SGN 21006 Advanced Signal Processing: Lecture 6 Linear Prediction

Ioan Tabus

Department of Signal Processing
Tampere University of Technology
Finland

Outline

- Dealing with three notions: PREDICTION, PREDICTOR, PREDICTION ERROR;
- FORWARD versus BACKWARD: Predicting the future versus (improper terminology) predicting the past;
- Fast computation of AR parameters: Levinson – Durbin algorithm;
- New AR parametrization: Reflection coefficients;
- Lattice filters

References : Chapter 3 from *S. Haykin- Adaptive Filtering Theory - Prentice Hall, 2002.*

Notations, Definitions and Terminology

- Time series:

$$u(1), u(2), u(3), \ldots, u(n-1), u(n), u(n+1), \ldots$$

- Linear prediction of order M – FORWARD PREDICTION

$$\begin{aligned}\hat{u}(n) &= w_1 u(n-1) + w_2 u(n-2) + \ldots + w_M u(n-M) \\ &= \sum_{k=1}^{M} w_k u(n-k) = \underline{w}^T \underline{u}(n-1)\end{aligned}$$

- Regressor vector

$$\underline{u}(n-1) = \begin{bmatrix} u(n-1) & u(n-2) & \ldots & u(n-M) \end{bmatrix}^T$$

- Predictor vector of order M – FORWARD PREDICTOR

$$\begin{aligned}\underline{w} &= \begin{bmatrix} w_1 & w_2 & \ldots & w_M \end{bmatrix}^T \\ \underline{a}_M &= \begin{bmatrix} 1 & -w_1 & -w_2 & \ldots & -w_M \end{bmatrix}^T \\ &= \begin{bmatrix} a_{M,0} & a_{M,1} & a_{M,2} & \ldots & a_{M,M} \end{bmatrix}^T\end{aligned}$$

and thus $a_{M,0} = 1,\ a_{M,1} = -w_1,\ a_{M,2} = -w_2,\ \ldots,\ a_{M,M} = -w_M,$

- Prediction error of order M – FORWARD PREDICTION ERROR

$$f_M(n) = u(n) - \hat{u}(n) = u(n) - \underline{w}^T \underline{u}(n-1) = \underline{a}_M^T \begin{bmatrix} u(n) \\ \underline{u}(n-1) \end{bmatrix} = \underline{a}_M^T \underline{u}(n)$$

Notations, Definitions and Terminology

$$r(k) = E[u(n)u(n+k)] \quad \text{– autocorrelation function}$$

$$R = E[\underline{u}(n-1)\underline{u}^T(n-1)] = E\begin{bmatrix} u(n-1) \\ u(n-2) \\ u(n-3) \\ u(n-M) \end{bmatrix}\begin{bmatrix} u(n-1) & u(n-2) & u(n-3) & \dots & u(n-M) \end{bmatrix}$$

$$= \begin{bmatrix} Eu(n-1)u(n-1) & Eu(n-1)u(n-2) & \dots & Eu(n-1)u(n-M) \\ Eu(n-2)u(n-1) & Eu(n-2)u(n-2) & \dots & Eu(n-2)u(n-M) \\ \vdots & \vdots & \vdots & \vdots \\ Eu(n-M)u(n-1) & Eu(n-M)u(n-2) & \dots & Eu(n-M)u(n-M) \end{bmatrix} =$$

$$R = \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r(1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \vdots & \vdots \\ r(M-1) & r(M-2) & \dots & r(0) \end{bmatrix} \quad \text{– autocorrelation matrix}$$

$$\underline{r} = E[\underline{u}(n-1)u(n)] = \begin{bmatrix} r(1) & r(2) & r(3) & \dots & r(M) \end{bmatrix}^T \quad \text{– autocorrelation vector}$$

$$\underline{r}^B = E[\underline{u}(n-1)u(n-M-1)] = \begin{bmatrix} r(M) & r(M-1) & r(M-2) & \dots & r(1) \end{bmatrix}^T$$

– Superscript ^{B} is the vector reversing (Backward) operator. i.e. for any vector $\underline{x}$, we have

$$\underline{x}^B = \begin{bmatrix} x(1) & x(2) & x(3) & \dots & x(M) \end{bmatrix}^B = \begin{bmatrix} x(M) & x(M-1) & x(M-2) & \dots & x(1) \end{bmatrix}$$

