The Levinson Algorithm

The Levinson algorithm is a fast recursion to solve the Yule-Walker prediction-error equations that model a random process, or more in general to implicitly invert a Toeplitz matrix. It gives rise to a computational structure (lattice filter) with useful properties. It can be used to *analyze* an autocorrelation sequence, and to *generate* a process with this autocorrelation.

It is used for parametric spectrum estimation (based on all-pole or AR modeling), and e.g. speech coding in GSM: instead of transmitting speech samples, the filter coefficients are transmitted so that the receiver can reconstruct the speech.

(These slides: real-valued signals. See book for the complex case.)

Recap: AR modeling of a random process

Suppose we want to model a random signal $x[n]$ using an AR filter applied to a white noise, $v[n]$,

$$x[n] = -a_1 x[n-1] - \cdots - a_p x[n-p] + v[n]$$

The input signal $v[n]$ is also known as the "innovation".

- White implies that $v[n]$ is not correlated to past samples ($v[n-1]$, etc), and hence $v[n]$ is uncorrelated to $x[n-1], x[n-2]$, etc:

$$E(v[n]x[n-m]) = 0, \quad m = 1, 2, \cdots$$

- In the z-domain, the model is written as

$$X(z) = \frac{B(z)}{A(z)} V(z), \qquad A(z) = 1 + a(1)z^{-1} + \cdots + a(p)z^{-p}, \quad B(z) = 1\,.$$

