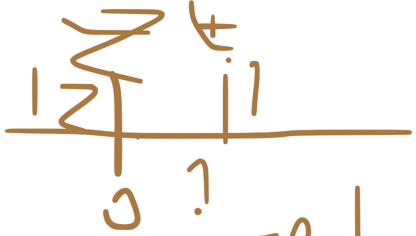
Statistical Signal Processing

Lecture 7

Parametric spectral estimation:

- parametric random process models: AutoRegressive (AR) processes
- linear prediction
- Levinson algorithm
- lattice filters
- AR modeling motivations: LP of an AR(N) process, asymptotics
- AR modeling interpretations, techniques, model order selection

Parametric Random Process Models



$$e_k \longrightarrow \boxed{\mathrm{H}(z)} \longrightarrow y_k$$

white noise

$z = e^{j2\pi f}$ $\quad |1 - z_i e^{-j2\pi f}|$

$z_i \to \frac{1}{z_i^*}$ $\quad \left|1 - \frac{1}{z_i^*} e^{-j2\pi f}\right| |z_i^*|$

- White noise drives a linear time-invariant causal and rational system

$$\mathrm{H}(z) \;=\; \sum_{k=0}^{\infty} h_k\, z^{-k} \;=\; \frac{\prod_i (1 - z_i z^{-1})}{\prod_k (1 - p_k z^{-1})} = \sum_i \alpha_i p_i^k = h_k$$

$|z_i^* e^{-j2\pi f}| =$

$=$

$= |1 - z_i e^{-j2\pi f}|$

- Zero mean input ⇒ zero mean output. Variance of the output

$$\sigma_y^2 \;=\; \sigma_e^2 \sum_{k=0}^{\infty} h_k^2$$

will be finite if the system is stable ($|p_k| < 1$: poles inside the unit circle). In that case, due to the wide-sense stationarity of the input, the output will be wide-sense stationary with power spectral density function (psdf)

$$S_{yy}(f) \;=\; \sigma_e^2\, |H(f)|^2$$

only magnitude counts

Parametric Random Process Models (2)

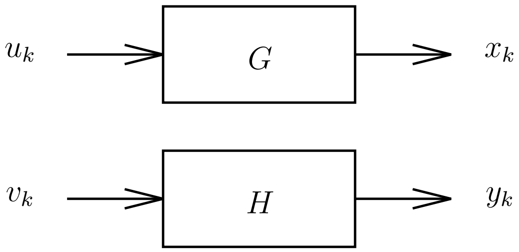
- For the modeling of $S_{yy}(f)$, only $|H(f)|$ is important. However, we desire to be able to determine $H(f)$ completely from $S_{yy}(f)$ and hence from $|H(f)|$. Can be done if $H(f)$ minimum-phase ($|z_i| < 1$: zeros inside the unite circle). Minimum-phase ⇒ system causally invertible ($\mathrm{H}^{-1}(z)$ is causal).

- Due to the uncorrelatedness of the input, we get

$$\mathrm{S}_{ye}(z) \;=\; \mathrm{H}(z)\,\sigma_e^2 \;\;\Rightarrow\;\; r_{ye}(k) \;=\; E\,y_{n+k}e_n \;=\; h_k\,\sigma_e^2 \;\;(=0,\;\; k<0\,)$$

⇒ output uncorrelated with future inputs.

- reminder: $\mathrm{S}_{xy}(z) = \mathrm{G}(z)\mathrm{S}_{uv}(z)\mathrm{H}^\dagger(z) \overset{z=e^{j2\pi f}}{\Longrightarrow} S_{xy}(f) = G(f)S_{uv}(f)H^*(f)$ in the figure:



Autoregressive (AR) Processes

- An Autoregressive process of order n (AR(n)) is obtained by taking an n-th order *all-pole* transfer function

$$= \prod_{i=1} (1 - p_i z^{-1})$$

$$\mathrm{H}(z) = \frac{1}{\mathrm{A}(z)}\,, \quad \mathrm{A}(z) = \sum_{i=0}^{n} A_i\, z^{-i}\,, \quad A_0 = 1$$

A(z) with $A_0 = 1$ is called a *monic* polynomial in z^{-1}. A(z) as a function of z needs to have all its roots inside the unit circle for H(z) to be minimum-phase.

- The input-output relation is described by the following difference equation
$y_k = \frac{1}{\mathrm{A}(q)} e_k \;\Rightarrow\; \mathrm{A}(q)\, y_k = e_k$ or

$$\mathrm{A}(q)\, y_k = \sum_{i=0}^{n} A_i\, \underline{q^{-i}}\, y_k = \sum_{i=0}^{n} A_i\, y_{k-i} = y_k + A_1 y_{k-1} + \cdots + A_n y_{k-n} = e_k$$

time domain →

where q^{-1} is the delay operator: $q^{-1}\, y_k = y_{k-1}$ (and q is the advance operator).

- The name autoregression becomes more obvious when we rewrite this as

$$y_k = -A_1 y_{k-1} - \cdots - A_n y_{k-n} + e_k$$

which expresses y_k as a linear regression on its own past plus independent noise.

Autoregressive (AR) Processes (2)

- The impulse response of the system satisfies the following recursion:

$$\mathrm{A}(z)\,\mathrm{H}(z) \;=\; 1 \quad\Rightarrow\quad \mathrm{A}(q)\,h_k \;=\; \delta_{k0}$$

  This allows one to find h_k recursively from $\mathrm{A}(z)$. In particular $h_0 = 1$.

- From the expression for the psdf, we can find

$$\mathrm{S}_{yy}(z) = \frac{\sigma_e^2}{\mathrm{A}(z)\mathrm{A}(1/z)} \quad\Rightarrow\quad \mathrm{A}(z)\,\mathrm{S}_{yy}(z) = \sigma_e^2\,\mathrm{H}(1/z) \quad \text{or} \quad \mathrm{A}(q)\,r_{yy}(k) = \sigma_e^2\,h_{-k}$$

  which are the so-called *Yule-Walker equations*.
  The Yule-Walker equations for $k = 0, 1, \ldots, n$ constitute $n{+}1$ linear equations that allow one to obtain $r_{yy}(0), \ldots, r_{yy}(n)$ from $\sigma_e^2, A_1, \ldots, A_n$ or vice versa:

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(1) & \cdots & r_{yy}(n) \\ r_{yy}(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{yy}(1) \\ r_{yy}(n) & \cdots & r_{yy}(1) & r_{yy}(0) \end{bmatrix} \begin{bmatrix} 1 \\ A_1 \\ \vdots \\ A_n \end{bmatrix} = \begin{bmatrix} \sigma_e^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

  If for example the $r_{yy}(0), \ldots, r_{yy}(n)$ have been obtained from $\sigma_e^2, A_1, \ldots, A_n$, then further Yule-Walker equations can be used to obtain the rest of the covariances recursively: $\quad \mathrm{A}(q)\,r_{yy}(k) \;=\; 0\,,\;\; k > n$

Moving Average (MA) Processes

- A Moving Average process of order m (MA(m)) is obtained by taking a m-th order *all-zero* transfer function

  $$\mathrm{H}(z) \;=\; \mathrm{B}(z) \;=\; \sum_{i=0}^{m} B_i\, z^{-i} \;, \quad B_0 = 1 \;.$$

  Again, $\mathrm{B}(z)$ is a monic polynomial in z^{-1}. $\mathrm{B}(z)$ as a function of z needs again to have all its roots inside the unit circle for $\mathrm{H}(z)$ to be minimum-phase.

- The input-output relation is described by the following difference equation

  $$y_k \;=\; \mathrm{B}(q)\, e_k \;=\; e_k + B_1 e_{k-1} + \cdots + B_m e_{k-m} \;.$$

  The name moving average stems from the fact that y_k is computed as a sliding (moving) weighted linear combination (average) of the $m{+}1$ last inputs.

Moving Average (MA) Processes (2)

- We get for the psdf
$$S_{yy}(z) \;=\; \sigma_e^2\,\mathrm{B}(z)\mathrm{B}(1/z)$$
or in the time domain
$$r_{yy}(k) \;=\; \sigma_e^2\,B_k * B_{-k}$$
which implies in particular that
$$r_{yy}(k) \;=\; 0\;,\;\; |k|>m\;.$$

- Due to the particular form of $\mathrm{S}_{yy}(z)$, $\mathrm{S}_{yy}(z)$ has precisely $2m$ zeros which are such that if z_i is a zero, then so is $1/z_i$. Hence, σ_e^2 and $\mathrm{B}(z)$ can be identified from the $r_{yy}(k)$ by finding the zeros of $\mathrm{S}_{yy}(z)$ and assigning the minimum-phase zeros ($|z_i| \leq 1$) to $\mathrm{B}(z)$, and $\sigma_e^2 = \mathrm{S}_{yy}(1)/\mathrm{B}^2(1)$. This process is called *spectral factorization.*

- Remark that the Blackman-Tukey spectral estimator implictly assumes a MA(M) process since the weighted acf is put equal to zero for lags bigger than M.

Autoregressive Moving Average (ARMA) Processes

- An autoregressive moving average (ARMA(n,m)) process is obtained by taking a rational transfer function

  $$\mathrm{H}(z) \;=\; \mathrm{B}(z)/\mathrm{A}(z)$$

  where $\mathrm{A}(z)$ and $\mathrm{B}(z)$ are monic minimum-phase polynomials as before.

- The input-output relation is described by the following difference equation

  $$\mathrm{A}(q)\,y_k \;=\; \mathrm{B}(q)\,e_k \;\; \text{or} \;\; y_k+A_1y_{k-1}+\cdots+A_ny_{k-n} \;=\; e_k+B_1e_{k-1}+\cdots+B_me_{k-m} \;.$$

  This process is clearly a combination of autoregression and moving average.

- The impulse response of the system satisfies the following recursion:

  $$\mathrm{A}(z)\,\mathrm{H}(z) \;=\; \mathrm{B}(z) \;\;\Rightarrow\;\; \mathrm{A}(q)\,h_k \;=\; B_k \;.$$

  This allows one to find h_k recursively from $\mathrm{A}(z)$ and $\mathrm{B}(z)$. In particular $h_0 = 1$.

Autoregressive Moving Average (ARMA) Processes (2)

- From the expression for the psdf, we can find

$$\mathrm{S}_{yy}(z) \;=\; \sigma_e^2\,\frac{\mathrm{B}(z)\mathrm{B}(1/z)}{\mathrm{A}(z)\mathrm{A}(1/z)}$$

$$\Rightarrow \quad \mathrm{A}(z)\,\mathrm{S}_{yy}(z) \;=\; \sigma_e^2\,\mathrm{B}(z)\mathrm{H}(1/z) \quad \text{or} \quad \mathrm{A}(q)\,r_{yy}(k) \;=\; \sigma_e^2\,B_k * h_{-k}$$

which are again the *Yule-Walker equations*.

- Given σ_e^2, $\mathrm{A}(z)$ and $\mathrm{B}(z)$, one can obtain the acf from the Yule-Walker equations in pretty much the same way as for an AR process. Given the acf, one can determine $\mathrm{A}(z)$ from n equations of the form

$$\mathrm{A}(q)\,r_{yy}(k) \;=\; 0\,,\; k > m\;.$$

σ_e^2 and $\mathrm{B}(z)$ can then be obtained by spectral factorization of $\mathrm{A}(z)\,\mathrm{A}(1/z)\,\mathrm{S}_{yy}(z)$.

(Forward) Linear Prediction

- Consider predicting the sample y_k (WSS process) linearly from the n previous samples:

  $$\text{prediction} \quad \widehat{y}_k \;=\; -\sum_{i=1}^{n} A_{n,i}\, y_{k-i} \quad \text{linear combination}$$

  double index of $A_{n,i}$: they will depend on the total number of previous samples n involved in the prediction.

- We shall adjust the coefficients $A_{n,i}$ to minimize the prediction error $y_k - \widehat{y}_k$. Of course, for a given sample y_k, it is always possible to find coefficients $A_{n,i}$ such that the prediction error is zero! However, we don't want to choose totally different coefficients for each sample, because then our coefficients would simply be a nonunique nonlinear transformation of our signal and they would not extract any important characteristic of our signal. We want to minizes the prediction error, not instantaneously, but on the average (LMMSE !).

