Optimal Signal Processing

Lesson 3

Chapter 5.
Levinson-Durbin Recursion

LTH

September 2013

Nedelko Grbic
(Mtrl from Bengt Mandersson)

Electrical and Information Technology, Lund University
Lund University

Chapter 5 Levinson-Durbin recursion

In chapter 4, we derive the normal equations or Yule-walker equations for an all-pole model (chap 4.4.3, page 162 – 165).

The difference equation for the input $\delta(n)$ **is**

$$x(n) + \sum_{k=1}^{p} a_p(k)\, x(n-k) = b(0)\delta(n)$$

and the system function

$$H(z) = \frac{b(0)}{1 + a_p(1)\, z^{-1} + a_p(2)\, z^{-2} + \cdots + a_p(p)\, z^{-p}} = \frac{b(0)}{1 + \sum_{k=1}^{p} a_p(k)\, z^{-k}}$$

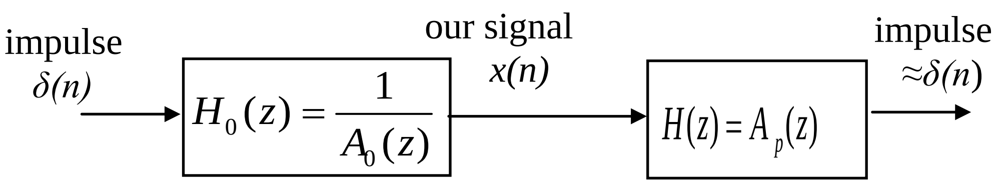
The output should be zero for all $n \neq 0$. **We define an error**

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

and we minimize

$$\varepsilon_p = \sum_{n=0}^{\infty} | e(n) |^2$$

This can be described by the following figure (*b(0)=1*).



$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k)\, z^{-k}$$

The solution derived in chapter 4 is

$$\sum_{l=1}^{p} a_p(l)\ r_x(k-l) = -r_x(k) \qquad k = 1,\dots,p$$

In matrix form (real signals)

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \dots & r_x(p-1) \\ r_x(1) & r_x(0) & r_x(1) & \dots & r_x(p-2) \\ r_x(2) & r_x(1) & r_x(0) & \dots & r_x(p-3) \\ \dots & & & & \\ r_x(p-1) & r_x(p-2) & r_x(p-3) & \dots & r_x(0) \end{bmatrix}}_{R_x} \underbrace{\begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \\ a_p(p) \end{bmatrix}}_{\bar{a}_p} = -\begin{bmatrix} r_x(1) \\ r_x(2) \\ r_x(3) \\ \\ r_x(p) \end{bmatrix}$$

The optimal coefficients can be found just by inverting the correlation matrix. The value of resulting minimum cost function was

$$\varepsilon_{p,\min} = \varepsilon_p = r_x(0) + \sum_{k=1}^{p} a(k)\ r_x\ (k)$$

The minimum cost is decreasing if the order p increases and can be used to chose an appropriate value of the order p. This is described more in chapter 8.

Combining the two equations into one matrix equation gives

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \dots & r_x(p) \\ r_x(1) & r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \dots & r_x(p-2) \\ r_x(3) & r_x(2) & r_x(1) & \dots & r_x(p-3) \\ \dots & & & & \\ r_x(p) & r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix}}_{R_x} \underbrace{\begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ a_p(3) \\ \\ a_p(p) \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} \varepsilon_p \\ 0 \\ 0 \\ 0 \\ \\ 0 \end{bmatrix}}_{\varepsilon_p u_1}$$

$$R_x\ a_p = \varepsilon_p u_1$$

We will now derive an iterative solution for these equations.

Levinson-Durbin recursion

The autocorrelation matrix for this system is Hermitian Toeplitz (symmetrical Toeplitz for real valued signals). For solving this types of matrix equation, a very well known algorithm is the Levinson-Durbin recursion. We here assume real valued signals (for complex signal, see the textbook).