Application: EEG signal modeling

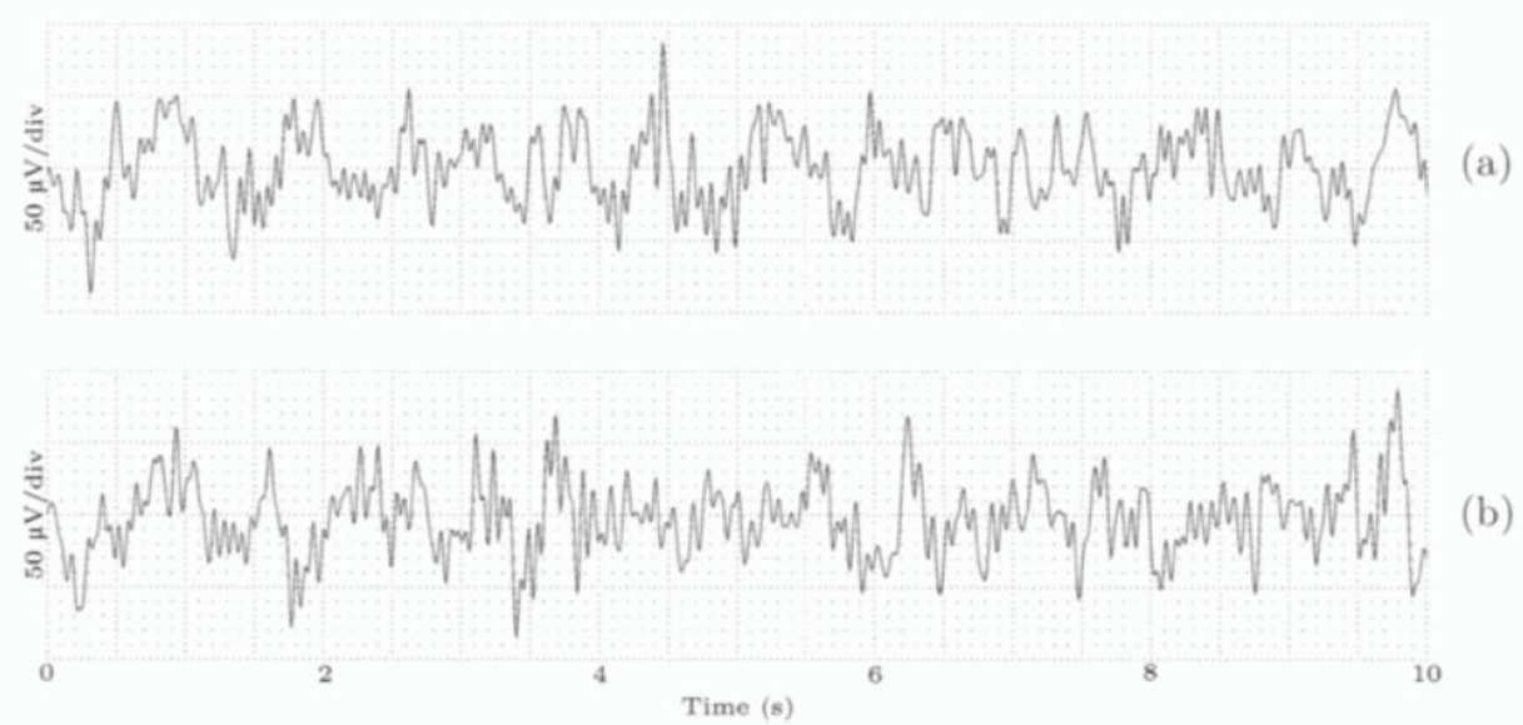


Figure 3.2: (a) An EEG signal with a prominent alpha rhythm, and (b) a simulated signal produced by an AR model whose parameters were estimated from the signal displayed in (a).

(*Bioelectrical signal processing in cardiac and neurological applications*, Leif Sornmo, Pablo Laguna)

Application: parametric spectrum estimation

- The power spectral density of $x[n]$ is given by

$$P_x(e^{j\omega}) = \sigma_v^2 \left| \frac{B(e^{j\omega})}{A(e^{j\omega})} \right|^2$$

- Parametric spectrum estimation is: given the correlation sequence $r_x(k)$ of $x[n]$, find $P_x(e^{j\omega})$. For this we need to find $A(z)$ and $B(z)$, i.e. model identification.

Application: prediction filter modeling

We are given a signal x and wish to *predict* the current sample $x[n]$ as a linear combination of p past samples $x[n-1], \cdots, x[n-p]$:

$$\hat{x}[n] = -a_1 x[n-1] - \cdots - a_p x[n-p]$$

To estimate the optimal coefficients, we can minimize the *prediction error*:

$$e[n] \;=\; x[n] - \hat{x}[n] \;=\; x[n] + a_1 x[n-1] - \cdots + a_p x[n-p]$$

- In a deterministic setting, we can minimize $\mathcal{E}_p = \sum |e[n]|^2$.
- In a stochastic setting, we minimize $\epsilon_p = E(|e[n]|^2)$.

This leads to the same equations as before (with $e[n] = v[n]$ and $\epsilon_p = \sigma_v^2$).

From prediction filter to Yule-Walker equations

Model for $x[n]$:

$$x[n] = -a(1)x[n-1] - \cdots - a(p)x[n-p] + v[n]$$

- In general, a random process x will not precisely satisfy such a model. For any model order p, we try to find the best fitting parameters $a(1), \cdots, a(p)$ that will minimize the residual noise variance ϵ_p. That is, we consider

$$x[n] + a(1)x[n-1] + \cdots + a(p)x[n-p] = v[n]$$

  and minimize the power of the *prediction error* $v[n]$.

- If the model holds, we can easily derive the *Yule-Walker equations*, as follows. Multiply by $x[n-m]$ and take expectation:

$$E(x[n]x[n-m]) = -a(1)E(x[n-1]x[n-m]) - \cdots - a(p)E(x[n-p]x[n-m]) + E(v[n]x[n-m])$$

For $m > 0$ this gives, with $r_x(m) = E(x[n]x[n-m]) = E(x[n]x[n+m])$,

$$r_x(m) \;=\; -a(1)r_x(m-1) \;-\; \cdots \;-\; a(p)r_x(m-p)\,, \qquad m > 0$$

Yule-Walker equations

- These equations can be written in matrix form (with rows for $m = 1, 2, \cdots, p$)

$$\begin{bmatrix} r_x(0) & r_x(1) & \ddots & r_x(p-1) \\ r_x(1) & r_x(0) & r_x(1) & \ddots \\ \ddots & r_x(1) & \ddots & r_x(1) \\ r_x(p-1) & \ddots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(p) \end{bmatrix} \quad \Leftrightarrow \quad \mathbf{R}_x\mathbf{a} = \mathbf{r}$$

  The matrix is constant along diagonals: a *Toeplitz* matrix. It is also symmetric.