Optimal forward linear prediction

- Optimality criterion

$$\begin{aligned} J(\underline{w}) &= E[f_M(n)]^2 = E[u(n) - \underline{w}^T \underline{u}(n-1)]^2 \\ J(\underline{a}_M) &= E[f_M(n)]^2 = E[\underline{a}_M^T \begin{bmatrix} u(n) \\ \underline{u}(n-1) \end{bmatrix}]^2 \end{aligned}$$

- Optimal solution:

Optimal Forward Predictor	$\underline{w}_o = R^{-1}\underline{r}$

Forward Prediction Error Power	$P_M = r(0) - \underline{r}^T \underline{w}_o$

- Two derivations of optimal solution
  1. Transforming the criterion into a quadratic form

$$\begin{aligned} J(\underline{w}) &= E[u(n) - \underline{w}^T \underline{u}(n-1)]^2 = E[u(n) - \underline{w}^T \underline{u}(n-1)][u(n) - \underline{u}(n-1)^T \underline{w}] \\ &= E[u(n)]^2 - 2E[u(n)\underline{u}(n-1)^T]\underline{w} + \underline{w}^T E[\underline{u}(n-1)\underline{u}(n-1)^T]\underline{w} \\ &= r(0) - 2\underline{r}^T \underline{w} + \underline{w}^T R \underline{w} \\ &= r(0) - \underline{r}^T R^{-1} \underline{r} + (\underline{w} - R^{-1}\underline{r})^T R (\underline{w} - R^{-1}\underline{r}) \qquad (1) \end{aligned}$$

Augmented Wiener Hopf equations

- The matrix R is positive semi-definite because

$$\underline{x}^T R \underline{x} = \underline{x}^T E[\underline{u}(n)\underline{u}(n)]^T \underline{x} = E[\underline{u}(n)^T \underline{x}]^2 \geq 0 \qquad \forall \underline{x}$$

and therefore the quadratic form in the right hand side of (1) $(\underline{w} - R^{-1}\underline{r})^T R(\underline{w} - R^{-1}\underline{r})$ attains its minimum when $(\underline{w}_o - R^{-1}\underline{r}) = 0$, i.e.

$$\underline{w}_o = R^{-1}\underline{r}$$

For the predictor $\underline{w}_o$, the optimal criterion in (1) equals

$$P_M = r(0) - \underline{r}^T R^{-1}\underline{r} = r(0) - \underline{r}^T \underline{w}_o$$

Augmented Wiener Hopf equations

- Derivation based on optimal Wiener filter design
  The optimal predictor evaluation can be rephrased as the following Wiener filter design problem:
  - find the FIR filtering process $y(n) = \underline{w}^T \underline{u}(n)$
  - "as close as possible" to desired signal $d(n) = u(n+1)$, i.e.
  - minimizing the criterion $E[d(n) - y(n)]^2 = E[u(n+1) - \underline{w}^T \underline{u}(n)]^2$

  Then the optimal solution is given by $\underline{w}_o = R^{-1}\underline{p}$ where $R = E[\underline{u}(n)\underline{u}(n)^T]$ and $\underline{p} = E[d(n)\underline{u}(n)] = E[u(n+1)\underline{u}(n)] = E[u(n)\underline{u}(n-1)] = \underline{r}$, i.e.

$$\underline{w}_o = R^{-1}\underline{r}$$

Augmented Wiener Hopf equations

The **optimal predictor filter** solution $\underline{w}_o$ and the **optimal prediction error power** satisfy

$$r(0) - \underline{r}^T \underline{w}_o = P_M$$
$$R\underline{w}_o - \underline{r} = 0$$

which can be written in a block matrix equation form

$$\begin{bmatrix} r(0) & \underline{r}^T \\ \underline{r} & R \end{bmatrix} \begin{bmatrix} 1 \\ -\underline{w}_o \end{bmatrix} = \begin{bmatrix} P_M \\ 0 \end{bmatrix}$$

With the previous notations

$$\begin{bmatrix} 1 \\ -\underline{w}_o \end{bmatrix} = \underline{a}_M$$

$$\begin{bmatrix} r(0) & \underline{r}^T \\ \underline{r} & R \end{bmatrix} = R_{M+1}$$ – Autocorrelation matrix of dimensions $(M+1) \times (M+1)$