- So we shall minimize the prediction error variance (MSE):

  $$\min_{A_{n,i}} \|y_k - \widehat{y}_k\|^2 \;=\; \min_{A_{n,i}} E\,(y_k - \widehat{y}_k)^2 \;=\; \min_{A_{n,i}} E\left(y_k + \sum_{i=1}^{n} A_{n,i}\, y_{k-i}\right)^2$$

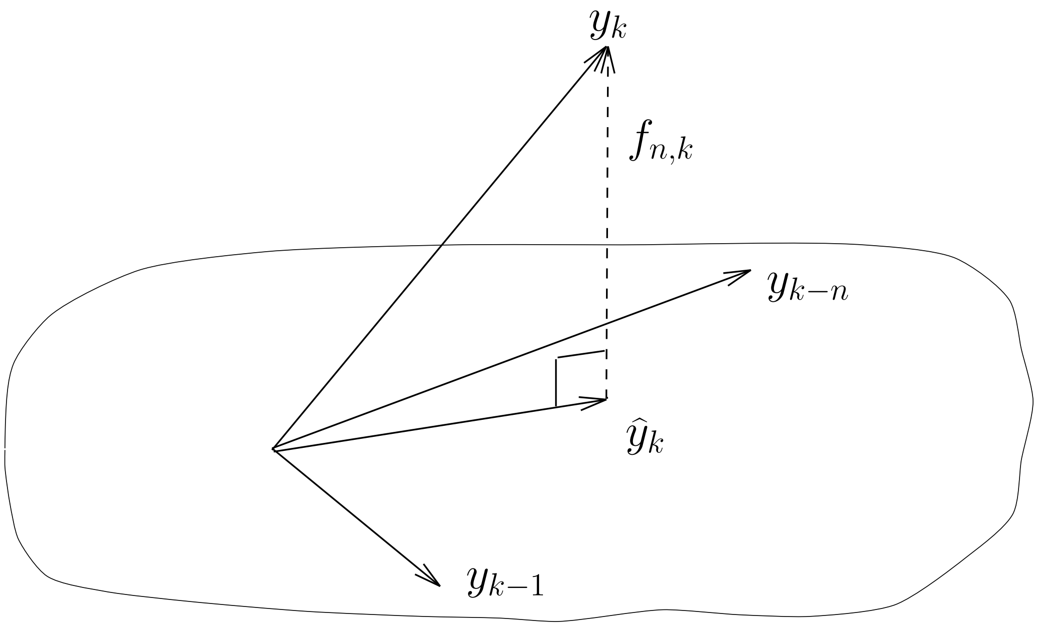
(Forward) Linear Prediction (2)

- Let us introduce the following notation for this forward prediction error of order n at time instant k

$$f_{n,k} \;=\; y_k - \widehat{y}_k \;=\; y_k + \sum_{i=1}^{n} A_{n,i} y_{k-i} \;=\; \sum_{i=0}^{n} A_{n,i} y_{k-i}$$

  where we introduced $A_{n,0} = 1$.

- the solution to the least-squares problem is characterized by the orthogonality condition: the point $\widehat{y}_k$ in the subspace spanned by $y_{k-1}, \ldots, y_{k-n}$ that is closest to y_k is the one that is the orthogonal projection of y_k onto that subspace.



(Forward) Linear Prediction (3)

- The orthogonality conditions can be written as

$$\langle f_{n,k}, y_{k-i}\rangle = Ef_{n,k}y_{k-i} = \sum_{j=0}^{n} A_{n,j}Ey_{k-i}y_{k-j} = \sum_{j=0}^{n} r_{|i-j|}A_{n,j} = 0\ ,\ i = 1,\ldots,n$$

where $r_{|i-j|} = Ey_{k-i}y_{k-j} = r_{yy}(|i-j|)$. Stationarity $\Rightarrow A_{n,i}$ time invariant (no k).

- Minimal value of the criterion:

$$\sigma^2_{f,n} = E\,f^2_{n,k} = \min_{A_{n,i}} Ef_{n,k}(y_k+\sum_{i=1}^{n} A_{n,i}y_{k-i}) = Ef_{n,k}y_k+\sum_{j=1}^{n} A_{n,j}\underbrace{Ef_{n,k}y_{k-j}}_{=0} = Ef_{n,k}y_k$$

- Introduce : $Y_{n+1}(k) = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-n} \end{bmatrix}\ ,\ \ A_n = \begin{bmatrix} 1 \\ A_{n,1} \\ \vdots \\ A_{n,n} \end{bmatrix}\ ,\ \ \Rightarrow\ \ f_{n,k} = Y^T_{n+1}(k)A_n$

- Assembling the orthogonality conditions with the expression for the minimal variance:

$$E\,Y_{n+1}(k)f_{n,k}\ = \begin{bmatrix} Ey_kf_{n,k} \\ Ey_{k-1}f_{n,k} \\ \vdots \\ Ey_{k-n}f_{n,k} \end{bmatrix} = \begin{bmatrix} \sigma^2_{f,n} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

(Forward) Linear Prediction (4)

- On the other hand

$$E\,Y_{n+1}(k)f_{n,k} \;=\; \left(E\,Y_{n+1}(k)Y^T_{n+1}(k)\right)A_n \;=\; R_{n+1}A_n \;=\; \begin{bmatrix} \sigma^2_{f,n} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\Rightarrow$ *normal/Yule-Walker equations*, where

$$R_{n+1} \;=\; E\,Y_{n+1}(k)Y^T_{n+1}(k) \;=\; \begin{bmatrix} r_0 & r_1 & \cdots & r_n \\ r_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_n & \cdots & r_1 & r_0 \end{bmatrix}$$

Due to the stationarity of y_k, R_{n+1} is Toeplitz.

- The YW equations are a bit unusual in that they are $n{+}1$ equations in $n{+}1$ unknowns, but n unknowns are on the LHS , while 1 unknown ($\sigma^2_{f,n}$) is on the RHS . One solves the last n equations for the n unknowns $A_{n,1},\ldots,A_{n,n}$, which then get substituted in the first equation to find $\sigma^2_{f,n}$.

- To solve a system of n equations in n unknowns takes on the order of n^3 operations (multplications, additions) in general.

Backward Linear Prediction

- Fast algorithms for solving the normal equations make use of the so-called backward prediction problem.

- Consider now the sense of the time axis as going backward in time, but we shall still work with the $n+1$ most recent samples of y_k. So consider the problem of linearly predicting y_{k-n} backward, i.e. from the n samples that come immediately afterward:

$$\widehat{y}_{k-n} \;=\; -\sum_{i=1}^{n} B_{n,i} y_{k-n+i}$$

- We want again to adjust the backward prediction coefficients $B_{n,i}$ to minimize the prediction error variance:

$$\min_{B_{n,i}} \|y_{k-n} - \widehat{y}_{k-n}\|^2 \;=\; \min_{B_{n,i}} E\left(y_{k-n} - \widehat{y}_{k-n}\right)^2 \;=\; \min_{B_{n,i}} E\left(y_{k-n} + \sum_{i=1}^{n} B_{n,i} y_{k-n+i}\right)^2$$

- notation: backward prediction error of order n at time instant k

$$b_{n,k} \;=\; y_{k-n} - \widehat{y}_{k-n} \;=\; y_{k-n} + \sum_{i=1}^{n} B_{n,i} y_{k-n+i} \;=\; \sum_{i=0}^{n} B_{n,i} y_{k-n+i}$$

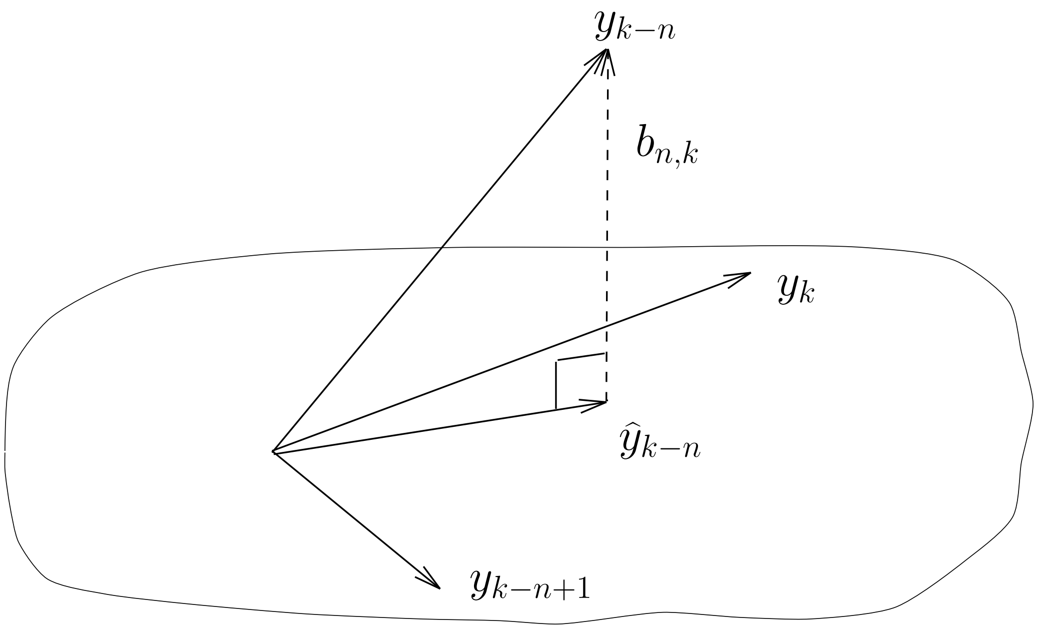
where $B_{n,0} = 1$.

Backward Linear Prediction (2)

- The solution to the least-squares problem is again characterized by the orthogonality condition: the point $\widehat{y}_{k-n}$ in the subspace spanned by $y_k, \ldots, y_{k-n+1}$ that is closest to y_{k-n} is the one that is the orthogonal projection of y_{k-n} onto that subspace.
- The orthogonality conditions can be written for $i = 1, \ldots, n$

$$\langle b_{n,k}, y_{k-n+i} \rangle = E b_{n,k} y_{k-n+i} = \sum_{j=0}^{n} B_{n,j} E y_{k-n+i} y_{k-n+j} = \sum_{j=0}^{n} r_{|i-j|} B_{n,j} = 0$$

Again the optimal prediction coefficients are constant (as a function of time).



Backward Linear Prediction (3)

- Minimal value of the criterion:

$$\sigma^2_{b,n} = E b^2_{n,k} = \min_{B_{n,i}} E b_{n,k}\Big(y_{k-n} + \sum_{i=1}^{n} B_{n,i} y_{k-n+i}\Big) = E b_{n,k} y_{k-n} + \sum_{j=1}^{n} B_{n,j} \underbrace{E b_{n,k} y_{k-n+j}}_{=0} = E b_{n,k} y_{k-n}$$

- Assembling the orthogonality conditions with the expression for the minimal variance:

$$E Y_{n+1}(k) b_{n,k} \; = \; \begin{bmatrix} E y_k b_{n,k} \\ \vdots \\ E y_{k-n+1} b_{n,k} \\ E y_{k-n} b_{n,k} \end{bmatrix} \; = \; \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma^2_{b,n} \end{bmatrix}$$

- On the other hand, $b_{n,k} = Y^T_{n+1}(k) B_n$ where $B_n \; = \; \begin{bmatrix} B_{n,n} \\ \vdots \\ B_{n,1} \\ 1 \end{bmatrix}$

- So we get the normal equations for the backward prediction problem

$$E\,Y_{n+1}(k) b_{n,k} \; = \; \big(E\,Y_{n+1}(k) Y^T_{n+1}(k)\big)\, B_n \; = \; R_{n+1} B_n \; = \; \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma^2_{b,n} \end{bmatrix}$$

Forward vs Backward Prediction Quantities

- reverse identity matrix J

$$J = \begin{bmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{bmatrix} , \quad J \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_n \\ \vdots \\ y_2 \\ y_1 \end{bmatrix} , \quad [y_1 \;\; y_2 \cdots y_n] \; J \; = \; [y_n \cdots y_2 \;\; y_1]$$

- A symmetric Toeplitz matrix is *persymmetric* (symmetric w.r.t. the antidiagonal)

$$J \, R_{n+1} \, J \; = \; R_{n+1}^T \; = \; R_{n+1}$$

where the second identity implies that R_{n+1} is also *centrosymmetric* (symmetric and persymmetric).