We start with the normal equation from chapter 4 (page 216-219)

$$r_x(k) + \sum_{l=1}^{p} a_p(l)\; r_k(l-k) = 0; \qquad k = 1, 2 \ldots p$$

and the error

$$\varepsilon_p = r_x(0) + \sum_{l=1}^{p} a(l)\; r_x(l)$$

In matrix form (index *p* denotes the order of the filter)

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(p) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x(p-1) \\ & & \cdots & & \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix}}_{R_p} \cdot \underbrace{\begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \cdot \\ a_p(p) \end{bmatrix}}_{a_p} = \underbrace{\begin{bmatrix} \varepsilon_p \\ 0 \\ 0 \\ \cdot \\ 0 \end{bmatrix}}_{\varepsilon_p u_1}$$

$$R_p\, a_p = \varepsilon_p u_1$$

Now we will derive an algorithm to solve this iteratively. The idea is to solve it in a recursive procedure starting a_1, then a_2, a_3, up to a_p.

Then, in step j we have

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(j) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(j-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x(j-2) \\ & & \cdots & & \\ r_x(j) & r_x(j-1) & r_x(j-2) & \cdots & r_x(0) \end{bmatrix}}_{R_j} \cdot \underbrace{\begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \cdot \\ a_j(j) \end{bmatrix}}_{a_j} = \underbrace{\begin{bmatrix} \varepsilon_j \\ 0 \\ 0 \\ \cdot \\ 0 \end{bmatrix}}_{\varepsilon_j u_1}$$

$$R_j\, a_j = \varepsilon_j u_1$$

Add one row and one column including the following equation

$$\gamma_j = r_x(j+1) + \sum_{i=1}^{j} a_j(i) r_x(j+1-i)$$

The new matrix is

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(j+1) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(j) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x(j-1) \\ & & \cdots & & \\ r_x(j) & r_x(j-1) & r_x(j-2) & \cdots & r_x(1) \\ r_x(j+1) & r_x(j) & r_x(j-1) & \cdots & r_x(0) \end{bmatrix}}_{R_{j+1}} \cdot \underbrace{\begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \cdot \\ a_j(j) \\ 0 \end{bmatrix}} = \underbrace{\begin{bmatrix} \varepsilon_j \\ 0 \\ 0 \\ \cdot \\ 0 \\ \gamma_j \end{bmatrix}}$$

Use the symmetry to write this as

$$\left(R_j = R_j^T, \;\; symmetrical \right)$$

$$\underbrace{\begin{bmatrix} r_x(0) & r_x(1) & r_x(2) & \cdots & r_x(j+1) \\ r_x(1) & r_x(0) & r_x(1) & \cdots & r_x(j) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x(j-1) \\ & & \cdots & & \\ r_x(j) & r_x(j-1) & r_x(j-2) & \cdots & r_x(1) \\ r_x(j+1) & r_x(j) & r_x(j-1) & \cdots & r_x(0) \end{bmatrix}} \cdot \underbrace{\begin{bmatrix} 0 \\ a_j(j) \\ a_j(j-1) \\ \cdot \\ a_j(1) \\ 1 \end{bmatrix}} = \underline{\begin{bmatrix} \gamma_j \\ 0 \\ 0 \\ \cdot \\ 0 \\ \varepsilon_j \end{bmatrix}}$$

Now, make a linear combination of these two equations

$$R_{j+1}\underbrace{\left\{\cdot\underbrace{\begin{bmatrix}1\\a_j(1)\\a_j(2)\\ \cdot\\a_j(j)\\0\end{bmatrix}}+\Gamma_{j+1}\begin{bmatrix}0\\a_j(j)\\a_j(j-1)\\ \cdot\\a_j(1)\\1\end{bmatrix}\right\}}_{a_{j+1}}=\underbrace{\begin{bmatrix}\varepsilon_j\\0\\0\\ \cdot\\0\\\gamma_j\end{bmatrix}+\Gamma_{j+1}\begin{bmatrix}\gamma_j\\0\\0\\ \cdot\\0\\\varepsilon_j\end{bmatrix}}_{\varepsilon_{j+1}u_1}$$

This new matrix equation must satisfy

$$R_{j+1}\,a_{j+1}=\varepsilon_{j+1}u_1$$

Then, the lowest element in the vector on the right side must be zero.

