EURECOM 07/06/2010

Statistical Signal Processing SSP

TP : Sinusoidal Spectrum and Parameter Estimation

- The students are asked to submit a report by 30/06/10.
  The report contains the answers to the questions (in boldface) below, and includes plots and Matlab code.
  The TP should be carried out individually since we want to make sure that every student will be able to do some Matlab simulations after the TP. As usual, you can consult moderately with others if you get stuck at some point, but your report/software should be written in your own words.

Part I: Spectrum Estimation

Review

Given N samples $\{y_0, y_1, \ldots, y_{N-1}\}$, the periodogram is given by

$$\widehat{S}_{yy}(f) \;=\; \widehat{S}_{PER}(f) \;=\; \frac{1}{N}\left|\sum_{n=0}^{N-1} w_{N,n}\, y_n\, e^{-j2\pi fn}\right|^2 \tag{1}$$

where $w_{N,n}$ is a windowing function of size N (rectangular by default). The window is normalized in the sense that $\frac{1}{N}\sum_{n=0}^{N-1} w_{N,n}^2 = 1$. In practice, the Fourier transform is computed as the Discrete Fourier Transform (DFT), for which fast algorithms exist known as Fast Fourier Transform (FFT). For computation of the DFT/FFT, it is customary to consider the frequency interval $[0,1]$ rather than $[-\frac{1}{2},\frac{1}{2}]$. The DFT evaluates the Fourier transform of a signal of length N at N equispaced frequencies $f_k = k/N\,,\; k = 0,1,\ldots,N{-}1$. So we get

$$\widehat{S}_{PER}(f_k) \;=\; \frac{1}{N}\left|\sum_{n=0}^{N-1} w_{N,n}\, y_n\, e^{-j2\pi f_k n}\right|^2 \;=\; \frac{1}{N}\left|\sum_{n=0}^{N-1} w_{N,n}\, y_n\, e^{-j2\pi \frac{k}{N} n}\right|^2 \quad , \quad k = 0,1,\ldots,N{-}1\;. \tag{2}$$

We can obtain a finer frequency spacing by padding the data with $N'{-}N$ zeros and then applying an N'-point DFT. The effective data set becomes

$$y'_n \;=\; \begin{cases} y_n & ,\; n = 0,1,\ldots,N{-}1 \\ 0 & ,\; n = N, N{+}1,\ldots,N'{-}1 \end{cases} \tag{3}$$

which has the same Fourier transform as the original data set. The frequency spacing of the DFT on the data set y'_n will be $\frac{1}{N'} < \frac{1}{N}$. This **zero padding** gives no extra resolution, but only a finer evaluation of the periodogram. The window is still of size N.

In the autoregressive (AR) modeling (parametric) approach, we take

$$\widehat{S}_{AR}(f) \;=\; \frac{\sigma^2_{f,n}}{|A_n(f)|^2} \tag{4}$$

for an AR model of order n. The prediction coefficients $A_n = [1\ A_{n,1} \cdots A_{n,n}]^T$ and the prediction error variance are obtained from the Yule-Walker (or normal) equations

$$R_{n+1} A_n \;=\; \begin{bmatrix} r_0 & r_1 & \cdots & r_n \\ r_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_n & \cdots & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ A_{n,1} \\ \vdots \\ A_{n,n} \end{bmatrix} \;=\; \begin{bmatrix} \sigma^2_{f,n} \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{5}$$

where the correlation sequence is estimated as

$$r_k \;=\; \widehat{r}_{yy}(k) \;=\; \frac{1}{N} \sum_{n=0}^{N-1-k} y_{n+k} y_n \;, \quad k = 0, 1, \ldots, n\;. \tag{6}$$