- the backward normal equations lead to the forward normal equations

$$R_{n+1} \, J B_n \; = \; J \, R_{n+1} \, J \, J B_n \; = \; J \, R_{n+1} B_n \; = \; J \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma_{b,n}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{b,n}^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$B_{n,i}$ and $\sigma_{b,n}^2$ satisfy exactly the same equations as the $A_{n,i}$ and $\sigma_{f,n}^2$ and hence $B_n \; = \; J \, A_n$ (or $B_{n,i} = A_{n,i}$) , $\sigma_{b,n}^2 = \sigma_{f,n}^2$

Levinson Algorithm

Goal: fast algorithm for solving the normal equations. The algorithm is recursive in nature. So suppose after recursion n we have A_n and $\sigma^2_{f,n}$. We now try to find the same quantities for order $n{+}1$. We first look at a trial solution for A_{n+1} which we obtain by appending one zero to A_n :

$$\underbrace{\begin{bmatrix} r_0 & r_1 & \cdots & r_n & r_{n+1} \\ r_1 & \ddots & \ddots & & r_n \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r_n & & \ddots & \ddots & r_1 \\ r_{n+1} & r_n & \cdots & r_1 & r_0 \end{bmatrix}}_{R_{n+2}} \underbrace{\left\{\left(\begin{bmatrix}1\\A_{n,1}\\ \vdots \\ A_{n,n} \\ 0\end{bmatrix} + K_{n+1} \begin{bmatrix}0\\A_{n,n}\\ \vdots\\A_{n,1}\\1\end{bmatrix}\right)\right\}}_{A_{n+1}} = \underbrace{\left\{\left(\begin{bmatrix}\sigma^2_{f,n}\\0\\ \vdots\\0\\ \Delta_{n+1}\end{bmatrix} + K_{n+1}\begin{bmatrix}\Delta_{n+1}\\0\\ \vdots\\0\\ \sigma^2_{f,n}\end{bmatrix}\right)\right\}}_{[\sigma^2_{f,n+1}\ 0 \cdots 0\ 0]^T}$$

Because R_{n+2} contains R_{n+1} as its upper-left submatrix of one dimension less, and because of the form of our trial solution, the corresponding RHS has the desired form except for the last entry, call it Δ_{n+1}. If we flip our trial solution upside down (the corresponding trial solution for the backward prediction problem), then the RHS also simply gets flipped upside down, because of the centrosymmetry of R_{n+2}. By linearity, a linear combination of the two trial solutions gives the same linear combination of the two RHS 's. We can choose K_{n+1} to get zero for the last element of the RHS .

Levinson Algorithm (2)

- So we should choose

$$\Delta_{n+1} + K_{n+1}\,\sigma^2_{f,n} = 0 \quad \Rightarrow \quad K_{n+1} \;=\; -\frac{\Delta_{n+1}}{\sigma^2_{f,n}}$$

- Now it becomes clear that the combination of the two trial solutions has itself the right structure (1 as first element) and when multiplied by R_{n+2} gives a right hand side that has the right structure. Hence

$$A_{n+1} \;=\; (I + K_{n+1}\,J)\begin{bmatrix} A_n \\ 0 \end{bmatrix}$$

  and in particular, $A_{n+1,n+1} = K_{n+1}$.

- Since we have found A_{n+1}, the top element of the RHS must be $\sigma^2_{f,n+1}$. Hence,

$$\underbrace{\sigma^2_{f,n+1}}_{\geq 0} \;=\; \sigma^2_{f,n} + K_{n+1}\Delta_{n+1} \;=\; \underbrace{\sigma^2_{f,n}}_{>0}\underbrace{\left(1 - K^2_{n+1}\right)}_{\geq 0}$$

  from which it follows that

$$|K_{n+1}| \;\leq\; 1$$

Levinson Algorithm (4)

- Levinson algorithm: $\begin{cases} A_n \\ \sigma^2_{f,n} \end{cases} \Rightarrow \begin{cases} A_{n+1} \\ \sigma^2_{f,n+1} \end{cases}$

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

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

- Per recursion, the Levinson algorithm needs about $2n$ multiplications and a similar amount of additions. So when the algorithm is run up to some full order N, the total computational complexity is

$$\sum_{n=1}^{N} 2n = \frac{2N(N+1)}{2} \approx N^2$$

So we really have a fast algorithm for finding A_N.

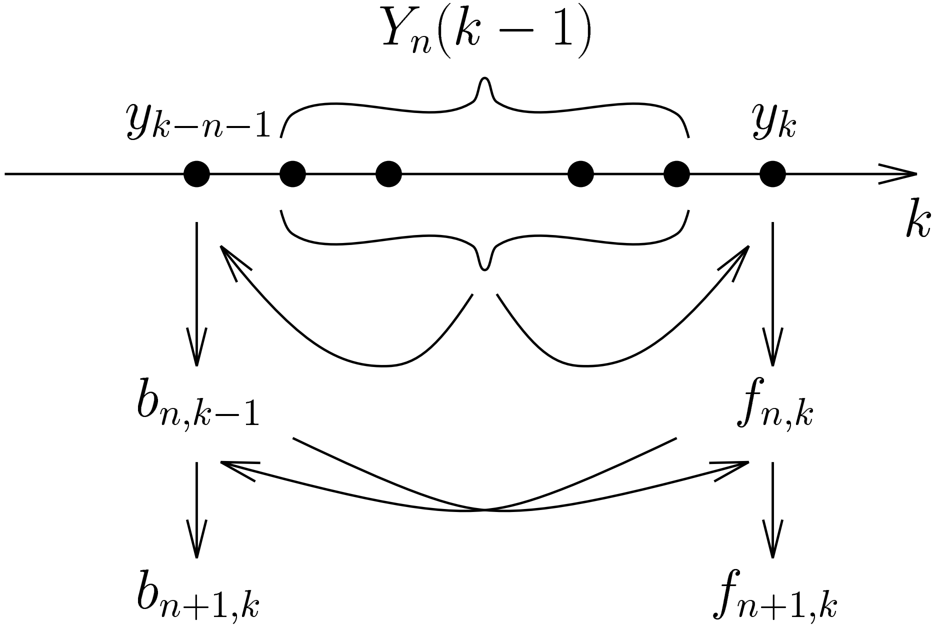
Levinson Algorithm (5)

- The coefficients $-K_{n+1}$ have an interpretation as correlation coefficients:

$$-K_{n+1} = \frac{\Delta_{n+1}}{\sigma^2_{f,n}} = \sigma^{-2}_{f,n}[\sigma^2_{f,n}\,0\cdots0\,\Delta_{n+1}][0\,A_{n,n}\cdots A_{n,1}\,1]^T = \sigma^{-2}_{f,n}\,[A_n^T\;0]\,R_{n+2}\begin{bmatrix}0\\ J\,A_n\end{bmatrix}$$

$$= \frac{EA_n^TY_{n+1}(k)Y^T_{n+1}(k{-}1)B_n}{\sigma_{f,n}\;\sigma_{b,n}} = \frac{Ef_{n,k}b_{n,k-1}}{\sqrt{Ef^2_{n,k}}\,\sqrt{Eb^2_{n,k-1}}} = \frac{\langle f_{n,k}, b_{n,k-1}\rangle}{\|f_{n,k}\|\;\;\|b_{n,k-1}\|}$$

This coefficient is in fact called *Partial Correlation* (PARCOR) coefficient because it describes the partial correlation between y_k and y_{k-n-1}, partial because the influence of $Y_n(k{-}1)$ in between those two is removed.



Levinson Algorithm (6)

- When we apply the Cauchy-Schwarz formula, then we find immediately $|K_{n+1}| \leq 1$ back.

- prediction filters: consider the z-transforms of the prediction error filter impulse responses:
$$[\mathrm{A}_n(z) \;\; \mathrm{B}_n(z)] \;=\; \left[1 \;\; z^{-1} \cdots z^{-n}\right] [A_n \;\; B_n]$$
The property $B_n = J\,A_n$ translates to $\;\mathrm{B}_n(z) \;=\; z^{-n}\,\mathrm{A}_n(z^{-1}) \;=\; z^{-n}\,\mathrm{A}_n^{\dagger}(z)$

- The positive definiteness of R_{n+1} has as a consequence that the filter $\mathrm{A}_n(z)$ is minimum-phase, i.e. has all its zeros inside the unit circle (on the unit circle when R_{n+1} is singular). This implies in particular that $1/\mathrm{A}_n(z)$ is guaranteed to be an exponentially stable filter.

Levinson Algorithm (7)

- The positive definiteness of R_{n+1} and the minimum-phase property of the filter $\mathrm{A}_n(z)$ are also related to the boundedness of the PARCORs. In fact, we have the following property.

  Schur-Cohn Test: A polynomial $\mathrm{A}_N(z)$ is minimum-phase if and only if the sequence of PARCORs is bounded: $|K_n| < 1,\ n = N, N{-}1, \ldots, 1.$

- In order to be able to apply this test, we have to know how to find the PARCORs from the filter $\mathrm{A}_n(z)$. Whereas the Levinson algorithm is essentially the following *step-up* procedure:

$$A_{n+1} \ = \left[\begin{array}{c} A_n \\ 0 \end{array}\right] \ + \ K_{n+1} \left[\begin{array}{c} 0 \\ J\,A_n \end{array}\right] \ = \ (I + K_{n+1}\,J) \left[\begin{array}{c} A_n \\ 0 \end{array}\right]$$

this procedure can be inverted to yield the following *step-down* procedure:

$$\left[\begin{array}{c} A_n \\ 0 \end{array}\right] \ = \ (I + K_{n+1}\,J)^{-1}\ A_{n+1} \ = \ \frac{1}{1 - K_{n+1}^2} (I - K_{n+1}\,J)\ A_{n+1}\ , \quad K_{n+1} = A_{n+1,n+1}$$

which can be reiterated to yield all PARCORs, starting from the highest order polynomial.

Levinson Algorithm (8)

- the complete set of equivalences is

$$\begin{tikzcd}
\boxed{R_{N+1} > 0} \arrow[r, shift left, "(a)"] \arrow[d, shift right, "(c)"'] & \boxed{\begin{array}{c} \sigma^2_{f,n} > 0 \\ n = 0, 1, \ldots, N \end{array}} \arrow[l, shift left] \arrow[d, shift right] \\
\boxed{\begin{array}{c} \mathrm{A}_N(z) \text{ minimum phase} \\ \sigma^2_{f,N} > 0 \end{array}} \arrow[u, shift right] \arrow[r, shift left] & \boxed{\begin{array}{c} r_0 = \sigma^2_{f,0} > 0\,, \quad |K_n| < 1 \\ n = N, N{-}1, \ldots, 1 \end{array}} \arrow[l, shift left, "(d)"] \arrow[u, shift right, "(b)"']
\end{tikzcd}$$

- The Schur-Cohn test, which is a subset of equivalence (d), can perhaps most easily be shown by using Rouché's theorem of complex variable theory.

- We mentioned equivalence (c) before.

- Equivalence (b) follows straightforwardly from $\sigma^2_{f,n} = \sigma^2_{f,n-1}(1 - K_n^2)$.

- Equivalence (a) follows from the triangular factorization of R_n^{-1} interpretation of linear prediction.

Lattice Filters

- We can write out the step-up (Levinson) procedure jointly for forward and backward prediction error filters (using $B_{n+1} = J\, A_{n+1}$):

$$\begin{aligned} A_{n+1} &= \begin{bmatrix} A_n \\ 0 \end{bmatrix} + K_{n+1} \begin{bmatrix} 0 \\ B_n \end{bmatrix} \\ B_{n+1} &= \begin{bmatrix} 0 \\ B_n \end{bmatrix} + K_{n+1} \begin{bmatrix} A_n \\ 0 \end{bmatrix} \end{aligned}$$

- If we multiply all sides with $[1 \;\; z^{-1} \;\; \cdots \;\; z^{-n-1}]$, we get

$$\begin{aligned} \mathrm{A}_{n+1}(z) &= \mathrm{A}_n(z) + K_{n+1}\, z^{-1}\, \mathrm{B}_n(z) \\ \mathrm{B}_{n+1}(z) &= K_{n+1} \mathrm{A}_n(z) + z^{-1}\, \mathrm{B}_n(z) \end{aligned}$$

which can be rewritten as

$$\begin{bmatrix} \mathrm{A}_{n+1}(z) \\ \mathrm{B}_{n+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{n+1} \\ K_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{A}_n(z) \\ \mathrm{B}_n(z) \end{bmatrix} , \quad \begin{bmatrix} \mathrm{A}_0(z) \\ \mathrm{B}_0(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

This formula describes one lattice section. From this formula, it is straightforward to draw the realization of the complete lattice filter, by cascading lattice sections and taking the proper initialization into account.

