# Statistical Digital Signal Processing

## Week 5

## Levinson-Durbin Recursion and Lattice Filters

**Lecturer:** ***Zelalem Hailu (Assistant Prof.)***

*Addis Ababa Science and Technology University,*
*Addis Ababa, Ethiopia*

# Previous Topic (Week-4)

## Stochastic Signal Modeling

- Stochastic Signal Modeling : Introduction
- Autoregressive Moving Average Model (ARMA)
- Autoregressive Model (AR)

# Lecture Learning Outcomes

1. Explain the fundamental concept and purpose of the Levinson-Durbin recursion in solving autocorrelation equations.
2. Derive the Levinson-Durbin recursion algorithm step-by-step from the normal equations of all-pole modeling.

3. Analyze the computational complexity of the Levinson-Durbin algorithm and compare it with Gaussian elimination method.
4. Describe the structure and operation of lattice filters and their relationship to the Levinson-Durbin recursion.
5. Interpret the properties of reflection coefficients and evaluate their role in ensuring stability and efficient signal representation.

# Week 5: Levinson-Durbin Recursion and Lattice Filters

## Outline

- Levinson-Durbin Recursion : Introduction
- Development of Levinson-Durbin Recursion
- Computational Complexity of Levinson-Durbin Recursion
- Lattice Filters
- Properties of The Reflection Coefficient

# Levinson-Durbin Recursion : Introduction

- In the previous sessions of Signal Modeling, we have seen different types of signal modeling problems require solving set of linear equations having the form:

$$R_x a_p = b \qquad (1)$$

**Where:**

$R_x$ Is Toeplitz matrix

- Toeplitz matrix or equations have been observed in pade approximation, Proney method, Shank Method, and modified Yule-Wlaker equations in ARMA process
- Solving the Toeplitz equations is required in a number of signal modeling problems
- Therefore, There is a need of efficient algorithm such as **Levinson-Durbin Recursion** to solve those Toeplitz equations

# Levinson-Durbin Recursion : Introduction

- In 1947, N. Levinson introduced a recursive algorithm to solve symmetric Toeplitz linear equations $\mathrm{R}_x \mathrm{a} = \mathrm{b}$
- **Levinson Algorithm:** A fast recursive method for solving the normal equations in signal modeling [1] (e.g., Prony's all-pole method), enabling efficient inversion of Toeplitz matrices
- Later, in 1961, Durbin improved the Levison recursion for special case in which the right hand side of Toeplitz equation is unit vector
- It leads to a lattice filter structure with useful computational and stability properties
- It enables analysis of an autocorrelation sequence and synthesis of a process with the same autocorrelation

# Development of Levinson-Durbin Recursion

- All-pole modeling using Prony's or the autocorrelation method requires solving normal equations
- For $p^{th}$ order model, we have:

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

- The modeling error is given by:

$$\varepsilon_p = r_x(0) + \sum_{l=1}^{p} a_p(l) r_x^*(l) \qquad (3)$$

**Where:**

$\varepsilon_p$ Is the modeling error

- The filter nominator coefficient also given by $b_0(0) = \sqrt{\varepsilon_p}$

# Development of Levinson-Durbin Recursion

- Writing eq(2) in matrix form:

$$\begin{bmatrix} r_x(1) & r_x(0) & r_x(-1) & \cdots & r_x(1-p) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x(2-p) \\ r_x(3) & r_x(2) & r_x(1) & \cdots & r_x(3-p) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4)$$

- Using the property $r_x(-k) = r_x^*(k)$, eq(4) can be written as:

$$\begin{bmatrix} r_x(1) & r_x(0) & r_x^*(1) & \cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0) & \cdots & r_x^*(p-2) \\ r_x(3) & r_x(2) & r_x(1) & \cdots & r_x^*(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5)$$