Finally, the augmented Wiener Hopf equations for **optimal forward prediction error filter** are

$$R_{M+1}\underline{a}_M = \begin{bmatrix} P_M \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r(1) & r(0) & \dots & r(M-1) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ r(M) & r(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,0} \\ a_{M,1} \\ a_{M,2} \\ . \\ . \\ a_{M,M} \end{bmatrix} = \begin{bmatrix} P_M \\ 0 \\ 0 \\ . \\ . \\ 0 \end{bmatrix}$$

Whenever R_M is nonsingular, and $a_{M,0}$ is set to 1, there are unique solutions $\underline{a}_M$ and P_M.

Optimal backward linear prediction

- Linear backward prediction of order M – BACKWARD PREDICTION

$$\begin{aligned}\hat{u}^b(n-M) &= g_1 u(n) + g_2 u(n-1) + \ldots + g_M u(n-M+1) \\ &= \sum_{k=1}^{M} g_k u(n-k+1) = \underline{g}^T \underline{u}(n)\end{aligned}$$

where the BACKWARD PREDICTOR is

$$\underline{g} = \begin{bmatrix} g_1 & g_2 & \ldots & g_M \end{bmatrix}^T$$

- Backward prediction error of order M – BACKWARD PREDICTION ERROR

$$b_M(n) = u(n-M) - \hat{u}^b(n-M) = u(n-M) - \underline{g}^T \underline{u}(n)$$

- Optimality criterion

$$J^b(\underline{g}) = E[b_M(n)]^2 = E[u(n-M) - \underline{g}^T \underline{u}(n)]^2$$

Optimal backward linear prediction

- Optimal solution:

Optimal Backward Predictor	$\underline{g}_o = R^{-1}\underline{r}^B = \underline{w}_o^B$

Forward Prediction Error Power	$P_M = r(0) - (\underline{r}^B)^T \underline{g}_o = r(0) - \underline{r}^T \underline{w}_o$

- Derivation based on optimal Wiener filter design
  The optimal backward predictor evaluation can be rephrazed as the following Wiener filter design problem:
  - find the FIR filtering process $y(n) = \underline{g}^T \underline{u}(n)$
  - "as close as possible" to desired signal $d(n) = u(n - M)$, i.e.
  - minimizing the criterion $E[d(n) - y(n)]^2 = E[u(n - M) - \underline{g}^T \underline{u}(n)]^2$

  Then the optimal solution is given by $\underline{g}_o = R^{-1}\underline{p}$ where $R = E[\underline{u}(n)\underline{u}(n)^T]$ and
  $\underline{p} = E[d(n)\underline{u}(n)] = E[u(n - M)\underline{u}(n)] = E[u(n - M - 1)\underline{u}(n - 1)] = \underline{r}^B$, i.e.

$$\underline{g}_o = R^{-1}\underline{r}^B$$

  and the optimal criterion value is

$$J^b(\underline{g}_o) \quad = \quad E[b_M(n)]^2 = E[d(n)]^2 - \underline{g}_o^T R \underline{g}_o = E[d(n)]^2 - \underline{g}_o^T \underline{r}^B = r(0) - \underline{g}_o^T \underline{r}^B$$

Relations between Backward and Forward predictors

- Relations between Backward and Forward predictors

$$\boxed{\underline{g}_o = \underline{w}_o^B}$$

- Useful mathematical result:
  If the matrix R is Toeplitz, then for all vectors $\underline{x}$

$$\begin{aligned}(R\underline{x})^B &= R\underline{x}^B \\ (R\underline{x})_i^B &= (R\underline{x}^B)_i \\ (R\underline{x})_{M-i+1} &= (R\underline{x}^B)_i\end{aligned}$$

Proof:

$$\begin{aligned}(R\underline{x}^B)_i &= \sum_{j=1}^{M} R_{i,j} x_{M-j+1} = \sum_{j=1}^{M} r(i-j) x_{M-j+1} \overset{j=M-k+1}{=} \sum_{k=1}^{M} r(i-M+k-1) x_k \\ &= \sum_{k=1}^{M} R_{M-i+1,k} x_k = (R\underline{x})_{M-i+1} = (R\underline{x})_i^B\end{aligned}$$