- For $m = 0$, we need to compute $E(v[n]x[n]) = E(v[n](-a(1)x[n-1] - \cdots - a(p)x[n-p] + v[n]))$. Because $E(v[n]x[n-i]) = 0$ for $i > 0$ (see before), we have $E(v[n]x[n]) = E(v[n]^2) = \sigma_v^2 =: \epsilon_p$. We obtain an additional equation:

$$r_x(0) = -a(1)r_x(1) - \cdots - a(p)r_x(p) + \epsilon_p$$

Yule-Walker equations

- To solve the Yule-Walker equations is straightforward: the filter coefficients are $\mathbf{a} = \mathbf{R}_x^{-1}\mathbf{r}$. The noise power (prediction error or innovation power) ϵ_p follows from the above equation for $m = 0$.

  This has a complexity of order p^3 multiplications. The Levinson algorithm can reduce this to order p^2, by exploiting the Toeplitz structure.

Levinson algorithm

We will derive a recursion, for $p = 1, 2, \cdots$. For each p, we have different filter coefficients $a_p(i)$ ($i = 1, \cdots, p$) and modeling/prediction error power ϵ_p.

- For order $p = 0$, the YW equations give $\epsilon_0 = r_x(0)$ (there is no filter)
- For order $p = 1$, we have $r_x(0)a_1(1) = -r_x(1)$, or $a_1(1) = -\frac{r_x(1)}{r_x(0)}$, and $\epsilon_1 = r_x(0) + a_1(1)r_x(1) = \frac{r_x(0)^2 - r_x(1)^2}{r_x(0)}$.
- For order p, combine the YW equations for $m = 0$ and $m > 0$ into a single matrix of size $p + 1$:

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

Suppose we know the solution $(a_p(1), \cdots, a_p(p))$ for order p, how can we find the solution for order $p + 1$ efficiently?

Levinson algorithm

Take as trial solution for order $p+1$ the old solution for order p, extended by 0:

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(p+1) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(p) \\ r_x(2) & r_x(1) & r_x(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & r_x(1) \\ r_x(p+1) & r_x(p) & \cdots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ \vdots \\ a_p(p) \\ 0 \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \gamma_p \end{bmatrix}$$

This is not the correct solution of the YW equations, because in general $\gamma_p \neq 0$. We need to modify the trial solution such that $\gamma_p = 0$; in that case the RHS has the required form and we have a solution to the YW equations of order $p+1$.

To modify the trial solution, we apply the following two tricks.

The Levinson algorithm

Trick 1: the Toeplitz structure gives rise to the following equation (for order p):

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

Trick 2: The two equations can be combined:

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

Levinson algorithm

For $p+1$, we take as trial solutions:

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(p+1) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(p) \\ r_x(2) & r_x(1) & r_x(0) & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & r_x(1) \\ r_x(p+1) & r_x(p) & \ddots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a_p(1) & a_p(p) \\ \vdots & \vdots \\ a_p(p) & a_p(1) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \epsilon_p & \gamma_p \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \gamma_p & \epsilon_p \end{bmatrix}$$

Both trial solutions are not correct solutions of the YW equations of order $p+1$, because $\gamma_p \neq 0$. But we can take linear combinations of the two trial solutions in such a way that γ_p becomes 0.

The Levinson algorithm

Indeed, for any 2×2 matrix $\mathbf{M}$:

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(p+1) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(p) \\ r_x(2) & r_x(1) & r_x(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & r_x(1) \\ r_x(p+1) & r_x(p) & \cdots & r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a_p(1) & a_p(p) \\ \vdots & \vdots \\ a_p(p) & a_p(1) \\ 0 & 1 \end{bmatrix} \mathbf{M} = \begin{bmatrix} \epsilon_p & \gamma_p \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \gamma_p & \epsilon_p \end{bmatrix} \mathbf{M}$$

We will take linear combinations $\mathbf{M}$ of the following form: (book uses $-\rho_{p+1} = \Gamma_{p+1}$)

$$\begin{bmatrix} \epsilon_p & \gamma_p \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \gamma_p & \epsilon_p \end{bmatrix} \underbrace{\begin{bmatrix} 1 & -\rho_{p+1} \\ -\rho_{p+1} & 1 \end{bmatrix}}_{\mathrm{M}} = \begin{bmatrix} \epsilon_{p+1} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & \epsilon_{p+1} \end{bmatrix}$$

For $\rho_{p+1} = \frac{\gamma_p}{\epsilon_p}$, this will bring the RHS into the required form. The LHS is then the solution of the YW equations of order $p+1$. (ρ_{p+1} is called a *reflection coefficient*.)