# Development of Levinson-Durbin Recursion

- Casting eq(3) in eq(5) , we will have:

$$\underbrace{\begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2)\cdots & r_x^*(p) \\ r_x(1) & r_x(0) & r_x^*(1)\cdots & r_x^*(p-1) \\ r_x(2) & r_x(1) & r_x(0)\cdots & r_x^*(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_x(p) & r_x(p-1) & r_x(p-2)\cdots & r_x(0) \end{bmatrix}}_{\mathrm{R}_p} \underbrace{\begin{bmatrix} 1 \\ a_p(1) \\ a_p(2) \\ \vdots \\ a_p(p) \end{bmatrix}}_{\mathrm{a}_p} = \varepsilon_p \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathrm{u}_1} \qquad (6)$$

- Writing eq(6) in compact form:

$$\mathrm{R}_p \mathrm{a}_p = \varepsilon_p \mathrm{u}_1 \qquad (7)$$

**Where:**

$\mathrm{R}_p \rightarrow (p+1)\times(p+1)$ Hermitian Toeplitz matrix

$\mathrm{u}_1 \rightarrow$ Unit vector with 1 in the first position

# Development of Levinson-Durbin Recursion

- Levinson-Durbin Recursion algorithm is used to solve eq (7) through recursive method in the model order.
- Using Levinson-Durbin Recursion algorithm, If the order-j solution $(a_j(1), a_j(2), ..., a_j(j))$ is known, the order-j+1 solution can be found from the order-j solution
- Suppose $\mathrm{a}_j$ is known as follows, we want to drive the solution to the following j+1 order normal equations:

$$\begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2)\cdots & r_x^*(j) \\ r_x(1) & r_x(0) & r_x^*(1)\cdots & r_x^*(j-1) \\ r_x(2) & r_x(1) & r_x(0)\cdots & r_x^*(j-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_x(j) & r_x(j-1) & r_x(j-2)\cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \vdots \\ a_j(j) \end{bmatrix} = \begin{bmatrix} \varepsilon_j \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (8)$$

- The derivation is performed by appending zero to the vector $\mathrm{a}_j$ and multiplying by $\mathrm{R}_{j+1}$ as follows

# Development of Levinson-Durbin Recursion

$$\underbrace{\begin{bmatrix} r_x(0) & r_x^*(1) & r_x^*(2)\cdots & r_x^*(j+1) \\ r_x(1) & r_x(0) & r_x^*(1)\cdots & r_x^*(j) \\ r_x(2) & r_x(1) & r_x(0)\cdots & r_x^*(j-1) \\ \vdots & \vdots & \vdots & \vdots \\ r_x(j) & r_x(j-1) & r_x(j-2)\cdots & r_1^*(1) \\ r_x(j+1) & r_x(j) & r_x(j-1)\cdots & r_x(0) \end{bmatrix}}_{\mathrm{R}_{j+1}} \begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \vdots \\ a_j(j) \\ 0 \end{bmatrix} = \begin{bmatrix} \varepsilon_j \\ 0 \\ 0 \\ \vdots \\ 0 \\ \gamma_j \end{bmatrix} \qquad (9)$$

**Where:**

$$\gamma_j = r_x(j+1) + \sum_{i=1}^{j} a_j(i) r_x(j+1-i) \qquad (10)$$

# Development of Levinson-Durbin Recursion

- From the Hermitian Toeplitz property of $R_{j+1}$ allows us to rewrite eq (9) in the following equivalent form:

$$\begin{bmatrix} r_x(0) & r_x(1) & r_x(2)\cdots & r_x(j+1) \\ r_x^*(1) & r_x(0) & r_x(1)\cdots & r_x(j) \\ r_x^*(2) & r_x^*(1) & r_x(0)\cdots & r_x(j-1) \\ \vdots & \vdots & \vdots & \vdots \\ r_x^*(j) & r_x^*(j-1) & r_x^*(j-2)\cdots & r_x(1) \\ r_x^*(j+1) & r_x^*(j) & r_x^*(j-1)\cdots & r_x(0) \end{bmatrix} \begin{bmatrix} 0 \\ a_j(j) \\ a_j(j-1) \\ \vdots \\ a_j(1) \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma_j \\ 0 \\ 0 \\ \vdots \\ 0 \\ \varepsilon_j \end{bmatrix} \qquad (11)$$