The frequencies f in (4) are again evaluated at the same discrete set. The prediction filter can be computed in an order-recursive fashion via the Levinson algorithm: $\left\{ \begin{array}{l} A_n \\ \sigma^2_{f,n} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} A_{n+1} \\ \sigma^2_{f,n+1} \end{array} \right.$

$$\begin{aligned} \Delta_{n+1} &= [r_{n+1} \cdots r_1]\ A_n \\ K_{n+1} &= -\tfrac{\Delta_{n+1}}{\sigma^2_{f,n}} \\ A_{n+1} &= \begin{bmatrix} A_n \\ 0 \end{bmatrix} + K_{n+1} \begin{bmatrix} 0 \\ J\, A_n \end{bmatrix} \\ \sigma^2_{f,n+1} &= \sigma^2_{f,n} \left(1 - K^2_{n+1}\right) \end{aligned}$$

Initialization: $A_0 = [1]$, $\sigma^2_{f,0} = r_0$.

TP environment

You will be working on a signal consisting of two real sinusoids in white noise, a signal that can be generated with the Matlab command file

`\\datas\teaching\courses\SSP\tp1matlab\sig.m`

that you can copy in your working directory (in Matlab, do “help sig” to get information on the function sig.m . You can also take a look at the file sig.m to get some insight into how Matlab works.). In the same directory you can find:

`\\datas\teaching\courses\SSP\tp1matlab\periodo.m`

which is a function that computes a periodogram (do “help periodo” in Matlab to get information on this function).

`\\datas\teaching\courses\SSP\tp1matlab\window.m`

is a function that provides six window types (window coefficients are provided in a vector).

`\\datas\teaching\courses\SSP\tp1matlab\guide_matlab.pdf`

contains some elementary explanations of Matlab functions in French.

Start Matlab by typing “matlab”. The command “help” will give you information about Matlab, while typing “help *commandname*” will give you information on the command *commandname*.

For each of the plots below to be handed in, create a title which contains at least the number (a), (b) etc. of the question that the plot addresses.

TP Session

Your report contains the answers to the questions below in boldface.
The random process that we shall use is of the form

$$y_k = A_1 \cos(2\pi f_1 k + \phi_1) + A_2 \cos(2\pi f_2 k + \phi_2) + v_k \quad (7)$$

where v_k is zero-mean unit variance white Gaussian noise. The phases ϕ_i are uniform over $[0, 2\pi]$, whereas the default values for the sinusoid amplitudes and frequencies are $A_1 = 20$, $A_2 = 20$, $f_1 = 0.057$, $f_2 = 0.082$.

(a) For the periodogram, how many data points N are required to resolve the two sinusoids? Why?

(b) Generate (using sig.m) one realization of y_k with $N = 256$ samples and compute (with the rectangular 'boxcar' window in periodo.m) the periodogram in $N' = 64$, 128, 256, 512 and 1024 frequency points (**make a plot with 6 subplots**). When $N' < N$, use only the first N' data samples (explicitely); when $N' > N$, use zero padding (done automatically by periodo.m). Comment on the effect of N' and **formulate a requirement for N' as a function of N for proper evaluation of the periodogram.**

(c) Estimate the correlation sequence of the signal. This can be done by convolving the signal with the time-reversed version of the signal. If for instance "y" contains the 1024 signal samples, the reversed version of "y" can be produced in Matlab via the expression "y(1024:-1:1)". Convolution can be performed with the Matlab command "conv" ("help conv"). The convolution result will be of length 2047. The correlation sequence $r_{0:n}$ will be the elements 1024:1024+n of the convolution, divided by 1024.

(d) Write a Matlab function "levinson.m" ("help function") that takes as input the correlation sequence $r_{0:20}$ and produces as output the sequence of prediction error variances $\sigma^2_{f,0:20}$, the prediction error filter coefficients of order 20 $A_{20,0:20}$ ($A_{20,0} = 1$), and the sequence of PARCORS $K_{1:20}$.
Hand in the levinson.m routine.
Plot in a single plot the periodogram (with $N = 1024$) and the autoregressive spectrum estimate corresponding to a prediction order of 20.
Hand in a plot with these two curves, as also the values of $A_{20,0:20}$, and a plot of the evolution of $\sigma^2_{f,0:20}$ and the PARCORS.
For plotting the autoregressive spectrum estimate, append zeros to the prediction error filter coefficients, to the desired number of frequency points.