Lattice Filters (1)

- By multiplying

$$\begin{bmatrix} \mathrm{A}_{n+1}(z) \\ \mathrm{B}_{n+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{n+1} \\ K_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{A}_{n}(z) \\ \mathrm{B}_{n}(z) \end{bmatrix} , \quad \begin{bmatrix} \mathrm{A}_{0}(z) \\ \mathrm{B}_{0}(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

with $\mathrm{Y}(z)$, the z-transform of y_k, we get

$$\begin{bmatrix} \mathrm{A}_{n+1}(z)\mathrm{Y}(z) \\ \mathrm{B}_{n+1}(z)\mathrm{Y}(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{n+1} \\ K_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{A}_{n}(z)\mathrm{Y}(z) \\ \mathrm{B}_{n}(z)\mathrm{Y}(z) \end{bmatrix} , \begin{bmatrix} \mathrm{A}_{0}(z)\mathrm{Y}(z) \\ \mathrm{B}_{0}(z)\mathrm{Y}(z) \end{bmatrix} = \begin{bmatrix} \mathrm{Y}(z) \\ \mathrm{Y}(z) \end{bmatrix} .$$

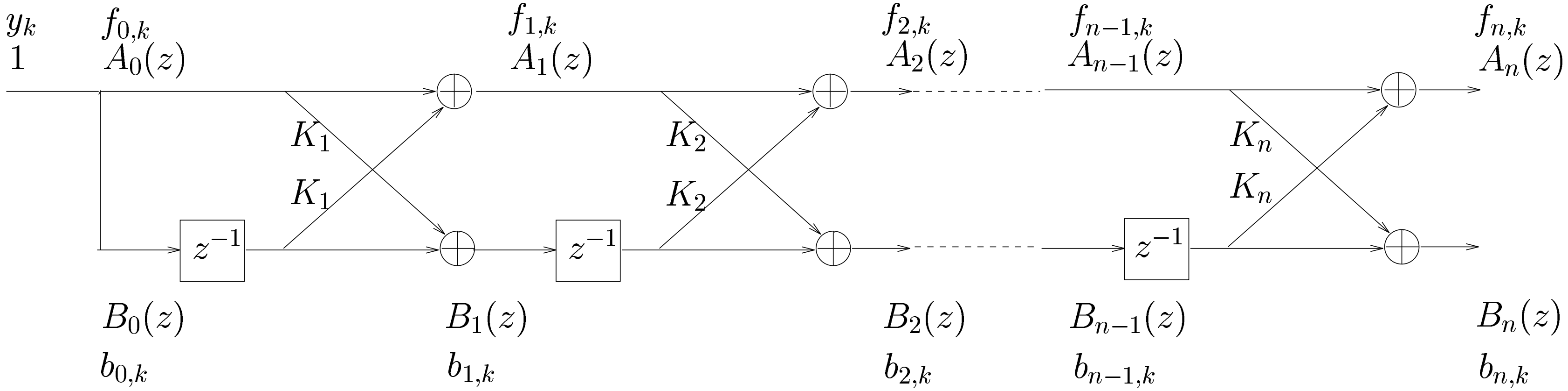
which can be rewritten in the time domain as

$$\begin{bmatrix} f_{n+1,k} \\ b_{n+1,k} \end{bmatrix} = \begin{bmatrix} 1 & K_{n+1} \\ K_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & q^{-1} \end{bmatrix} \begin{bmatrix} f_{n,k} \\ b_{n,k} \end{bmatrix} , \quad \begin{bmatrix} f_{0,k} \\ b_{0,k} \end{bmatrix} = \begin{bmatrix} y_k \\ y_k \end{bmatrix} .$$

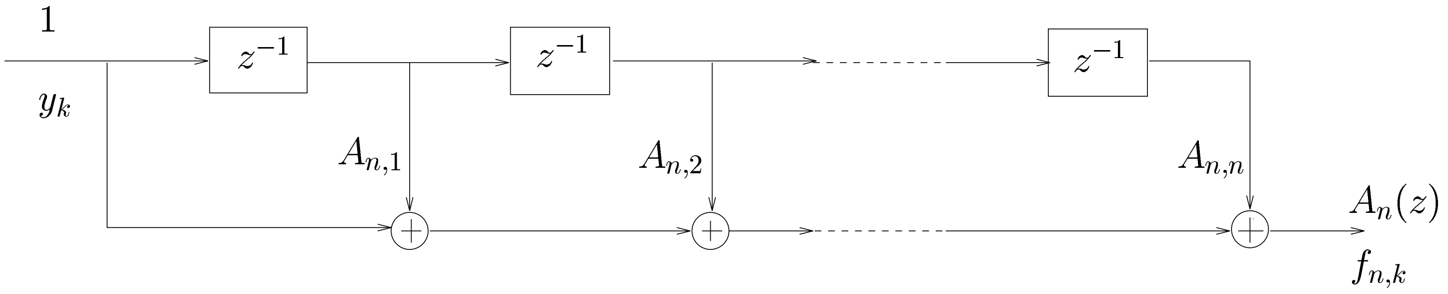
This formula describes one lattice section. From this formula, it is straightforward to draw the realization of the complete lattice filter, by cascading lattice sections and taking the proper initialization into account.

Lattice Filters (2)

- realization of forward and backward prediction via the analysis lattice filter



- equivalent "tapped delay line" or transversal filter realization of the FIR foward prediction error filter



Lattice Filters (3)

Advantages of the lattice filter realization.

- *order-recursiveness*: since we have that $f_{n,k} = \mathrm{A}_n(q)y_k$, $b_{n,k} = \mathrm{B}_n(q)y_k$, the prediction errors of all orders show up in the lattice filter if we excite it with the signal y_k.

- The lattice filter has some numerical advantages (especially in fixed-point implementation).

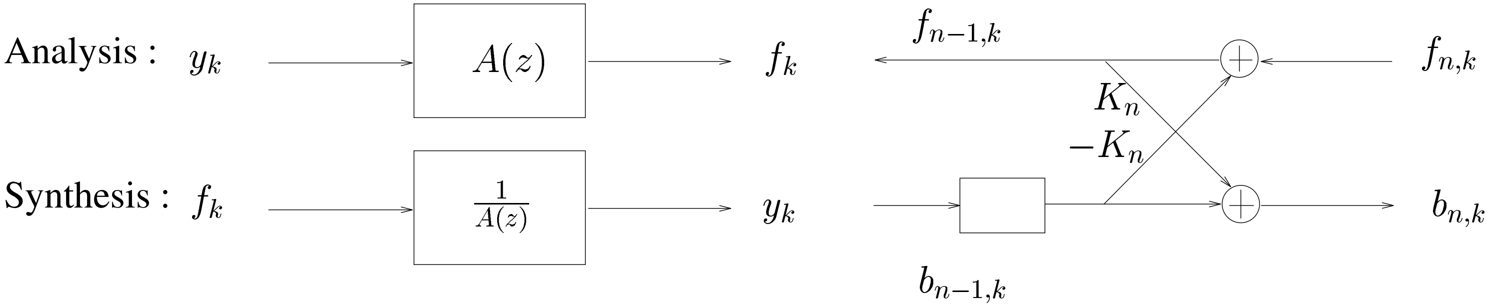
  1. the multipliers that appear in the lattice filter are the PARCORs which are bounded by 1 in magnitude: $|K_n| < 1$

  2. the various prediction errors have lower variance than the input signal: $\sigma^2_{f,n} \leq \sigma^2_{f,0} = \sigma^2_y$. Therefore, if the input signal is scaled to be in the interval $[-1,+1]$ (e.g. $\sigma_y < 0.25$ assuming Gaussian signal), then so will be all the signals appearing at all the internal nodes in the filter.

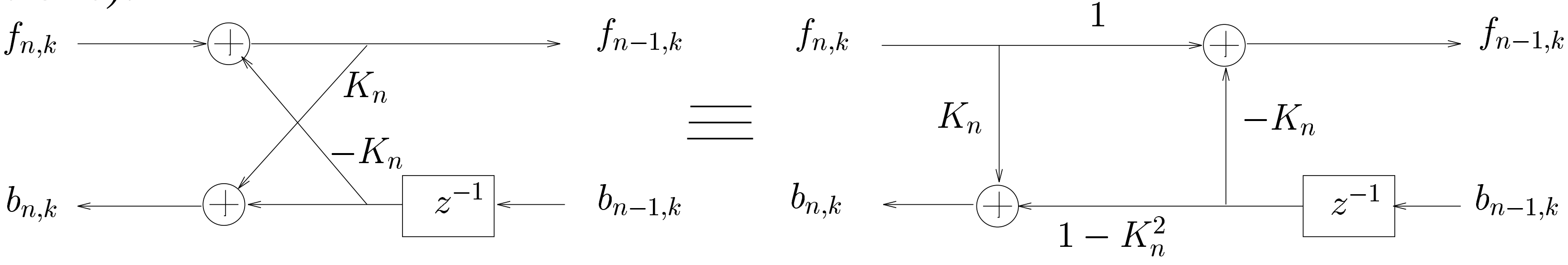
  3. The transfer function has also fairly low sensitivity to perturbations in the filter coefficients K_n.

Synthesis Lattice Filters

- Whereas for linear prediction we are interested in an analysis lattice filter that realizes $\mathrm{A}(z)$, for modeling we are interested in the synthesis lattice filter that realizes $1/\mathrm{A}(z)$. The synthesis lattice can be obtained from the analysis lattice by straightforward flowgraph manipulations. The roles of input and output get interchanged, we change the direction of the flow:

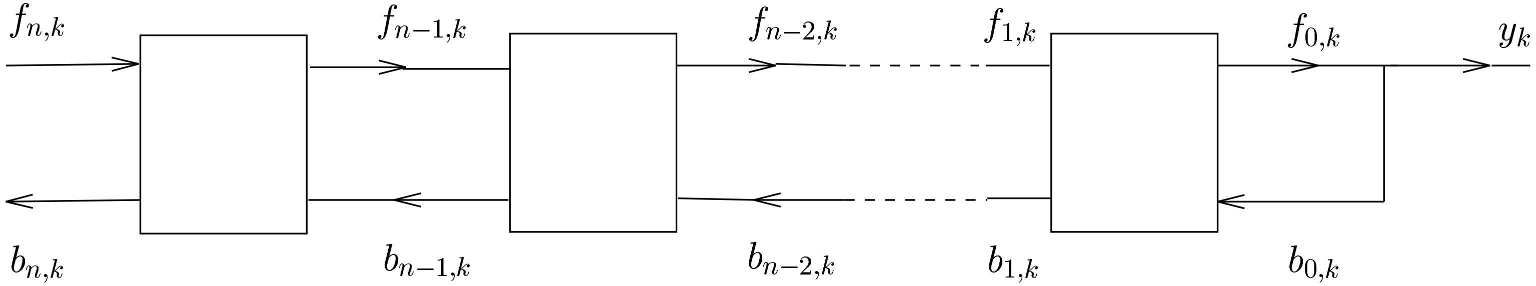


- Since it is usual to have the input on the left and the output on the right, we shall flip the above synthesis lattice section around. This yields the result below, which can also be transformed into the so-called 3-multiplier lattice section on the right (whereas the lattice sections considered so far are 2-multiplier sections):

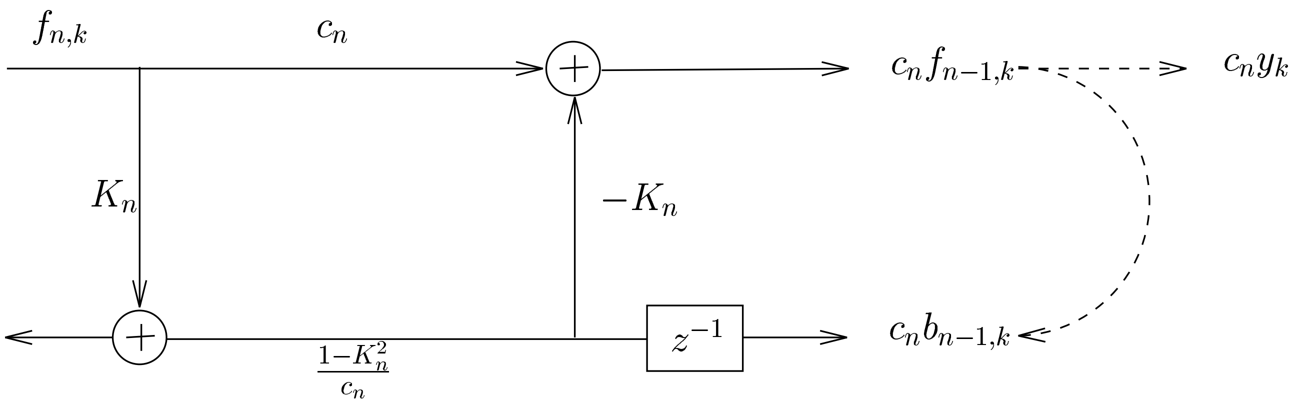


Synthesis Lattice Filters (2)

- general presentation of the all-pole IIR synthesis lattice filter:



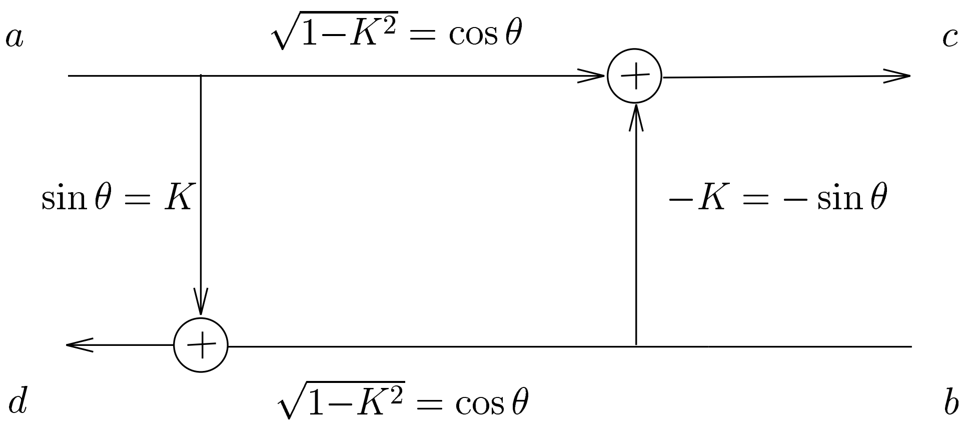
- *lattice section transformations*: suppose we factor the gain $1-K_n^2$ of the lower branch into two factors, one of which (c_n) is moved to the upper branch, then this will not change the loop gain of the feedback loop and so the dynamics remain unaltered. However, the net effect of such an operation is that all signals to the right are amplified by the factor c_n. As a result, the overall transfer function becomes $\left(\prod_{n=1}^{N} c_n\right) / \mathrm{A}_N(z)$.