# Development of Levinson-Durbin Recursion

- Taking the complex conjugate of eq(11) and combining with eq(9), we will have

$$R_{j+1}\underbrace{\left[\begin{bmatrix}1\\a_j(1)\\a_j(2)\\\vdots\\a_j(j)\\0\end{bmatrix}+\Gamma_{j+1}\begin{bmatrix}0\\a_j^*(1)\\a_j^*(2)\\\vdots\\a_j^*(j)\\1\end{bmatrix}\right]}_{a_{j+1}}=\underbrace{\begin{bmatrix}\varepsilon_j\\0\\0\\\vdots\\0\\\gamma_j\end{bmatrix}+\Gamma_{j+1}\begin{bmatrix}\gamma_j^*\\0\\0\\\vdots\\0\\\varepsilon_j^*\end{bmatrix}}_{\varepsilon_{j+1}u_1} \qquad (12)$$

Solution for the (j+1)-order ➡ $a_{j+1}$

**Where:**

$$\Gamma_{j+1}=-\frac{\gamma_j}{\varepsilon_j^*} \qquad (13)$$

# Development of Levinson-Durbin Recursion

- Writing eq(12) in compact form, we have:

$$\mathrm{R}_{j+1}\mathrm{a}_{j+1} = \varepsilon_{j+1}\mathrm{u}_1 \qquad (14)$$

- The Solution for the (p+1)st-order becomes:

$$\mathrm{a}_{j+1} = \begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \vdots \\ a_j(j) \\ 0 \end{bmatrix} + \Gamma_{j+1} \begin{bmatrix} 0 \\ a_j^*(1) \\ a_j^*(2) \\ \vdots \\ a_j^*(j) \\ 1 \end{bmatrix} \qquad (15)$$

# Development of Levinson-Durbin Recursion

- From eq(12), we can write the (j+1)-order modeling error, $\varepsilon_{j+1}$ , as:

$$\varepsilon_{j+1} = \varepsilon_j + \Gamma_{j+1} \gamma_j^* = \varepsilon_j \left[ 1 - |\Gamma_{j+1}|^2 \right] \qquad (16)$$

- If we define $a_j(0) = 1$ and $a_j(j+1) = 0$ , the Levinson order-update equation can be expressed as:

$$a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j^*(j-i+1) \; ; \; i = 0,1,2,...,j+1 \qquad (17)$$

- To complete the recursion, it is also required to initialize by setting the solution for 0th order model as follows:

$$a_0(0) = 1 \qquad (18)$$

$$\varepsilon_0 = r_x(0) \qquad (19)$$

# Development of Levinson-Durbin Recursion

- The steps of the Levinson-Durbin recursion Algorithm are as follows :

```
1. Initialize the recursion
    (a). $a_0(0) = 1$
    (b). $\varepsilon_0(0) = r_x(0)$
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). $\Gamma_{j+1} = -\gamma_j \big/ \varepsilon_j^*$
    (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). $\varepsilon_{j+1} = \varepsilon_j \left[ 1 - |\Gamma_{j+1}|^2 \right]$
3. $b(0) = \sqrt{\varepsilon_{j+1}}$
```

# Development of Levinson-Durbin Recursion

- As shown in the steps of the Levinson-Durbin recursion Algorithm, first the recursion is initialized with zero-order solution as given in eq (18 & 19)
- Then, the (j+1)-order model is found from jth-order model in three steps for $j = 0, 1, \ldots, p-1$
- The first step uses eq(10) and eq(13) to find

$$\Gamma_{j+1} \rightarrow (j+1)st \text{ Reflection Coefficent}$$

- The next step is to used Levinson order-update equation to compute $a_{j+1}$ from $a_j$
- The final step is to update the error $\varepsilon_{j+1}$ using eq(16)
- In addition to eq(16), the error $\varepsilon_{j+1}$ can also be written in the following two alternative forms:

$$\varepsilon_{j+1} = \varepsilon_j \left[ 1 - |\Gamma_{j+1}|^2 \right] = r_x(0) \prod_{i=1}^{j+1} \left[ 1 - |\Gamma_i|^2 \right] \qquad (20)$$

$$\varepsilon_{j+1} = r_x(0) + \sum_{i=1}^{j+1} a_{j+1}(i) r_x(i) \qquad (21)$$

# Computational Complexity of Levinson-Durbin Recursion

- Compared to Gaussian elimination method for solving p*th*-order autocorrelation normal equation having p linear equations with p unknowns:

**Gaussian elimination** → $\frac{1}{3}p^3$ Multiplications & Divisions

**Levinson-Durbin** → 
$$\begin{cases} \sum_{j=0}^{p-1}(2j+3) = p^2 + 2p & \text{Multiplications \& Divisions} \\ \sum_{j=0}^{p-1}(2j+1) = p^2 & \text{Additions} \end{cases}$$

- The number of multiplications & divisions proportional to $p^2$ for Levinson compared to $p^3$ for gaussian elimination which shows the advantage of Levinson

# Computational Complexity of Levinson-Durbin Recursion

- Another advantage of Levinson is the requirement of less memory for data storage as follows :

**Gaussian elimination** $\longrightarrow$ $p^2$ Memory Locations

**Levinson-Durbin** $\longrightarrow$ $2p+1$ Memory Locations $= \begin{cases} p+1 & \text{for } r_x(0), r_x(1), \ldots, r_x(p) \\ p & \text{for } a_p(1), a_p(2), \ldots, a_p(p) \\ 1 & \text{for } \varepsilon_p \end{cases}$

- However, It is also worth to note that solving the normal equations contributes small fraction of the total complexity in modeling process **(E.g. additional computations for computing the autocorrelations from the data)**

# Levinson-Durbin Recursion: Example

- Apply the Levinson–Durbin recursion to solve the autocorrelation (normal) equations and obtain a third-order all-pole model for the signal with the following autocorrelations [2]:

$$r_x(0) = 1, \quad r_x(1) = 0.5, \quad r_x(2) = 0.5, \quad r_x(3) = 0.25$$

**Solution:** The normal equations for the third-order all-pole model:

$$\begin{pmatrix} r_x(1) & r_x(0) & r^*(1) & r^*(2) \\ r_x(2) & r_x(1) & r_x(0) & r^*(1) \\ r_x(3) & r_x(2) & r_x(1) & r_x(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_p(1) \\ a_p(2) \\ a_p(3) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} r_x(0) & r^*(1) & r^*(2) \\ r_x(1) & r_x(0) & r^*(1) \\ r_x(2) & r_x(1) & r_x(0) \end{pmatrix} \begin{pmatrix} a_3(1) \\ a_3(2) \\ a_3(3) \end{pmatrix} = - \begin{pmatrix} r_x(1) \\ r_x(2) \\ r_x(3) \end{pmatrix}$$

# Levinson-Durbin Recursion: Example

- Substituting the autocorrelation values:

$$\begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}\begin{pmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \end{pmatrix} = -\begin{pmatrix} 0.5 \\ 0.5 \\ 0.25 \end{pmatrix}$$

- Using the Levinson-Durbin recursion we will start by driving the First-Order Model

### ❖ First-Order Model

$$\gamma_0 = r_x(1)$$

$$\Gamma_1 = -\frac{\gamma_0}{\varepsilon_0} = -\frac{r_x(1)}{r_x(0)} = -\frac{1}{2} \quad ; \qquad \varepsilon_1 = r_x(0)\left[1-\Gamma_1^2\right] = \frac{3}{4}$$

and

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ \Gamma_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1/2 \end{bmatrix}$$

# Levinson-Durbin Recursion: Example

- **Second-Order Model**

$$\gamma_1 = r_x(2) + a_1(1) r_x(1) = \frac{1}{4}$$

$$\Gamma_2 = -\frac{\gamma_1}{\varepsilon_1} = -\frac{1}{3}$$

$$\varepsilon_2 = \varepsilon_1 \left[ 1 - \Gamma_2^2 \right] = \frac{2}{3}$$

and

$$\mathbf{a}_2 = \begin{bmatrix} 1 \\ a_1 \\ 0 \end{bmatrix} + \Gamma_2 \begin{bmatrix} 0 \\ a_1^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1/2 \\ 0 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 0 \\ -1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1/3 \\ -1/3 \end{bmatrix}$$