Part II: Parameter Estimation

Background on ML Estimation of Sinusoid in Noise Parameters

For this second part, consider a single sinusoid in white Gaussian noise with variance σ^2

$$y_k = s_k + v_k = A_1 \cos(2\pi\, f_1\, k + \phi_1) + v_k \quad , \quad k = 0, \ldots, n-1 \ . \tag{8}$$

The Maximum Likelihood estimates of the paramteres σ^2, A_1, ϕ_1 and f_1 can be shown to be obtained as

$$\begin{aligned}
\widehat{f}_1 &= \arg\max_f |\mathcal{Y}(f)| \\
\widehat{A}_1 &= \tfrac{2}{n} \left|\mathcal{Y}(\widehat{f}_1)\right| \\
\widehat{\phi}_1 &= \arg \mathcal{Y}(\widehat{f}_1) \\
\widehat{\sigma^2} &= \tfrac{1}{n} \sum_{k=0}^{n-1} (y_k - \widehat{A}_1 \cos(2\pi\, \widehat{f}_1\, k + \widehat{\phi}_1))^2
\end{aligned} \tag{9}$$

where $\mathcal{Y}(f) = \sum_{k=0}^{n-1} y_k\, e^{j\, 2\pi\, f\, k}$ and $\arg\left\{\rho\, e^{j\,\theta}\right\} = \theta$. In order to turn the optimization problem for $\widehat{f}_1$ into a practical algorithm, we shall use the DFT. First decide on an acceptable "bias" in the ability to resolve the maximum of $|\mathcal{Y}(f)|$. Let's call the frequency resolution Δf:

$$\Delta f \;=\; \frac{1}{m} \;, \quad m \;=\; \frac{1}{\Delta f} \;. \tag{10}$$

In order to have a DFT with such a frequency resolution, we need to have a signal of length m. We assume $m > n$, the number of samples available. Hence zero pad y_k to obtain $y_0, y_1, \ldots, y_{m-1}$ where in fact $y_n = y_{n+1} = y_{m-1} = 0$. Take the DFT of the zero padded sequence (in Matlab, y_0 is the first element of a signal vector, y_1 the second etc.)

$$\mathcal{Y}_l \;=\; \sum_{k=0}^{m-1} y_k\, e^{j2\pi\, \frac{l}{m}\, k} \;=\; \sum_{k=0}^{n-1} y_k\, e^{j2\pi\, \frac{l}{m}\, k} \;. \tag{11}$$

Then the Maximum Likelihood estimates can be approximatively obtained as

$$\begin{aligned}
&\widehat{l} \;=\; \arg\max_l |\mathcal{Y}_l| \;, \quad \widehat{f}_1 \;=\; \frac{\widehat{l}}{m} \;, \quad \widehat{A}_1 \;=\; \frac{2}{n} |\mathcal{Y}_{\widehat{l}}| \\
&\widehat{\phi}_1 \;=\; \arg \mathcal{Y}_{\widehat{l}} \;, \quad \widehat{\sigma^2} \;=\; \tfrac{1}{n} \sum_{k=0}^{n-1} (y_k - \widehat{A}_1 \cos(2\pi\, \widehat{f}_1\, k + \widehat{\phi}_1))^2
\end{aligned} \tag{12}$$

The search for f_1 should be limited to the interval $[0, \frac{1}{2}]$ (the DFT will show a symmetrical peak at $1 - \widehat{f}_1$). Note that zeropadding only allows us to get within $\Delta f = \frac{1}{m}$ of the maximum of $|\mathcal{Y}(f)|$. It does not improve the estimation accuracy (variance) of the estimator $\widehat{f}_1$. The variance can only be reduced by increasing the number of samples n (see Cramer-Rao bound).