Synthesis Lattice Filters (3)

- choice 1 : $c_n = \sqrt{1 - K_n^2} \quad \Rightarrow \quad$ 4-multiplier lattice or *normalized* lattice. This lattice has very good numerical properties since the input-output behavior of the static part of the lattice section is a 2×2 orthogonal rotation, which conserves energy. In a normalized lattice, all signals have the same variance.

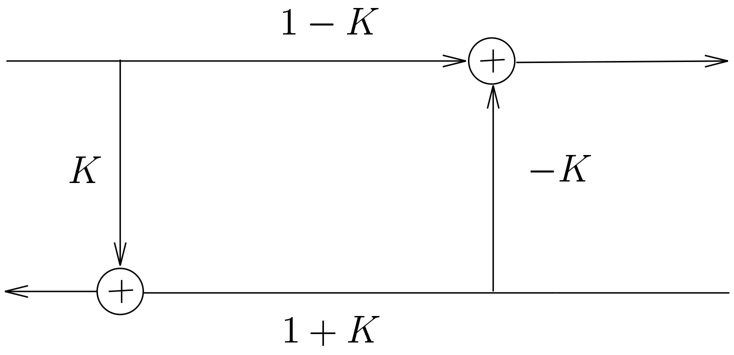
$c_n = \sqrt{1-k_n^2}$

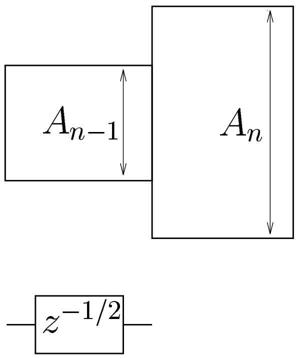


$$\begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad \Rightarrow \quad c^2 + d^2 \; = \; a^2 + b^2$$

Synthesis Lattice Filters (4)

- choice 2 : $c_n = 1 - K_n \quad \Rightarrow$ Kelly-Lochbaum lattice which corresponds exactly to a section of a transmission line. For this reason, the PARCORs are also called *reflection coefficients*. So there exists a close relationship between the all-pole synthesis lattice and a speech production model.





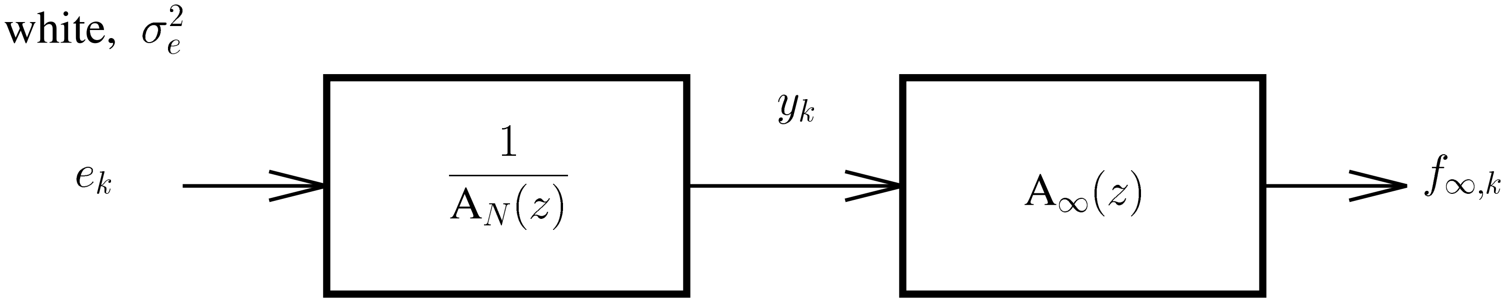
$$c_n = 1 - K_n$$

$$K_n = \frac{A_n - A_{n-1}}{A_n + A_{n-1}}$$

Linear Prediction of AR(N)

- normal equations of LP $\equiv$ Yule-Walker equations for AR(N) process $\Rightarrow$ since solution to linear equations unique, prediction error filter $\mathrm{A}_N(z)$ equals denominator of all-pole filter generating the AR process

- alternative point of view: for an AR process, minimizing prediction error variance leads to white prediction errors



white, σ_e^2

e_k

$\frac{1}{\mathrm{A}_N(z)}$

y_k

$\mathrm{A}_\infty(z)$

$f_{\infty,k}$

- overall transfer from white input e_k to prediction error $f_{\infty,k} = \mathrm{H}(q)\, e_k$

$$\mathrm{H}(z) = \frac{\mathrm{A}_\infty(z)}{\mathrm{A}_N(z)} = \sum_{i=0}^{\infty} h_i z^{-i} \quad , \quad h_0 = \mathrm{H}(\infty) = \frac{\mathrm{A}_\infty(\infty)}{\mathrm{A}_N(\infty)} = \frac{1}{1} = 1$$

($\frac{1}{\mathrm{A}_N(z)}$ is causal since $\mathrm{A}_N(z)$ is minimum-phase)

Linear Prediction of AR(N) (2)

- $$\begin{aligned}\sigma^2_{f,\infty} &= E\, f^2_{\infty,k} = E\left(\sum_{i=0}^{\infty} h_i e_{k-i}\right)^2 = E \sum_{i=0}^{\infty}\sum_{l=0}^{\infty} h_i h_l e_{k-i} e_{k-l} \\ &= \sum_{i=0}^{\infty}\sum_{l=0}^{\infty} h_i h_l r_{ee}(i-l) = \sigma_e^2 \sum_{i=0}^{\infty}\sum_{l=0}^{\infty} h_i h_l \delta_{il} = \sigma_e^2 \sum_{i=0}^{\infty} h_i^2 = \sigma_e^2\,(1+\sum_{i=1}^{\infty} h_i^2)\end{aligned}$$

- minimization: $\min\limits_{A_{\infty,i},i>0} \sigma^2_{f,\infty} = \min\limits_{h_i,i>0} \sigma^2_{f,\infty} = \sigma^2_e$ for $h_i = 0\,,\ i > 0\ .$

  Hence H$(z) = 1$ and A$_\infty(z) =$ A$_N(z)$ but also A$_n(z)\ =\ A_N(z)\ ,\quad n \geq N$

- So for an AR(N) process, we get

$$\begin{cases} K_n \ = \ A_{n,n} \ = \ -\dfrac{\Delta_n}{\sigma^2_{f,n-1}} \ = \ 0\ ,\ \ n > N \\ f_{n,k} \ = \ e_k \ = \ \text{white !}\ ,\ \ n \geq N\ . \end{cases}$$

- special case: white noise = AR(0).

  Hence A$_n(z) = 1,\ K_n = 0,\ \sigma^2_{f,n} = r_0,\ f_{n,k} = y_k,\ n \geq 0.$

  In particular also $\widehat{y}_{n,k} = y_k - f_{n,k} = 0$: white noise is unpredictible.

Linear Prediction Asymptotics

- *spectral factorization*: psdf $\mathrm{S}_{yy}(z)$ of a WSS process y_k can be factored as

$$\mathrm{S}_{yy}(z) \;=\; \mathrm{S}^{+}_{yy}(z)\,\mathrm{S}^{-}_{yy}(z) \;\;,\quad \mathrm{S}^{-}_{yy}(z) \;=\; \mathrm{S}^{+}_{yy}(1/z) \;=\; \mathrm{S}^{+\dagger}_{yy}(z)$$

$\mathrm{S}^{+}_{yy}(z)$ = the causal minimum-phase *spectral factor* of $\mathrm{S}_{yy}(z)$
$\mathrm{S}^{-}_{yy}(z)$ = the anticausal maximum-phase spectral factor of $\mathrm{S}_{yy}(z)$

- can interpret

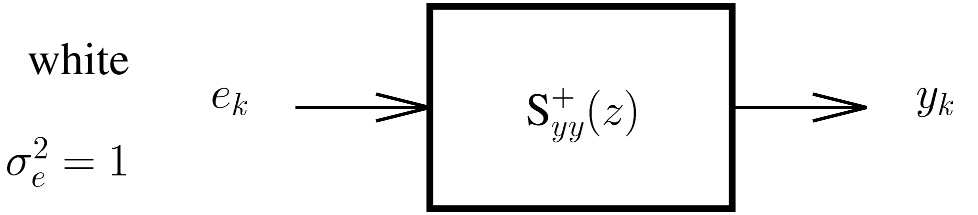
$$y_k \;=\; \mathrm{S}^{+}_{yy}(q)\; e_k \;,\;\; \sigma_e^2 = 1$$

where e_k = white noise

$$\Rightarrow\;\; \mathrm{S}_{yy}(z) \;=\; \mathrm{S}^{+}_{yy}(z)\mathrm{S}_{ee}(z)\mathrm{S}^{+}_{yy}(1/z) \;=\; \mathrm{S}^{+}_{yy}(z)\sigma_e^2\mathrm{S}^{+}_{yy}(1/z) \;=\; \mathrm{S}^{+}_{yy}(z)\mathrm{S}^{-}_{yy}(z)$$

Wold decomposition: any WSS process = MA(∞)

Linear Prediction Asymptotics (2)



white

$\sigma_e^2 = 1$

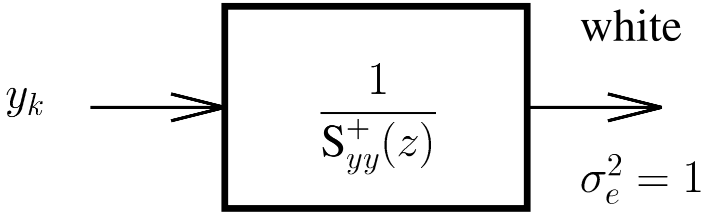
e_k

$\mathrm{S}_{yy}^{+}(z)$

y_k

Wold decomposition

WSS process = MA(∞)



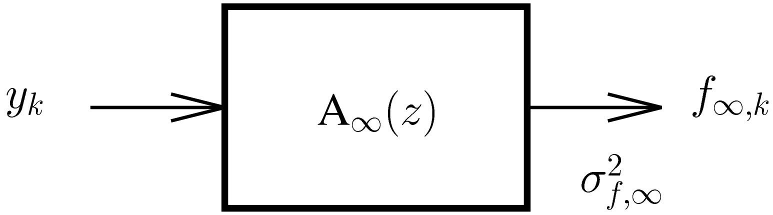
y_k

$\dfrac{1}{\mathrm{S}_{yy}^{+}(z)}$

white

e_k

$\sigma_e^2 = 1$

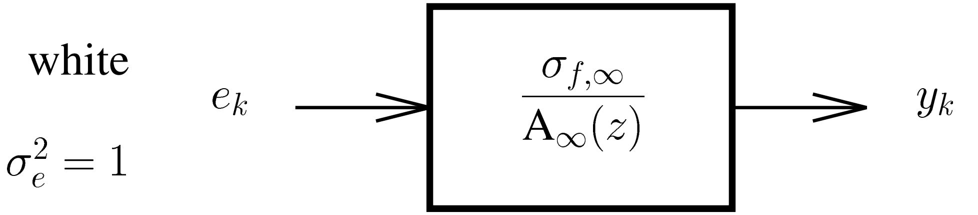


y_k

$\mathrm{A}_{\infty}(z)$

$f_{\infty,k}$

$\sigma_{f,\infty}^2$



white

$\sigma_e^2 = 1$

e_k

$\dfrac{\sigma_{f,\infty}}{\mathrm{A}_{\infty}(z)}$

y_k

Kolmogorov decomposition

WSS process = AR(∞)

Linear Prediction Asymptotics (3)

- consider LP(∞), vector spaces span $\{y_i,\ i \leq k\}$ and span $\{f_{\infty,i},\ i \leq k\}$

$$\begin{cases} f_{\infty,k} = \mathrm{A}_\infty(q)\, y_k\ ,\ \mathrm{A}_\infty(z) \text{ causal } \Rightarrow \text{ span}\,\{f_{\infty,i},\ i \leq k\} \subset \text{span}\,\{y_i,\ i \leq k\} \\ y_k = \dfrac{1}{\mathrm{A}_\infty(q)}\, f_{\infty,k}\ ,\ \dfrac{1}{\mathrm{A}_\infty(z)} \text{ causal } \Rightarrow \text{ span}\,\{f_{\infty,i},\ i \leq k\} \supset \text{span}\,\{y_i,\ i \leq k\} \end{cases}$$

$$\Rightarrow \quad \text{span}\,\{f_{\infty,i},\ i \leq k\} \; = \; \text{span}\,\{y_i,\ i \leq k\}$$