The Levinson algorithm

Thus, we need to set $\rho_{p+1} = \frac{\gamma_p}{\epsilon_p}$. We obtain as solution

$$\begin{bmatrix} 1 & 0 \\ a_p(1) & a_p(p) \\ \vdots & \vdots \\ a_p(p) & a_p(1) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\rho_{p+1} \\ -\rho_{p+1} & 1 \end{bmatrix} =: \begin{bmatrix} 1 & a_{p+1}(p+1) \\ a_{p+1}(1) & a_{p+1}(p) \\ \vdots & \vdots \\ a_{p+1}(p) & a_{p+1}(1) \\ a_{p+1}(p+1) & 1 \end{bmatrix}$$

- This is the update step in the Levinson recursion. The recursion is initiated by

$$(p = 0 :) \qquad r_x(0)[1, \;\; 1] = [\epsilon_0, \;\; \epsilon_0] \qquad \Rightarrow \qquad \epsilon_0 = r_x(0)$$

  With this, we enter the recursion. In the next step, we obtain the equations

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{bmatrix} = \begin{bmatrix} \epsilon_0 & \gamma_0 \\ \gamma_0 & \epsilon_0 \end{bmatrix} \begin{bmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{bmatrix}$$

  where $\gamma_0 = r_x(1)$. We set $\rho_1 = \gamma_0/\epsilon_0 = r_x(1)/r_x(0)$, and follow the recursion.

- The pth step in the recursion has complexity p. Overall, the complexity of solving the pth order YW equations is of order p^2: more efficient than a direct inversion.

The Levinson algorithm

To show that this recursion works and does not break down, we need to prove that we always can compute a suitable ρ. This will follow from the main assumption: $\mathbf{R}_x > 0$ (i.e., the covariance matrix strictly positive definite–all its eigenvalues are positive) for all p.

- $\epsilon_p > 0$. This follows from the YW equations:

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

  In particular, $1 = [\mathbf{R}_x^{-1}]_{0,0}\,\epsilon_p$. Since $\mathbf{R}_x$ is strictly positive definite (hence also invertible), the inverse $\mathbf{R}_x^{-1}$ exists and is also strictly positive definite. This implies that $[\mathbf{R}_x^{-1}]_{0,0} > 0$. Hence $\epsilon_p > 0$ for any p.

  In particular, $\epsilon_p \neq 0$, so that we can compute ρ_{p+1}.

The Levinson algorithm

We can show a bit more:

- $|\rho_{p+1}| < 1$

  The update equation is

  $$\epsilon_{p+1} = \epsilon_p - \rho_{p+1}\gamma_p = \epsilon_p - \frac{\gamma_p^2}{\epsilon_p} = \frac{\epsilon_p^2 - \gamma_p^2}{\epsilon_p} > 0 \quad \Rightarrow \quad |\epsilon_p| > |\gamma_p|$$

  so that $|\rho_{p+1}| < 1$.

- From this we also see that $\epsilon_{p+1} \leq \epsilon_p$: the modeling/prediction error always decreases.

- Special case: if we have a true AR process of order p, then we will find $\epsilon_{p+1} = \epsilon_p$ and $\gamma_p = 0$. At this point the recursion stops (we can take $\rho_{p+1} = 0$, so that the AR coefficients do not change anymore).

- Special case: if $|\rho_{p+1}| = 1$, then $\epsilon_{p+1} = 0$. The prediction error is zero, $x[n]$ can be exactly predicted from its past, the process is called *deterministic*.

  This can occur only if $\mathbf{R}_x$ is singular (thus the matrix is not strictly positive definite).