Then we get

$$\gamma_j+\Gamma_{j+1}\;\varepsilon_j=0$$

$$\Gamma_{j+1}=-\frac{\gamma_j}{\varepsilon_j}$$

and also $\varepsilon_{j+1}=\varepsilon_j+\Gamma_{j+1}\;\gamma_j=\varepsilon_j\,(1-|\,\Gamma_{j+1}|^2)$

This results in the update equation for *a*

$$\begin{bmatrix} 1 \\ a_{j+1}(1) \\ a_{j+1}(2) \\ . \\ a_{j+1}(j) \\ a_{j+1}(j+1) \end{bmatrix} = \begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ . \\ a_j(j) \\ 0 \end{bmatrix} + \Gamma_{j+1} \begin{bmatrix} 0 \\ a_j(j) \\ a_j(j-1) \\ . \\ a_j(1) \\ 1 \end{bmatrix}$$

Alternatively, we can write

$$a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j(j-i+1) \qquad i = 1, 2, \ldots, j+1$$

Note that

$$a_j(0) = 1$$

$$a_j(j+1) = 0$$

$$a_{j+1}(j+1) = \Gamma_{j+1}$$

$$\varepsilon_0 = r_x(0)$$

This algorithm is easy to implement in a computer program. This is shown in table 5.1 in the textbook. The parameters Γ_j are called the reflection parameters. For stable filters (all poles inside the unit circle), the reflection parameters are bounded by

$$|\Gamma_j| < 1.$$

Table 5.1 ***The Levinson-Durbin Recursion***

1. Initialize the recursion
   (a) $a_0(0) = 1$
   (b) $\epsilon_0 = r_x(0)$
2. For $j = 0, 1, \ldots, p-1$
   (a) $\gamma_j = r_x(j+1) + \sum_{i=1}^{j} a_j(i) r_x(j-i+1)$
   (b) $\Gamma_{j+1} = -\gamma_j / \epsilon_j$
   (c) For $i = 1, 2, \ldots, j$
   $$a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j^*(j-i+1)$$
   (d) $a_{j+1}(j+1) = \Gamma_{j+1}$
   (e) $\epsilon_{j+1} = \epsilon_j \left[1 - |\Gamma_{j+1}|^2\right]$
3. $b(0) = \sqrt{\epsilon_p}$

The relation between a_j and Γ_j can be written as (page 234)

$$a_0 = 1$$

$$a_1 = \begin{bmatrix} a_1(0) \\ a_1(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \Gamma_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \Gamma_1 \end{bmatrix}$$

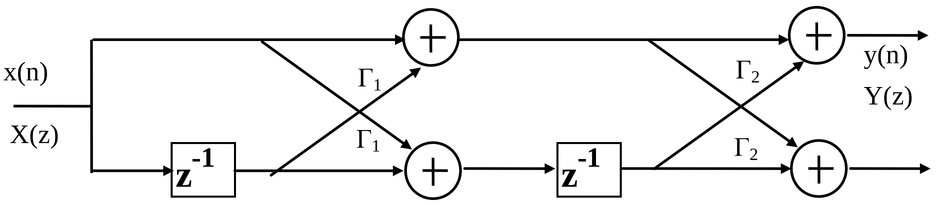
$$a_2 = \begin{bmatrix} a_2(0) \\ a_2(1) \\ a_2(2) \end{bmatrix} = \begin{bmatrix} 1 \\ a_1(1) \\ 0 \end{bmatrix} + \Gamma_2 \begin{bmatrix} 0 \\ a_1(1) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \Gamma_1 \\ 0 \end{bmatrix} + \Gamma_2 \begin{bmatrix} 0 \\ \Gamma_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \Gamma_1 + \Gamma_1 \cdot \Gamma_2 \\ \Gamma_2 \end{bmatrix}$$

$$a_3 = \begin{bmatrix} a_3(0) \\ a_3(1) \\ a_3(2) \\ a_3(2) \end{bmatrix} = \begin{bmatrix} 1 \\ a_2(1) \\ a_2(2) \\ 0 \end{bmatrix} + \Gamma_3 \begin{bmatrix} 0 \\ a_2(2) \\ a_2(1) \\ 1 \end{bmatrix}$$