The Cramer-Rao bounds for the estimation of the various parameters can be shown to be:

$$\begin{aligned}
CRB_{\widehat{\sigma^2}} &= \frac{2\,\sigma^4}{n} \qquad & CRB_{\widehat{A}_1} &= \frac{2\,\sigma^2}{n} \\
CRB_{\widehat{\phi}_1} &= \frac{8\,\sigma^2}{n\, A_1^2} \qquad & CRB_{\widehat{f}_1} &= \frac{24\,\sigma^2}{n^3\, A_1^2}
\end{aligned} \tag{13}$$

(e) Sinusoid in White Noise: ML Estimates and CRBs.

In Matlab, generate 10 realizations of a sinusoid in Gaussian white noise as in (8) with $\sigma^2 = 1$, $A_1 = \sqrt{2}$, $\phi_1 = 0$, $f_1 = 1/8$ and $n = 32$. Using the ML procedure given above, find for each of the 10 runs the estimates $\widehat{\sigma^2}(i)$, $\widehat{A}_1(i)$, $\widehat{\phi}_1(i)$ and $\widehat{f}_1(i)$, for $i = 1, \ldots, 10$.

Given the $CRB_{\widehat{f}_1} = \dfrac{24\,\sigma^2}{n^3\,A_1^2}$**, what is an appropriate choice for** $m = \frac{1}{\Delta f}$ **?**

Compute the sample mean and the sample variances for each of the four estimates for the 10 realizations. For instance, for the $\widehat{A}_1$, the sample mean and variance are

$$\overline{\widehat{A}_1} = \frac{1}{10}\sum_{i=1}^{10} \widehat{A}_1(i)\ , \quad Var\left(\widehat{A}_1\right) \ = \ \frac{1}{10}\sum_{i=1}^{10}\left(\widehat{A}_1(i) - \overline{\widehat{A}_1}\right)^2 \ . \tag{14}$$

Give the sample mean and sample variance for $\widehat{\sigma^2}$**,** $\widehat{A}_1$**,** $\widehat{\phi}_1$ **and** $\widehat{f}_1$**.**
Do the estimates for the four parameters appear to be biased?
Compare the sample variances of the four estimates to the Cramer-Rao lower bounds given in (13).

(f) Sinusoid in White Noise: Covariance Matching.

Another estimation method for the parameters in a parametric signal model consists of matching some moments of the random signal. If we assume for a moment that the phase of the sinusoidal signal would be random, then we get for the covariance sequence of the stationary noisy signal y_k

$$r_p = r_{yy}(p) = E\ y_k y_{k+p} \ = \ \frac{A_1^2}{2}\cos(2\pi\, f_1\, p) + \sigma^2\delta_{p0} \tag{15}$$

From the correlations at lags $p = 0, 1, 2$, we can retrieve the signal parameters as follows:

$$\begin{cases} X &= \ \frac{1}{2}\left(-r_2 + \sqrt{r_2^2 + 8\, r_1^2}\right) \\ A_1 &= \ \sqrt{2X} \\ f_1 &= \ \frac{1}{2\pi}\arccos\left(\frac{r_1}{X}\right) \\ \sigma^2 &= \ r_0 - X \ . \end{cases} \tag{16}$$

We now get estimates for these parameters by replacing the theoretical correlations r_p by sample correlations as in (6). In Matlab, generate 10 sinusoids in Gaussian white noise as in (8) with $\sigma^2 = 1$, $A_1 = \sqrt{2}$, $\phi_1 = 0$, $f_1 = 1/8$ and $n = 32$ (you can use the 10 realizations from question **(e)** above). Using the moment matching procedure given above, find for each of the 10 runs the estimates $\widehat{\sigma^2}(i)$, $\widehat{A}_1(i)$ and $\widehat{f}_1(i)$, for $i = 1, \ldots, 10$.

Give the sample mean and sample variance for $\widehat{\sigma^2}$**,** $\widehat{A}_1$ **and** $\widehat{f}_1$**.**
Do the estimates for these three parameters appear to be biased?
Compare the sample variances of the three estimates to the Cramer-Rao lower bounds given in (13) and to the sample variances obtained in question (e) above using the ML approach.