The Levinson algorithm

Szegö polynomials

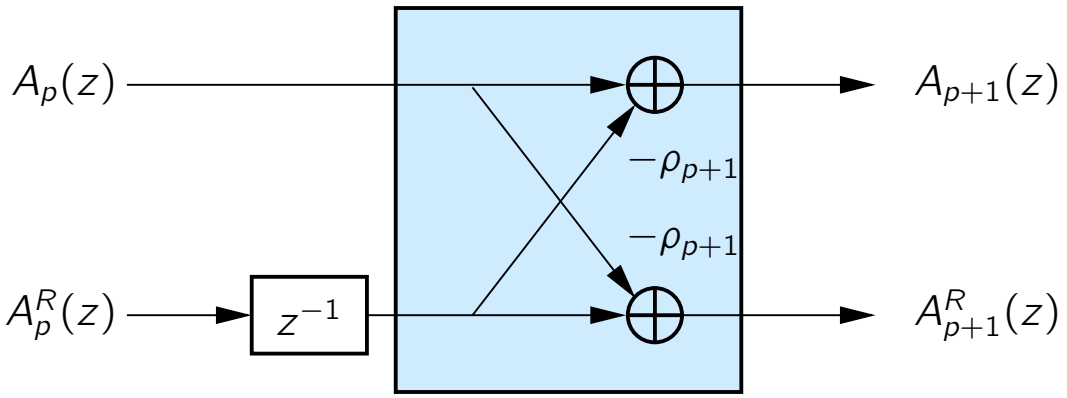
We can define filter functions corresponding to the filter coefficients:

$$\begin{cases} A_p(z) &= 1 + a_p(1)z^{-1} + \cdots + a_p(p)z^{-p} \\ A_p^R(z) &= a_p(p) + a_p(p-1)z^{-1} + \cdots + z^{-p} \end{cases}$$

This allows to write the Levinson recursion compactly as

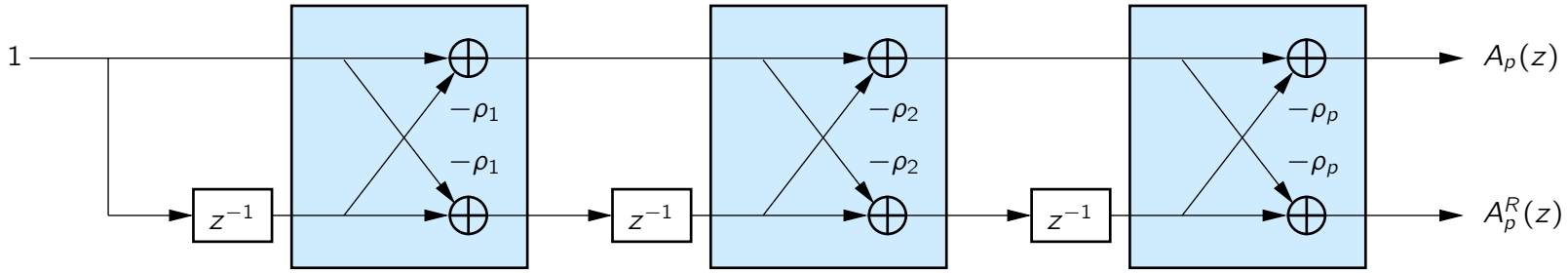
$$[A_{p+1}(z), \quad A_{p+1}^R(z)] \quad = \quad [A_p(z), \quad A_p^R(z)] \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\rho_{p+1} \\ -\rho_{p+1} & 1 \end{bmatrix}$$

with initialization ($p = 0$) as $[A_0(z), \quad A_0^R(z)] = [1, \quad 1]$.