Lattice filter

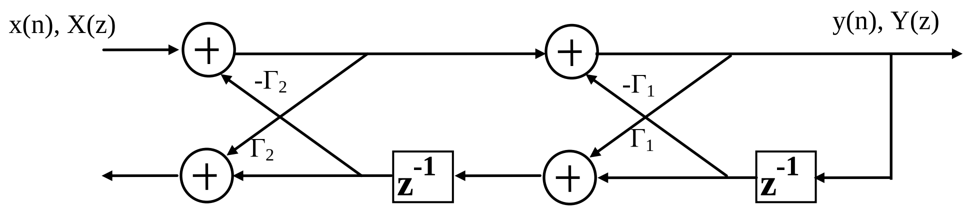
The parameters Γ_j can be interpreted in a specific structure of digital filters, called the lattice filters (see page 225 and chapter 6).

a) Second order FIR-lattice filter



$$H(z) = 1 + (\Gamma_1 + \Gamma_1\Gamma_2)z^{-1} + \Gamma_2 z^{-2}$$

b) Second order IIR-lattice filter



$$H(z) = \frac{1}{1 + (\Gamma_1 + \Gamma_1\Gamma_2)z^{-1} + \Gamma_2 z^{-2}}$$

Relation between the polynomial a_j and the reflection coefficients Γ_j using z-transform

We have $a_j = \begin{bmatrix} 1 & a_j(1) & a_j(2) \cdots a_j(j) \end{bmatrix}^T$

Then define $a_j^R = \begin{bmatrix} a_j(j) & \cdots & a_j(2) & a_j(1) & 1 \end{bmatrix}^T$

Then we can write the update equation (page 224, 235, 236)

$$a_{j+1}(i) = a_j(i) + \Gamma_{j+1}\, a_j^R(i-1)$$

Make a variable substitution $i \Rightarrow j-i+1$ **gives**

$$\underbrace{a_{j+1}(j-i+1)}_{a_{j+1}^R(i)} = \underbrace{a_j(j-i+1)}_{\hat{a}_j^R(i-1)} + \Gamma_{j+1}\, \underbrace{a_j^R(j-i)}_{a_j(i)}$$

Taking the z-transform of these equations gives

Forwards: (from gamma to polynomial)

$$\begin{cases} A_{j+1}(z) = A_j(z) + z^{-1}\,\Gamma_{j+1}\,A_j^R(z) \\ A_{j+1}^R(z) = z^{-1}\,A_j^R + \Gamma_{j+1}\,A_j(z) \end{cases}$$

In matrix form this can be written (page 224 and page 236)

$$\begin{bmatrix} A_{j+1}(z) \\ A_{j+1}^R(z) \end{bmatrix} = \begin{bmatrix} 1 & z^{-1}\,\Gamma_{j+1} \\ \Gamma_{j+1} & z^{-1} \end{bmatrix} \begin{bmatrix} A_j(z) \\ A_j^R(z) \end{bmatrix}$$

Backwards: (From polynomial to gamma)

$$\begin{bmatrix} A_j(z) \\ A_j^R(z) \end{bmatrix} = \frac{1}{z^{-1}(1-|\Gamma_{j+1}|^2)} \begin{bmatrix} z^{-1} & -z^{-1}\,\Gamma_{j+1} \\ -\Gamma_{j+1} & 1 \end{bmatrix} \begin{bmatrix} A_{j+1}(z) \\ A_{j+1}^R(z) \end{bmatrix}$$

$$A_j(z) = \frac{1}{(1-|\Gamma_{j+1}|^2)}\left[A_{j+1}(z) - \Gamma_{j+1}A_{j+1}^R(z)\right]$$

Backwards iteratively

If the filter is given by $a_p(n)$, the reflection parameters Γ_j can be determined, see textbook page 235, page 236, table 5.3.

