat{\beta}_3$ from $y_3(t)$. Obtain $y_2(t)$ from $\hat{\omega}_3, \hat{\beta}_3, \hat{\omega}_2, \hat{\beta}_2$.
   - Estimate $\hat{\omega}_1$ and $\hat{\beta}_1$ from $y_2(t)$. Obtain $y_1(t)$ from $\hat{\omega}_1, \hat{\beta}_1, \hat{\omega}_2, \hat{\beta}_2$.
   - Estimate $\hat{\omega}_2$ and $\hat{\beta}_2$ from $y_3(t)$.
4. Assume $n = 4, \ldots$

Continue the recursion until $n$ is large enough!

Remarks:
(i) The RELAX algorithm is often better than other high-resolution algorithms, especially in obtaining better $\hat{\beta}_k, k = 1, \ldots, n$.
(ii) The algorithm is fairly robust to the choice of $n$, and to data model errors.
(iii) The algorithm can be numerically cumbersome, especially for larger $n$.

Subspace based model order estimation

As mentioned, $A^*G = 0$. The level of orthogonality between these spaces for a nominal model order, $L$, can thus be used to estimate the unknown model order. The average of the principal angles between the spaces can be expressed as

$$J_L = \frac{1}{K} \sum_{k=1}^{K} \cos^2(\theta_k) \approx \frac{1}{MK} ||A^*G||_F^2$$

where $K = \min(L, M - L)$. The resulting estimator

- works for both damped and undamped sinusoids.
- works for both Gaussian and non-Gaussian (colored) noise.

[Christensen, Jakobsson & Jensen, J Adv Sig Proc, 2009]