- Now, by the orthogonality condition of LMMSE estimation, we have

$$f_{\infty,k} \perp \text{span}\,\{y_i,\ i < k\} = \text{span}\,\{f_{\infty,i},\ i < k\} \quad \Rightarrow \quad f_{\infty,k} \perp \text{span}\,\{f_{\infty,i},\ i < k\}$$

which implies that $f_{\infty,k}$ is a white process! Hence

$$\sigma^2_{f,\infty} \; = \; \mathrm{S}_{f_\infty f_\infty}(z) \; = \; \mathrm{A}_\infty(z)\mathrm{S}_{yy}(z)\mathrm{A}_\infty(1/z) \;\Rightarrow\; \mathrm{S}_{yy}(z) \; = \; \frac{\sigma^2_{f,\infty}}{\mathrm{A}_\infty(z)\mathrm{A}_\infty(1/z)}$$

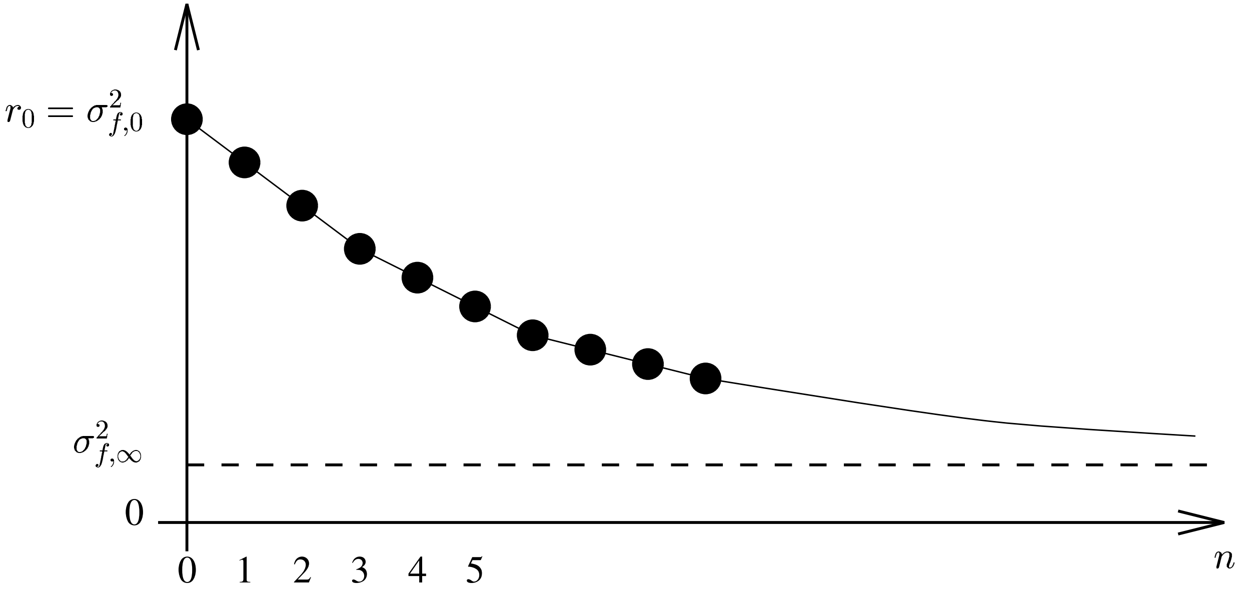
$$\Rightarrow\; \mathrm{S}^+_{yy}(z) \; = \; \frac{\sigma_{f,\infty}}{\mathrm{A}_\infty(z)}\ , \quad \mathrm{A}_\infty(\infty) \; = \; 1 \;\Rightarrow\; \sigma_{f,\infty} \; = \; \mathrm{S}^+_{yy}(\infty)\ , \quad \mathrm{A}_\infty(z) \; = \; \frac{\mathrm{S}^+_{yy}(\infty)}{\mathrm{S}^+_{yy}(z)}$$

- *Kolmogorov decomposition*: WSS process = AR(∞)

- in general: $\sigma^2_{f,\infty} \; = \; e^{\int_{-0.5}^{0.5} \ln S_{yy}(f) df}$ (> 0 for purely random processes)

AR Modeling via Linear Prediction

- $0 \leq \sigma^2_{f,n} = \sigma^2_{f,n-1}(1-K_n^2) \leq \sigma^2_{f,n-1} \;\; \Rightarrow \;\; \sigma^2_{f,n} \searrow \sigma^2_{f,\infty}$



- stationary segment of speech: $K_n \approx 0\,,\; n > 10 \;\Rightarrow\;$ the curve of $\sigma^2_{f,n}$ decreases rapidly at low orders, but starts to flatten out at n around 8 to 10: $\sigma^2_{f,10} \gtrapprox \sigma^2_{f,\infty}$.

- $$\begin{cases} f_{\infty,k} = \text{ white noise } \Rightarrow \text{ unpredictible} \\ f_{0,k} = y_k = \text{ very predictible if } \sigma^2_{f,\infty} \ll \sigma^2_{f,0} = r_0 \\ \text{if } \sigma^2_{f,N} \gtrapprox \sigma^2_{f,\infty} \text{ then } f_{N,k} \text{ no longer very predictible } \Rightarrow \approx \text{ white} \end{cases}$$

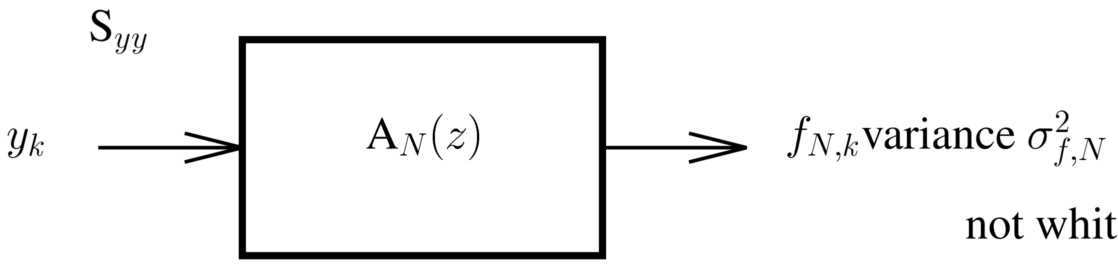
AR Modeling via Linear Prediction (2)

- autoregressive modeling: consider $f_{N,k} \equiv$ white with variance $\sigma^2_{f,N} = \widehat{\mathrm{S}}_{f_N f_N}(z)$

$$\widehat{\mathrm{S}}_{yy}(z) = \frac{\widehat{\mathrm{S}}_{f_N f_N}(z)}{\mathrm{A}_N(z)\mathrm{A}_N(1/z)} = \frac{\sigma^2_{f,N}}{\mathrm{A}_N(z)\mathrm{A}_N(1/z)} \approx \frac{\mathrm{S}_{f_N f_N}(z)}{\mathrm{A}_N(z)\mathrm{A}_N(1/z)} = \mathrm{S}_{yy}(z)$$

hat = approximation, not estimation

- If y_k = AR(n) $n \leq N$, then $f_{N,k}$ exactly white $\Rightarrow$ no approximation



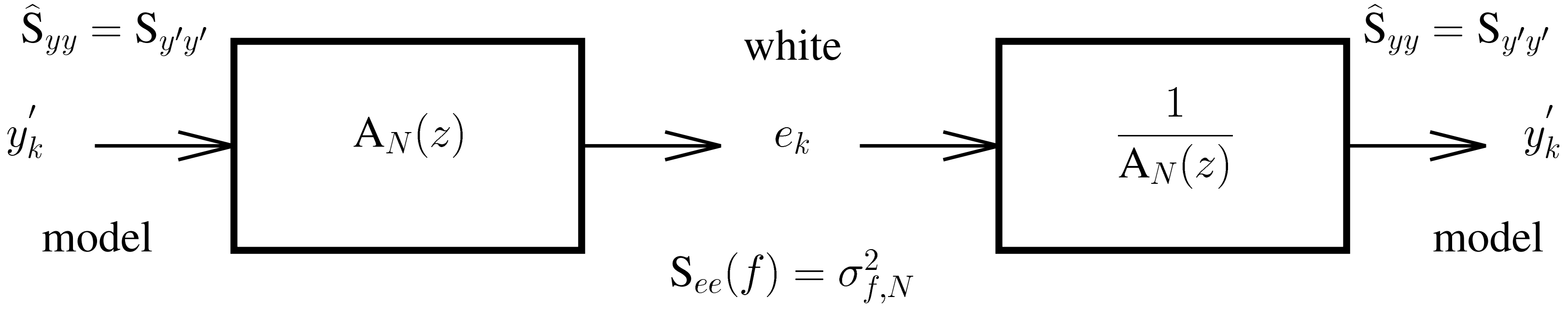
S_{yy}

y_k

$\mathrm{A}_N(z)$

$f_{N,k}$variance $\sigma^2_{f,N}$

not white: $\mathrm{S}_{f_N f_N}(f) \neq \sigma^2_{f,N}$



$\hat{\mathrm{S}}_{yy} = \mathrm{S}_{y'y'}$

y'_k

model

$\mathrm{A}_N(z)$

white

e_k

$\mathrm{S}_{ee}(f) = \sigma^2_{f,N}$

$\frac{1}{\mathrm{A}_N(z)}$

$\hat{\mathrm{S}}_{yy} = \mathrm{S}_{y'y'}$

y'_k

model

AR Modeling: Spectral Interpretation

- prediction error variance minimization in the frequency domain:

$$\sigma^2_{f,N} = E\, f^2_{N,k} = \int_{-0.5}^{0.5} S_{f_N f_N}(f) df = \min_{A_{N,i} i=1,\ldots,N} \int_{-0.5}^{0.5} |A_N(f)|^2\, S_{yy}(f) df$$

- on the other hand: $\int_{-0.5}^{0.5} |A_N(f)|^2\, df = \|A_N\|^2 = \sum_{n=0}^{N} |A_{N,n}|^2 \geq |A_{N,0}|^2 = 1$

  monicity $\Rightarrow$ $|A_N(f)|$ cannot be small at all frequencies

- The degrees of freedom $A_{N,i}$ are used to minimize the total contribution of $|A_N(f)|^2\, S_{yy}(f) \geq 0$. $\sigma^2_{f,N}$ is made small by making $|A_N(f)|$ small. However, with a limited number of $A_{N,i}$, $|A_N(f)|$ cannot be made small at all frequencies. Hence, $|A_N(f)|$ should preferably be small at those frequencies where $S_{yy}(f)$ is big in order to minimize $\sigma^2_{f,N}$ well.

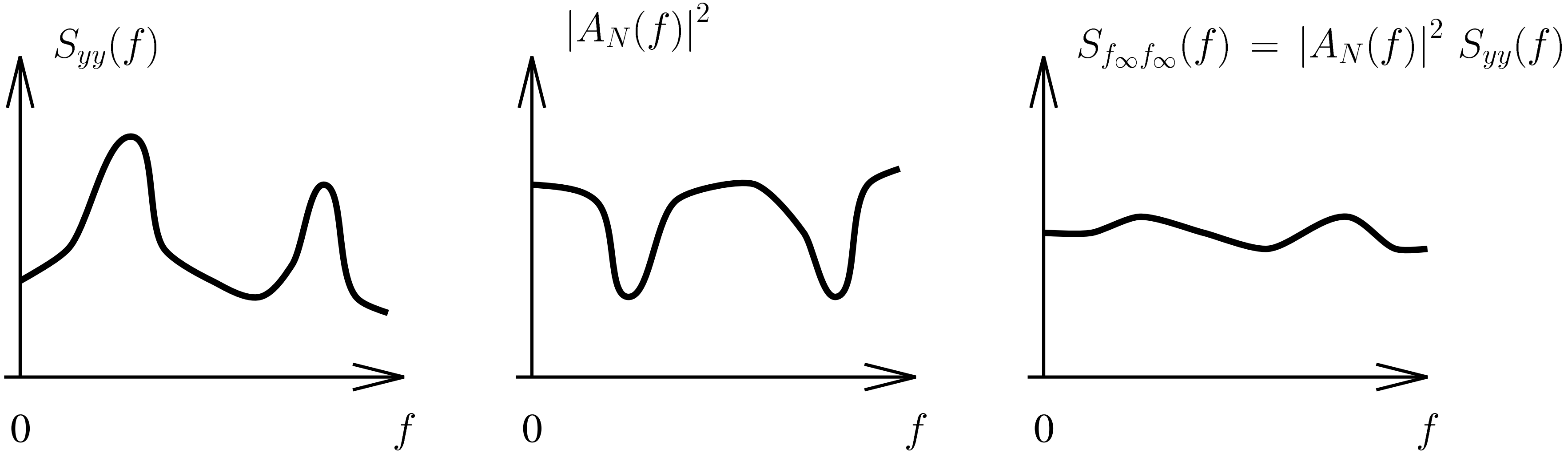
- This explains why $\mathrm{A}_N(z)$ has its zeros on the unit circle if y_k consists of at most N complex sinusoids, in which case $\sigma^2_{f,N} = 0$.