# Levinson-Durbin Recursion: Example

## ❖ Third-Order Model

$$\gamma_2 = r_x(3) + a_2(1) r_x(2) + a_2(2) r_x(1) = -\frac{1}{12}$$

$$\Gamma_3 = -\frac{\gamma_2}{\varepsilon_2} = \frac{1}{8}$$

$$\varepsilon_3 = \varepsilon_2 \left[ 1 - \Gamma_3^2 \right] = \frac{21}{32}$$

and

$$\mathbf{a}_3 = \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ 0 \end{bmatrix} + \Gamma_3 \begin{bmatrix} 0 \\ a_2^* \\ a_1^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1/3 \\ -1/3 \\ 0 \end{bmatrix} + \frac{1}{8} \begin{bmatrix} 0 \\ -1/3 \\ -1/3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -3/8 \\ -3/8 \\ 1/8 \end{bmatrix}$$

# Levinson-Durbin Recursion: Example

- Finding $b(0)$ at last:

$$b(0) = \sqrt{\varepsilon_3} = \frac{1}{8}\sqrt{42}$$

- Then the third-order all pole model becomes

$$H_3(z) = \frac{\frac{\sqrt{42}}{8}}{1 - \frac{3}{8}z^{-1} - \frac{3}{8}z^{-2} + \frac{1}{8}z^{-3}}$$

# The Lattice Filter

- One of the important application of Levinson-Durbin recursion algorithm is the development of Lattice structure in digital filters
- Key notable features of the lattice structure include its **modular design** and **low sensitivity to parameter quantization effects**, making it robust and efficient for practical implementations
- Levinson order-update equation is used to drive the lattice filter structure for FIR digital filters
- The derivation of the lattice filter begins with the Levinson order-update equation given in **eq (17)**
- For the sake of convince, let's define the reciprocal vector $\mathrm{a}_j^R$ obtained by reversing the order of the elements in $\mathrm{a}_j$ and taking the complex conjugate as follows:

# The Lattice Filter

$$\mathrm{a}_j = \begin{bmatrix} 1 \\ a_j(1) \\ a_j(2) \\ \vdots \\ a_{j-1} \\ a_j(j) \end{bmatrix} \longrightarrow \begin{bmatrix} a_j^*(j) \\ a_j^*(j-1) \\ a_j^*(j-2) \\ \vdots \\ a_j^*(1) \\ 1 \end{bmatrix} = \mathrm{a}_j^R \qquad (22)$$

- Alternatively writing eq (22) in compact form:

$$a_j^R(i) = a_j^*(j-i) \qquad (23)$$

# The Lattice Filter

- For $i = 0, 1, 2, \ldots, j$, the Levinson order-update can be written the reciprocal vector as follows:

$$a_{j+1}(i) = a_j(i) + \Gamma_{j+1} a_j^R(i-1) \qquad (24)$$

- Taking the z-transform of $a_j^R(i)$, we obtain $A_j^R(z)$ from eq (23) as:

$$A_j^R(z) = z^{-j} A_j^*(1/z^*) \qquad (25)$$

- The order-update equation for $A_j(z)$ obtained by taking the z-transform of eq (24):

$$A_{j+1}(z) = A_j(z) + \Gamma_{j+1} \underbrace{z^{-1} A_j^R(z)}_{\Big\Uparrow \; \mathrm{Z}\left\{a_j^R(i-1)\right\}} \qquad (26)$$

# The Lattice Filter

- Next we will drive the order-update equations for $a_j^R(i)$ and $A_j^R(z)$ by taking the complex conjugate of eq (24) and replacing $i$ by $j-i+1$ as follows:

$$a^*_{j+1}(j-i+1) = a^*_j(j-i+1) + \Gamma^*_{j+1}\left[a_j^R(j-i)\right]^* \qquad (27)$$

- From eq (23), we have:

$$a_j^R(i) = a_j^*(j-i) \quad \Longrightarrow \quad a_j^R(j-i) = a_j^*(j-(j-i)) = a_j^*(i)$$

$$\Downarrow$$

$$\left[a_j^R(j-i)\right]^* = \left[a_j^*(i)\right]^* = a_j(i)$$

- Then eq (27) can be written as:

$$a^*_{j+1}(j-i+1) = a^*_j(j-i+1) + \Gamma^*_{j+1}a_j(i) \qquad (28)$$

# The Lattice Filter

- Incorporating the definition of reciprocal vector given in eq (23):

$$a_j^R(i) = a_j^*(j-i) \quad \Rightarrow \quad a_{j+1}^R(i) = a_{j+1}^*(j+1-i)$$