The step down recursion (see table 5.3)

Identify $\Gamma_p = a_p(p)$

Loop j=p-1, p-2, ... 1

Then, determine a_j from a_{j+1}

$$a_j(i) = \frac{1}{1-\Gamma_{j+1}^2}\left[a_{j+1}(i) - \Gamma_{j+1}\; a_{j+1}(j-i+1)\right] \quad i = 1, 2, ..., j$$

Identify $\Gamma_j = a_j(j)$

Table 5.3 *The Step-down Recursion*

1. Set $\Gamma_p = a_p(p)$
2. For $j = p-1, p-2, \ldots, 1$

   (a) For $i = 1, 2, \ldots, j$

   $$a_j(i) = \frac{1}{1-|\Gamma_{j+1}|^2}\left[a_{j+1}(i) - \Gamma_{j+1}a^*_{j+1}(j-i+1)\right]$$

   (b) Set $\Gamma_j = a_j(j)$

   (c) If $|\Gamma_j| = 1$, Quit.
3. $\epsilon_p = b^2(0)$

The Levinson-Durbin algorithm determine relation between autocorrelation *r(x)*, polynomial *a(k)* and the reflection coefficients. This can be summarized in the figure below and in the table 5.1 –5.4.

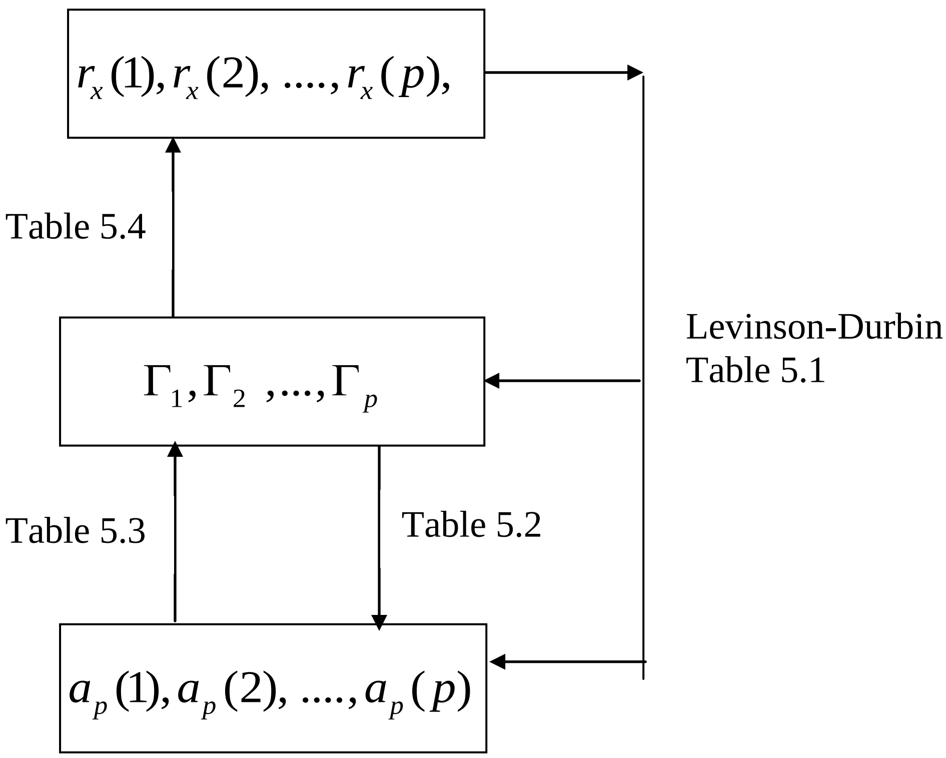


Table 5.4 ***The Inverse Levinson-Durbin Recursion***

1. Initialize the recursion
   (a) $r_x(0) = \epsilon_p / \prod_{i=1}^{p}(1 - |\Gamma_i|^2)$
   (b) $a_0(0) = 1$
2. For $j = 0, 1, \ldots, p-1$
   (a) For $i = 1, 2, \ldots, j$
   $a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j^*(j - i + 1)$
   (b) $a_{j+1}(j+1) = \Gamma_{j+1}$
   (c) $r_x(j+1) = -\sum_{i=1}^{j+1} a_{j+1}(i) r_x(j+1-i)$
3. Done

Table 5.2 ***The Step-up Recursion***

1. Initialize the recursion: $a_0(0) = 1$
2. For $j = 0, 1 \ldots, p-1$
   (a) For $i = 1, 2, , \ldots, j$
   $a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j^*(j - i + 1)$
   (b) $a_{j+1}(j+1) = \Gamma_{j+1}$
3. $b(0) = \sqrt{\epsilon_p}$