AR Modeling: Spectral Interpretation (2)

- So $|A_N(f)|$ will be such that $|A_N(f)|^2 S_{yy}(f)$ will not exceed its average value by much when $S_{yy}(f)$ is big.

- But $|A_N(f)|$ will not do much to avoid that $S_{f_\infty f_\infty}(f) = |A_N(f)|^2 S_{yy}(f)$ is small when $S_{yy}(f)$ is small.

- Hence $\widehat{S}_{yy}(f) = \widehat{\mathrm{S}}_{yy}(e^{j2\pi f})$ will match $S_{yy}(f)$ closely at the peaks of $S_{yy}(f)$. We say that the AR model follows the ($N/2$ most significant) peaks of $S_{yy}(f)$.



AR Modeling: Covariance Matching

- Suppose we want to model y_k by an AR(N) process $y_k^{'}$. This means that we want to approximate $\mathrm{S}_{yy}(z)$ by

$$\widehat{\mathrm{S}}_{yy}(z) \;=\; \mathrm{S}_{y'y'}(z) \;=\; \frac{\sigma^2_{f,N}}{\mathrm{A}_N(z)\mathrm{A}_N(1/z)}\;.$$

  $\Rightarrow$ we choose a particular parametric form for $\widehat{\mathrm{S}}_{yy}(z)$ in which the parameters $\sigma^2_{f,N}$, $A_{N,1},\ldots,A_{N,N}$ are to be determined.

- Consider now fixing these parameters by introducing $N{+}1$ covariance matching constraints:

$$\int_{-0.5}^{0.5}\widehat{S}_{yy}(f)e^{j2\pi fk}df = \hat{r}_k \;=\; r_k = \int_{-0.5}^{0.5} S_{yy}(f)e^{j2\pi fk}df\;,\quad k=0,1,\ldots,N$$

- Then the parameters $\sigma^2_{f,N}$, $A_{N,1},\ldots,A_{N,N}$ that make the first $N{+}1$ covariance lags match are the ones that are found from linear prediction! Indeed, the AR(N) process satisfies the Yule-Walker equations with covariance sequence $\hat{r}_k$. But since $\hat{r}_k = r_k,\; k=0,1,\ldots,N$, these Yule-Walker equations for lags $0,1,\ldots,N$ become the normal equations of linear prediction. Hence, the AR(N) model determined by linear prediction matches the first $N{+}1$ covariance lags.

AR Modeling: Itakura-Saito Distance Minimization

- Itakura and Saito introduced the following distance measure between two power spectral densities:

$$d(S,\widehat{S}) = \int_{-0.5}^{0.5} \left\{ \frac{S(f)}{\widehat{S}(f)} - \ln \frac{S(f)}{\widehat{S}(f)} - 1 \right\} df$$

note that the function $x - 1 - \ln x \geq 0$ for $x \geq 0$ and is only zero for $x = 1$.

- investigate the three properties of a valid distance function:

(i) $d(S,S) = 0$

(ii) $d(S,\widehat{S}) \geq 0,\ d(S,\widehat{S}) > 0$ if $S \neq \widehat{S}$

(iii) $d(S,\widehat{S}) \neq d(\widehat{S},S)$. Because the symmetry property is not satisfied, the Itakura-Saito distance is not a true distance function. Nevertheless it is a useful measure.

- Assume again an AR(N) model: $\widehat{\mathrm{S}}_{yy}(z) = \dfrac{\sigma^2_{f,N}}{\mathrm{A}_N(z)\mathrm{A}_N(1/z)}$.

This time, we shall determine the parameters $\sigma^2_{f,N}$, $A_{N,1},\ldots,A_{N,N}$ by

$$\min_{\sigma^2_{f,N},A_{N,i}} d(S_{yy},\widehat{S}_{yy}) \quad \Rightarrow \quad \sigma^2_{f,N}, A_{N,1},\ldots,A_{N,N} \text{ from linear prediction again}$$

AR Modeling: Maximum Entropy Method

- Suppose: only statistical info about y_k is $r_n\,,\ n = 0, 1, \ldots, N$ (in practice, r_n estimated from data). In classical spectral estimation theory (Blackman-Tukey spectral estimator), we take $\hat{r}_n = r_n\,,\ n = 0, 1, \ldots, N$,
$\hat{r}_n = 0,\ n > N \quad \Rightarrow$ spectral estimate with limited resolution capability.

- Suppose: want to model y_k by $y_k^{'}$ with $\hat{r}_n$ such that $\hat{r}_n = r_n,\ n = 0, 1, \ldots, N$, and $\hat{r}_n,\ n > N$ is taken such that the modeled y_k is as random as possible: we don't want to introduce any further assumptions about the process (the $\hat{r}_n$) other than $\hat{r}_n = r_n,\ n = 0, 1, \ldots, N$. Measure of randomness: *entropy*.

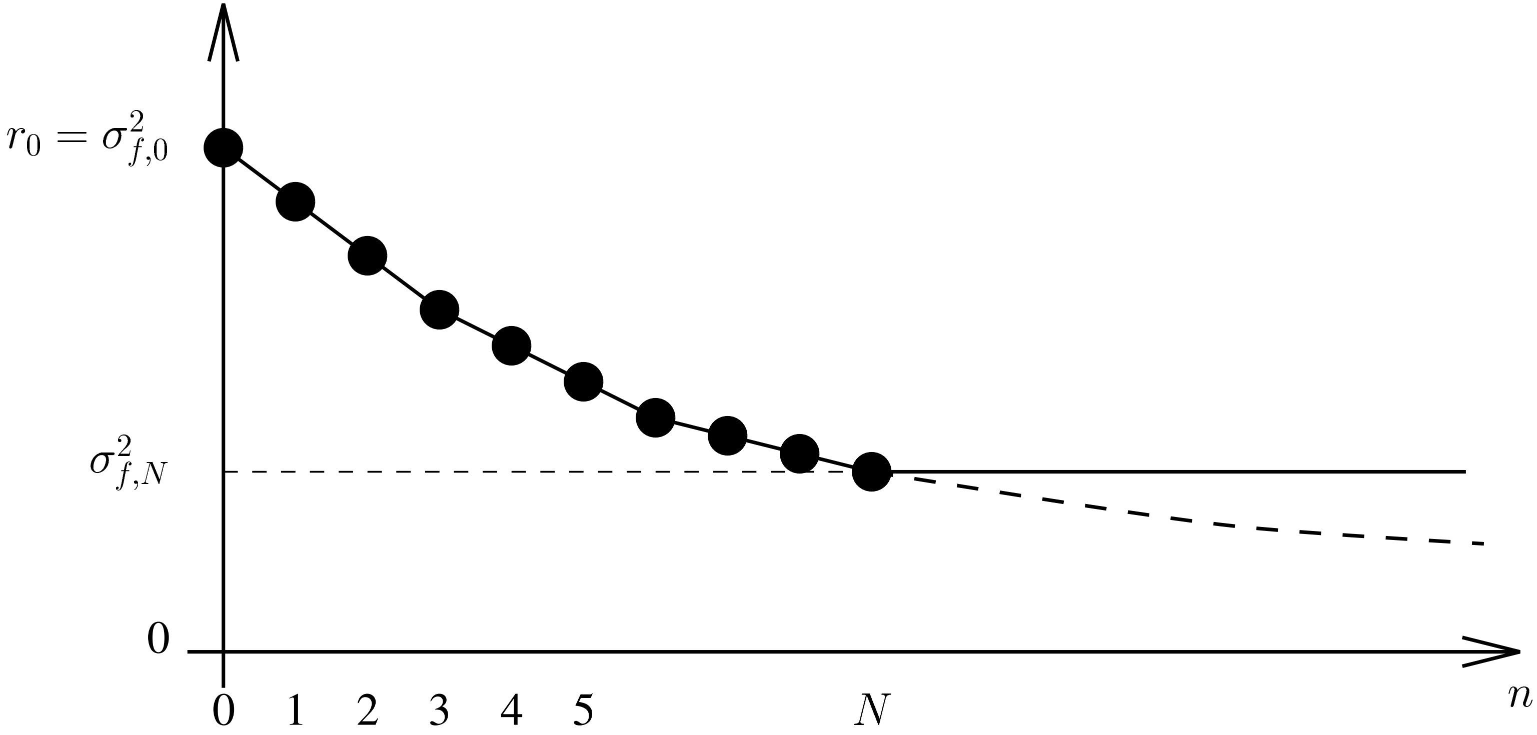
- Key results: when we maximize the entropy w.r.t. a distribution function subject to constraints on some of its second-order moments, the the resulting distribution is Gaussian. For a stationary Gaussian process, the entropy rate per sample is given by $\log \sigma^2_{f,\infty}$. Hence, to maximize the entropy, we have to maximize $\sigma^2_{f,\infty}$ for a Gaussian process.

- $\hat{r}_n = r_n,\ n = 0, 1, \ldots, N$ are given $\Rightarrow\ \sigma^2_{f,n},\ n = 0, 1, \ldots, N$ are determined. Then

$$\sigma^2_{f,\infty} \;=\; \sigma^2_{f,N} \prod_{n=N+1}^{\infty} (1 - K_n^2) \;\leq\; \sigma^2_{f,N}$$

We can choose $\{\hat{r}_n,\ n > N\} \quad \Rightarrow \quad \{K_n,\ n > N\}$.

AR Modeling: Maximum Entropy Method (2)



- $\max\limits_{K_n, n>N} \sigma^2_{f,\infty} \;=\; \sigma^2_{f,N} \max\limits_{K_n, n>N} \prod\limits_{n=N+1}^{\infty} (1-K_n^2) \;=\; \sigma^2_{f,N}$ for $K_n = 0,\; n > N$
- $\Rightarrow$ Gaussian AR(N) process that satisfies LP normal equations
- Note that we obtain the maximum entropy covariance extension $\hat{r}_n,\; n > N$:

$$\hat{r}_n \;=\; r_n \;,\;\; n = 0, 1, \ldots, N \;, \qquad \hat{r}_n \;=\; -\sum_{i=1}^{N} A_{N,i} \hat{r}_{n-i} \;,\;\; n > N$$

This contrasts with MA(N) model matching for which $\hat{r}_n = 0,\; n > N$.

AR Modeling: Spectral Flatness Measure

- first: spectral flatness measure for a covariance matrix R_N with positive real eigenvalues $\lambda_1 \ldots \lambda_N$.
- For white noise, we have $R_N = \sigma_y^2 I_N$ and hence $\lambda_1 = \cdots = \lambda_N = \sigma_y^2$.
- For a general covariance matrix, how close is the process to white noise? Answer: flatness measure FM of the distribution of the λ_i:

$$FM = \frac{\text{geometric avg.}}{\text{arithmetic avg.}} = \frac{(\prod_{i=1}^{N} \lambda_i)^{1/N}}{\frac{1}{N}\sum_{i=1}^{N} \lambda_i} = \frac{e^{\ln(\prod_{i=1}^{N} \lambda_i)^{1/N}}}{\frac{1}{N}\sum_{i=1}^{N} \lambda_i} = \frac{e^{\frac{1}{N}\sum_{i=1}^{N} \ln \lambda_i}}{\frac{1}{N}\sum_{i=1}^{N} \lambda_i} \leq 1$$

$FM = 1$ iff $\lambda_i \equiv \sigma_y^2$

AR Modeling: Spectral Flatness Measure (2)

- to show $FM \leq 1$: *Jensen's Inequality*

  Let $f(.)$ be a convex function and X a random variable. Then

  $$f(E\,X) \leq E\,f(X)\;.$$

  If $f(.)$ is strictly convex, then strict inequality holds unless the distribution of X is concentrated in one point. If X has a discrete distribution, taking on the M values x_i with probabilities $\alpha_i > 0$, $\Sigma_{i=1}^{M}\,\alpha_i = 1$ then this can be written as

  $$f(\sum_{i=1}^{M} \alpha_i\, x_i) \leq \sum_{i=1}^{M} \alpha_i\, f(x_i)\;.$$

  Proof: recursively: for $M = 2$: definition of a convex function.Then

  $$f(\sum_{i=1}^{M} \alpha_i\, x_i) = f(\alpha_1 x_1 + (1-\alpha_1) \sum_{i=2}^{M} \frac{\alpha_i}{\Sigma_{k=2}^{M}\,\alpha_k}\, x_i) \leq \alpha_1 f(x_1) + (1-\alpha_1)\, f(\sum_{i=2}^{M} \frac{\alpha_i}{\Sigma_{k=2}^{M}\,\alpha_k}\, x_i)$$

  The property for a continuous distribution can be shown by letting $M \to \infty$.