- Then, from eq (28) the order update equation for $a_{j+1}^R(i)$ can be obtained as follows:

$$a_{j+1}^R(i) = a_j^R(i-1) + \Gamma_{j+1}^* a_j(i) \qquad (29)$$

- Taking the z-transform of eq (29), the order update equation of $A_{j+1}^R(z)$ becomes:

$$A_{j+1}^R(z) = z^{-1} A_j^R(z) + \Gamma_{j+1}^* A_j(z) \qquad (30)$$

- In conclusion, we have the following set of coupled difference equations for updating $a_j(n)$ and $a_j^R(n)$:

$$\boxed{\begin{aligned} a_{j+1}(n) &= a_j(n) + \Gamma_{j+1} a_j^R(n-1) \\ a_{j+1}^R(n) &= a_j^R(n-1) + \Gamma_{j+1}^* a_j(n) \end{aligned}} \qquad (31)$$

# The Lattice Filter

- We have also coupled equations also used for updating $A_j^R(z)$ and $A_j^R(z)$ which can be written in matrix form as:

$$\begin{bmatrix} A_{j+1}(z) \\ A_{j+1}^R(z) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{j+1} z^{-1} \\ \Gamma_{j+1}^* & z^{-1} \end{bmatrix} \begin{bmatrix} A_j(z) \\ A_j^R(z) \end{bmatrix} \qquad (32)$$

- Both representations given in eq (31 & 32) describe the two-port network given in figure (1):

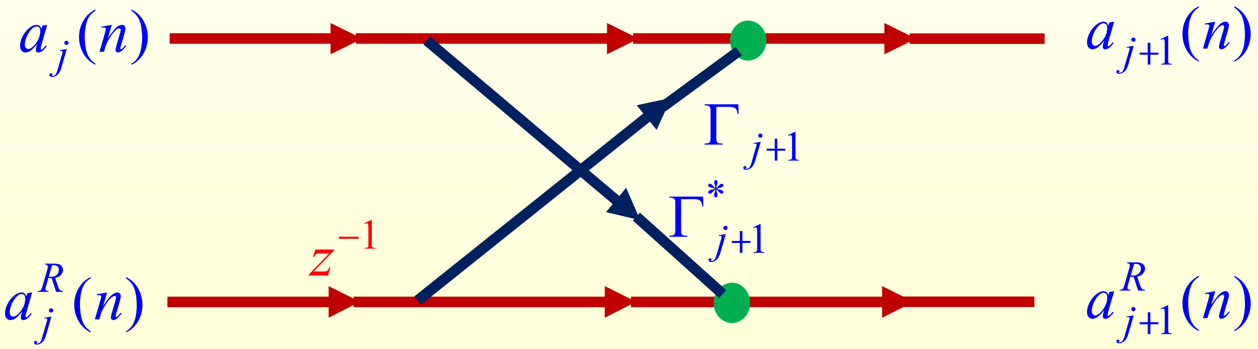

**Figure 1:** two port network

# The Lattice Filter

- This two-port network is the basic module used to implement an FIR lattice filter
- Cascading $p$ lattice filter modules, each with reflection coefficients $\Gamma_1, \Gamma_2, \ldots, \Gamma_p$ , results in a $p^{th}$ - order lattice filter