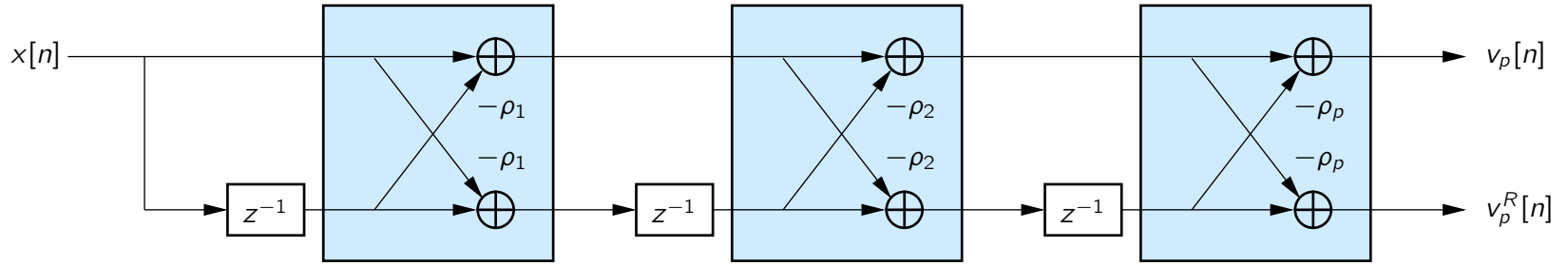
The Levinson algorithm

FIR lattice filter



Thus, the resulting system has impulse response $A_p(z)$ (and also $A_p^R(z)$, not used).

If we use this system with input $x[n]$, we obtain the prediction error sequence $v_p[n]$, see Slide 4.



The filter structure is known as a lattice filter; note it is an FIR filter.

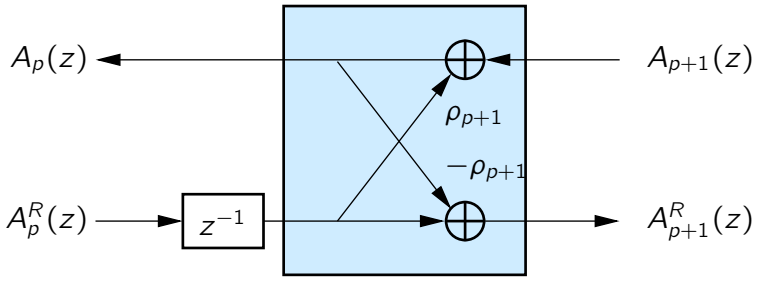
The Levinson algorithm

The recursion can be reversed as follows:

$$[A_{p+1}(z), \quad A^R_{p+1}(z)] \quad = \quad [A_p(z), \quad A^R_p(z)] \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\rho_{p+1} \\ -\rho_{p+1} & 1 \end{bmatrix}$$

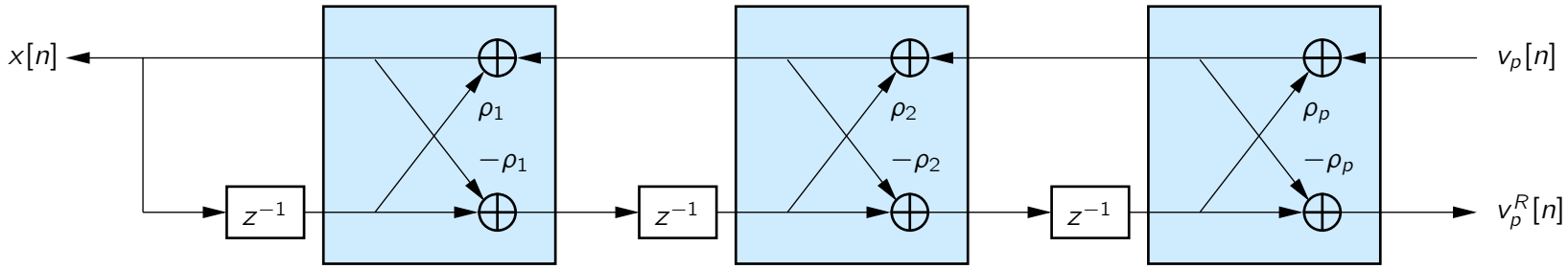
$$\Leftrightarrow \quad \begin{cases} A_{p+1}(z) &= A_p(z) - z^{-1}\rho_{p+1}A^R_p(z) \\ A^R_{p+1}(z) &= z^{-1}A^R_p(z) - \rho_{p+1}A_p(z) \end{cases} \quad \Leftrightarrow \quad \begin{cases} A_p(z) &= A_{p+1}(z) + z^{-1}\rho_{p+1}A^R_p(z) \\ A^R_{p+1}(z) &= z^{-1}A^R_p(z) - \rho_{p+1}A_p(z) \end{cases}$$

$$\Leftrightarrow \quad [A_p(z), \quad A^R_{p+1}(z)] \quad = \quad [A_{p+1}(z), \quad A^R_p(z)] \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\rho_{p+1} \\ \rho_{p+1} & 1-\rho^2_{p+1} \end{bmatrix}$$



The Levinson algorithm

IIR lattice filter



This allows to compute $x[n]$ from $v_p[n]$: the filter impulse response is $\frac{1}{A_p(z)}$. This is an IIR filter, the filter structure is recursive, i.e., the filter coefficients of $\frac{1}{A_p(z)}$ are not explicitly computed.