- Application: $f(x) = -\ln x$ (strictly convex), $x_i = \lambda_i$, $\alpha_i = 1/N$, $M = N \Rightarrow$

  $$-\ln(\frac{1}{N}\sum_{i=1}^{N} \lambda_i) \leq -\frac{1}{N}\sum_{i=1}^{N} \ln \lambda_i \;\Rightarrow\; \frac{1}{N}\sum_{i=1}^{N} \lambda_i \geq e^{\frac{1}{N}\sum\limits_{i=1}^{N} \ln \lambda_i} \;\Rightarrow\; FM \leq 1$$

AR Modeling: Spectral Flatness Measure (3)

- As $N \to \infty$, the distribution of the λ_i behaves similarly as $S_{yy}(f)$, e.g.

$$\lim_{N\to\infty} \frac{\max\limits_{i=1,\ldots,N} \lambda_i}{\min\limits_{i=1,\ldots,N} \lambda_i} = \frac{\max\limits_{f\in[0,0.5]} S_{yy}(f)}{\min\limits_{f\in[0,0.5]} S_{yy}(f)} .$$

- Similarly, the flatness measure becomes in the limit as $N \to \infty$ the spectral flatness measure SFM of y_k

$$FM = \frac{e^{\frac{1}{N}\sum\limits_{i=1}^{N} \ln \lambda_i}}{\frac{1}{N}\sum\limits_{i=1}^{N} \lambda_i} \xrightarrow{N\to\infty} \frac{e^{\int_{-0.5}^{0.5} \ln S_{yy}(f)df}}{\int_{-0.5}^{0.5} S_{yy}(f)df} = \frac{\sigma^2_{f,\infty}}{\sigma^2_y} = SFM = \xi_y \in [0,1]$$

  Note that if e_k is white noise, then $\xi_e = 1$.

- Apply SFM to $f_{N,k}$: $\xi_{f_N} = \frac{\sigma^2_{f,\infty}}{\sigma^2_{f,N}} \Rightarrow \max\limits_{A_{N,i},\, i=1,\ldots,N} \xi_{f_N} \leftrightarrow \min\limits_{A_{N,i},\, i=1,\ldots,N} \sigma^2_{f,N}$

  Hence, LP = choose $A_{N,i}$ to minimize $\sigma^2_{f,N}$
  = make $S_{f_N f_N}(f) = S_{yy}(f)\,|A_N(f)|^2$ as flat (white) as possible.

AR Modeling: Spectral Estimation Qualities

So far: AR(N) modeled from $r_{yy}(k),\ k = 0, 1, \ldots, N$.
In practice: we estimate $\sigma^2_{f,N}, A_{N,1}, \ldots, A_{N,N}$ from M samples $y_0, y_1, \ldots, y_{M-1}$.

Given all the previous observations, we can conclude that the AR(N) model obtained with linear prediction gives a good spectral estimate for N sufficiently high. More precisely we can state:

- *bias*: the AR(N) model is only unbiased for AR(n) processes with $n \leq N$. Bias smaller for spectral peaks than for spectral valleys. In general, the bias disappears as $N \rightarrow \infty$.

- *variance*: so far we have assumed $r_0, \ldots, r_N$ known. In practice, they will have to be estimated from data. However, since they represent $N{+}1$ parameters to be estimated, we can state that the variance will be roughly proportional to N/M. So the variance will be low if $N \ll M$.

- *resolution*: due to the all-pole filter, we can in principle model arbitrarily closely spaced spectral peaks (sinusoids). There is no spectral smearing. For this reason, the AR modeling is also called a *high resolution* technique.

AR Modeling: Techniques: Least-Squares

- least-squares: replace statistical averages by temporal averages

$$\begin{bmatrix} f_{N,0} \\ \vdots \\ \\ f_{N,N} \\ \vdots \\ f_{N,M-1} \\ \vdots \\ \\ f_{N,M+N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} y_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ & & \ddots & 0 \\ y_N & & \cdots & y_0 \\ \vdots & & & \vdots \\ y_{M-1} & \cdots & & y_{M-N+1} \\ 0 & \ddots & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & y_{M-1} \end{bmatrix}}_{=\ \mathcal{Y}\ \text{(Toeplitz data matrix)}} \begin{bmatrix} 1 \\ A_{N,1} \\ \vdots \\ A_{N,N} \end{bmatrix} \quad \begin{matrix} \}\,\mathcal{Y}^{pre} & \\ \}\,\mathcal{Y}^{cov} & \}\,\mathcal{Y}^{corr} \\ \}\,\mathcal{Y}^{po} & \end{matrix}$$

1. *correlation method* (pre- and postwindowed)

$$\min_{A_{N,i}} \sum_{k=0}^{M+N-1} f_{N,k}^2 \;\Rightarrow\; \mathcal{Y}^T\mathcal{Y}\,A_N = [(M+N)\sigma^2_{f,N}\;0\cdots0]^T\,,\quad \mathcal{Y}=\mathcal{Y}^{corr}$$

$\mathcal{Y}^T\mathcal{Y}$ = Toeplitz $\Rightarrow$ Levinson, A$_N(z)$ minimum-phase

AR Modeling: Techniques (2)

2. *covariance method* (unwindowed)

only take prediction errors calculated with actual data

$$\min_{A_{N,i}} \sum_{k=N}^{M-1} f_{N,k}^2 \;\Rightarrow\; \mathcal{Y}^T\mathcal{Y}\,A_N = [(M-N)\sigma_{f,N}^2\; 0\cdots 0]^T\;,\;\; \mathcal{Y} = \mathcal{Y}^{cov}$$

$\mathcal{Y}^T\mathcal{Y} \;\neq$ Toeplitz $\Rightarrow$ not Levinson, $\mathrm{A}_N(z)$ not guaranteed minimum-phase
nevertheless, better estimation quality due to absence of windowing, especially for short data lengths M

3. *modified covariance method* (unwindowed)

backward prediction errors: $\begin{bmatrix} \vdots \\ b_{N,k} \\ \vdots \end{bmatrix} = \mathcal{Y} \begin{bmatrix} A_{N,N} \\ \vdots \\ A_{N,1} \\ 1 \end{bmatrix}$

$$\min_{A_{N,i}} \sum_{k=N}^{M-1} (f_{N,k}^2 + b_{N,k}^2) \;\Rightarrow\; \left(\mathcal{Y}^T\mathcal{Y} + J\mathcal{Y}^T\mathcal{Y}J\right)\; A_N = [2(M-N)\sigma_N^2\; 0\cdots 0]^T\,,\; \mathcal{Y} = \mathcal{Y}^{cov}$$

$\mathcal{Y}^T\mathcal{Y} + J\mathcal{Y}^T\mathcal{Y}J$ centro-symmetric $\Rightarrow$ further improved estimate

AR Modeling: Techniques (3)

4. *Itakura-Saito method* keep Levinson recursions, but replace statistical average by temporal average in the PARCOR calculations:

$$K_{n+1} = -\frac{\sum_k f_{n,k} b_{n,k-1}}{\sqrt{\sum_k f_{n,k}^2}\sqrt{\sum_k b_{n,k-1}^2}}$$

5. *Burg method* take the Levinson recursions for the prediction errors:

$$\begin{cases} f_{n+1,k} = f_{n,k} + K_{n+1} b_{n,k-1} \\ b_{n+1,k} = b_{n,k-1} + K_{n+1} f_{n,k} \end{cases}$$

and take the modified covariance criterion

$$\min_{K_{n+1}} \sum_k \left(f_{n+1,k}^2 + b_{n+1,k}^2\right) \Rightarrow K_{n+1} = -\frac{\sum_k f_{n,k} b_{n,k-1}}{\frac{1}{2}\left(\sum_k f_{n,k}^2 + \sum_k b_{n,k-1}^2\right)}$$

One can show that

$$\left|K_n^{Burg}\right| \leq \left|K_n^{Ita-S}\right| \leq 1 \quad \Rightarrow \mathrm{A}_N(z) \text{ minimum-phase}$$

first inequality: arithmetic average $\geq$ geometric average
second inequality: Cauchy-Schwarz

AR Modeling: Techniques (4)

6. *Maximum-Likelihood*

   assume the y_k given $\theta = [\sigma^2_{f,N}\ A_{N,1} \cdots A_{N,N}]^T$ Gaussian and AR(N), and estimate θ via the ML method

7. *Method of Moments*

   take the normal equations of linear prediction

$$R_{N+1}\, A_N \;=\; [\sigma^2_{f,N}\, 0 \cdots 0]^T$$

   and replace $r_n,\ n = 0, 1, \ldots, N$ by sample estimates (such that $\widehat{R}_{N+1} > 0$).

   example: correlation method (biased sample moments $\hat{r}_n = \frac{1}{M}\sum\limits_{k=0}^{M-1-n} y_{k+n} y_k$)

   another example: unbiased sample moments $\hat{r}_n = \frac{1}{M-n}\sum\limits_{k=0}^{M-1-n} y_{k+n} y_k$ do not guarantee $\widehat{R}_{N+1} > 0$ but usually $\widehat{R}_{N+1} > 0$ since $R_{N+1} > 0$ and the $\hat{r}_n$ are consistent estimates of the r_n

AR Modeling: Order Selection

- given $r_0, r_1, \ldots \;\Rightarrow\; \sigma^2_{f,N} \searrow$ as $N \to \infty$: the higher N the better
- given data $y_0, \ldots, y_{M-1}$, so far: assumed N given
- Due to least-squares fit: estimated $\widehat{\sigma}^2_{f,N}$ decreases with N and $\widehat{\sigma}^2_{f,N} < \sigma^2_{f,N}$. example: covariance method with $M = 2N \;\Rightarrow\; \widehat{\sigma}^2_{f,N} = 0\,!$ Exact fit possible (exactly determined equations).
  On the basis of $\widehat{\sigma}^2_{f,N}$: the higher N the better.
- When try $\widehat{A}_N$, estimated with certain data, on other data, $E\widehat{f}^2_{N,k} > \sigma^2_{f,N}$ due to estimation errors in $\widehat{A}_N$.
- order selection criteria: $\min\limits_N c(N)\;,\;\; c(N)\; = g(\widehat{\sigma}^2_{f,N}) + d(N)$
  where $g(.)$, $d(.)$ are monotonuously increasing functions $\Rightarrow\; g(\widehat{\sigma}^2_{f,N})$ decreases with N whereas $d(N)$ increases with $N \Rightarrow$ a compromise has to be made, leading to a finite optimal N

1. Akaike ['70]: *Final Prediction Error (FPE)*

$$FPE(N) \;=\; \frac{M+N}{M-N}\widehat{\sigma}^2_{f,N} = \widehat{\sigma}^2_{f,N} + \frac{2N}{M-N}\widehat{\sigma}^2_{f,N}$$

AR Modeling: Order Selection (2)

2. Akaike ['74]: *Akaike Information Criterion (AIC)*

$$AIC(N) \;=\; M\;\ln\widehat{\sigma}^2_{f,N} + 2N$$

for $\dfrac{N}{M} \ll 1,\;\; AIC(N) \;\approx\; M\;\ln FPE(N)$.

3. Rissanen ['78]: *Minimum Description Length (MDL)*

$$MDL(N) \;=\; \ln\widehat{\sigma}^2_{f,N} + (N{+}1)\,\frac{\ln M}{M}$$

MDL gives consistent estimates: $\widehat{N}_{MDL} \rightarrow N$ as $M \rightarrow \infty$ for an AR(N) process

- remark that the Levinson-style order-recursive solutions are helpful: find AR estimates for all orders and then choose the best order according to $\min\limits_{N} c(N)$ where $c = FPE, AIC, MDL$

Linear Time-Frequency Representations

- non-stationary processes ⇒ no ergodicity ⇒ cannot obtain statistical averages as limits of time averages ⇒ no time-averaging

- spectral estimation → spectral representation

- fundamental spectral representation: Fourier transform

$$Y(f) \;=\; \int_{-\infty}^{\infty} y(t)\, e^{-j2\pi f t}\, dt$$

  $y(t)$: we know precisely at what time something happens but we don't know at which frequencies
  $Y(f)$: we know precisely the different spectral components of the signal, but we don't know when they occur

- non-stationary signals: would like a joint time-frequency representation
  e.g.: piano piece: pitch (fundamental frequency) is a piecewise constant function of time (notes being played), $y(t)$ does not tell us which notes are being played, $Y(f)$ shows all the notes but does not tell us when they occur.