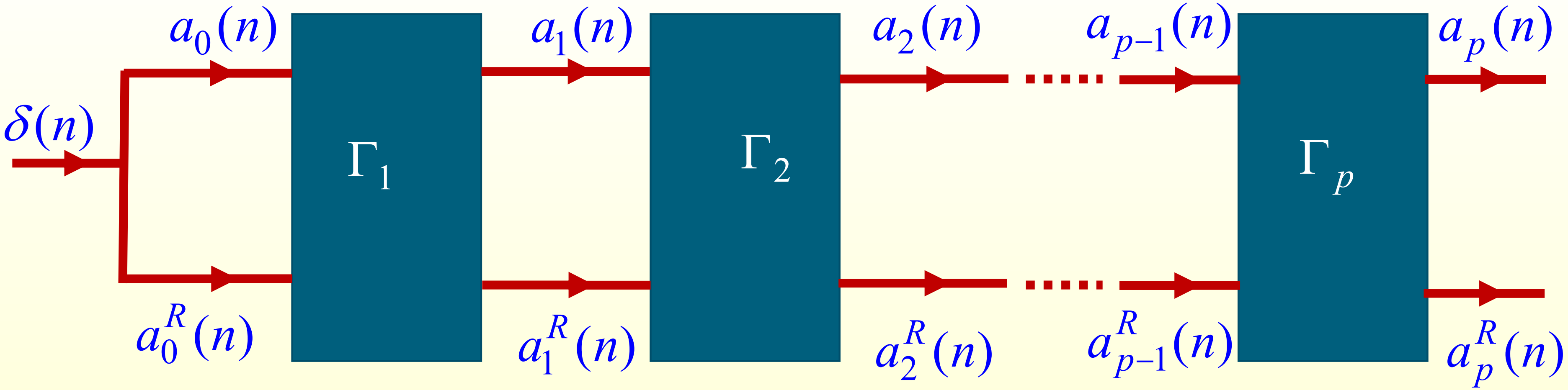

**Figure 2:** pth order lattice filter

# Properties of The Reflection Coefficient

## Property-1

- The reflection coefficients obtained from the Levinson–Durbin recursion, used to solve the autocorrelation normal equations, have magnitudes less than or equal to one: $|\Gamma_j| \le 1$
- The base of this property is the fact that $\varepsilon_j$ non negative or $\varepsilon_j \ge 0$

## Property-2

- If $a_p(k)$ represents a set of model parameters and $\Gamma_j$ the corresponding reflection coefficients, $A_p(z)$ is a minimum-phase polynomial (all its roots lie inside the unit circle) if and only if $|\Gamma_j| \le 1$ for all $j$

# Properties of The Reflection Coefficient

## Property-3

- If $\mathbf{a}_p$ is the solution to the Toeplitz normal equations $\mathbf{R}_p\mathbf{a}_p = \varepsilon_p\mathbf{u}_1$, then $A_p(z)$ is a minimum-phase polynomial if and only if $\mathbf{R}_p$ is positive definite ($\mathbf{R}_p > 0$)

## Property-4

- The autocorrelation method yields a stable all-pole model

## Property-5

- Let $a_p(k)$ be a set of filter coefficients and $\Gamma_j$ the corresponding reflection coefficients. If $|\Gamma_j| < 1$ for $j = 1, \ldots, p-1$ and $\Gamma_p = 1$, then the polynomial $A_p(z)$ has all its roots on the unit circle

# Properties of The Reflection Coefficient

## Property-6: Autocorrelation matching property

- If $b(0)$ satisfies the energy-matching constraint, then $b(0)=\sqrt{\varepsilon_p}$ , and the autocorrelation sequences of $x(n)$ and $h(n)$ are identical for $|k| \leq p$

# Summary

- **Levinson-Durbin Algorithm:**
  - ✓ Levinson Algorithm is a fast recursive method for solving the normal equations in signal modeling storage
  - ✓ Durbin improved the Levison recursion for special case in which the right hand side of Toeplitz equation is unit vector
  - ✓ Solving the Normal equations by beginning with a filter of order 0 and recursively generating filters of order 1, 2, 3, and so on, up to the desired order P
  - ✓ The computational complexity or effort is less than Gaussian elimination method and Matrix inversion
  - ✓ One of the important application of Levinson-Durbin recursion algorithm is the development of **Lattice structure** in digital filters

# References

[1] Charles W. Therrien,‘’ *Discrete Random Signals and Statistical Signal Processing”*, Prentice Hall, Pp.422-423, 1992.

[2] Monson H. Hayes, *“Statistical Digital Signal Processing and Modeling”,* John Wiley and sons, Pp.220-221, 1996.

Contents Here

# Thank You!