It can be shown that the filter is stable as long as all reflection coefficients are strictly smaller than 1. This is the case if the original $\mathbf{R}_x$ is strictly positive definite.

If we replace $v_p[n]$ by any white noise sequence (with variance ϵ_p), then the resulting output signal is a random process with the same correlation sequence $\{r_x(0), \cdots, r_x(p)\}$ as $x[n]$.

The Levinson algorithm

This is exploited in the GSM system, where $x[n]$ is the speech signal.

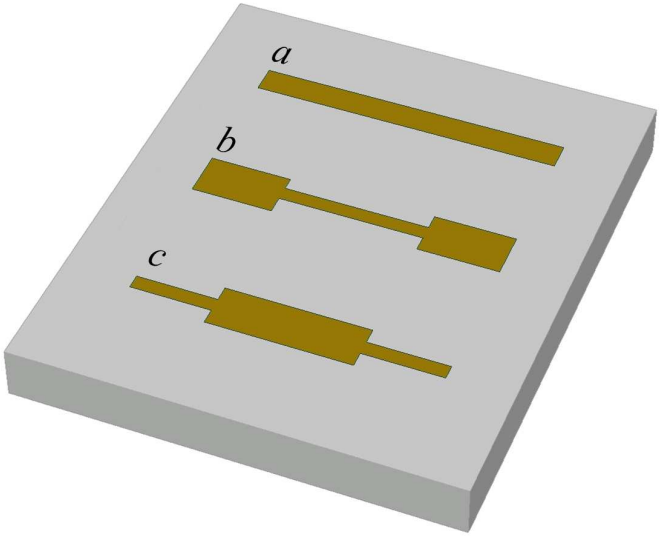
- At the transmitter, the speech signal is split in short frames, each of 20 ms. For each frame, the correlation sequence $\{r_x(0), \cdots, r_x(p)\}$ is estimated. Using the Levinson algorithm, the reflection coefficients and residual noise power ϵ_p are estimated, coded and transmitted to the receiver.

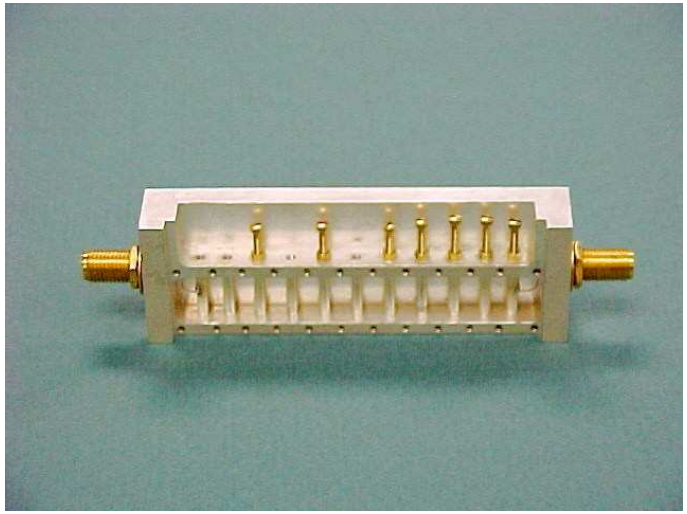
- The receiver uses $\{\rho_1, \cdots, \rho_p\}$ as coefficients in the reverse filter, and generates a white noise sequence in place of $v_p[n]$. It computes the corresponding $x[n]$. It is not the same as the original speech signal, but has the same correlation sequence, which is good enough for the ear (at least for unvoiced speech).

Microstrip filter

RF filters: microstrip and cavity filters

The resulting filter structure is also employed in RF microstrip filters. At the interface between wire segments of different width, reflections and transmissions occur (the width ratios determine the reflection coefficients). The overall structure is passive (ideally lossless).





The structure is also used in models of transmission lines, and earth layers (acoustic transmission/reflection in seismic/geophysic studies of the earth)