Relations between Backward and Forward predictors

- The Forward and Backward optimal predictors are solutions of the systems

$$R\underline{w}_o = \underline{r}$$
$$R\underline{g}_o = \underline{r}^B$$

$$R\underline{g}_o = \underline{r}^B = (R\underline{w}_o)^B = R\underline{w}_o^B$$

and since R is supposed nonsingular, we have

$$\underline{g}_o = \underline{w}_o^B$$

Augmented Wiener-Hopf equations for Backward prediction

- Augmented Wiener-Hopf equations for Backward prediction error filter
  The **optimal Backward predictor filter** solution $\underline{g}_o$ and the **optimal Backward prediction error power** satisfy

$$R\underline{g}_o - \underline{r}^B = 0$$
$$r(0) - (\underline{r}^B)^T \underline{g}_o = P_M$$

which can be written in a block matrix equation form

$$\begin{bmatrix} R & \underline{r}^B \\ (\underline{r}^B)^T & r(0) \end{bmatrix} \begin{bmatrix} -\underline{g}_o \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ P_M \end{bmatrix}$$

where we can identify the factors as:

$$\begin{bmatrix} -\underline{g}_o \\ 1 \end{bmatrix} = \underline{c}_M$$

$$\begin{bmatrix} R & \underline{r}^B \\ (\underline{r}^B)^T & r(0) \end{bmatrix} = R_{M+1}$$ – Autocorrelation matrix of dimensions $(M+1) \times (M+1)$

Augmented Wiener-Hopf equations for Backward prediction

Finally, the augmented Wiener Hopf equations for **optimal backward prediction error filter** are

$$R_{M+1}\underline{c}_M = \begin{bmatrix} 0 \\ P_M \end{bmatrix}$$

or

$$\begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r(1) & r(0) & \dots & r(M-1) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ r(M) & r(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} c_{M,0} \\ c_{M,1} \\ c_{M,2} \\ . \\ . \\ c_{M,M} \end{bmatrix} =$$

$$\begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r(1) & r(0) & \dots & r(M-1) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ r(M) & r(M-1) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,M} \\ a_{M,M-1} \\ a_{M,M-2} \\ . \\ . \\ a_{M,0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ P_M \end{bmatrix}$$

The augmented Wiener-Hopf equations for forward and backward prediction are very similar and they help in finding recursive in order recursions, as shown next.

Levinson – Durbin algorithm

- The augmented Wiener-Hopf equations for forward and backward prediction provide a short derivation of Levinson-Durbin main recursions.
- Stressing the order of the filter: all variables will receive a subscript expressing the order of the predictor: $R_m, \underline{r}_m, \underline{a}_m, \underline{w_o}_m$.
- Several order recursive equations can be written for the involved quantities:

$$\underline{r}_{m+1} = \begin{bmatrix} r(1) & r(2) & \dots & r(m) & r(m+1) \end{bmatrix}^T = \begin{bmatrix} \underline{r}_m \\ r(m+1) \end{bmatrix}$$

$$R_{m+1} = \begin{bmatrix} R_m & \underline{r}_m^B \\ (\underline{r}_m^B)^T & r(0) \end{bmatrix} = \begin{bmatrix} r(0) & \underline{r}_m^T \\ \underline{r}_m & R_m \end{bmatrix}$$

LD recursion derivation:

Let define the vectors $\underline{\psi}$ and the scalar Δ_{m-1} and evaluate them using augmented WH equations:

$$\underline{\psi} = \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}}{P_{m-1}} \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix} \tag{2}$$

$$\Delta_{m-1} = \underline{r}^T_m \underline{a}^B_{m-1} = \underline{a}^T_{m-1} \underline{r}^B_m = r(m) + \sum_{k=1}^{m-1} a_{m-1,k} r(m-k)$$

Multiplying the right hand side of Equation (2) by $R_{m+1} = \begin{bmatrix} R_m & \underline{r}^B_m \\ (\underline{r}^B_m)^T & r(0) \end{bmatrix} = \begin{bmatrix} r(0) & \underline{r}^T_m \\ \underline{r}_m & R_m \end{bmatrix}$ we obtain

$$R_{m+1}\underline{\psi} = R_{m+1} \left\{ \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} - \frac{\Delta_m}{P_{m-1}} \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix} \right\}$$

$$= \begin{bmatrix} R_m & \underline{r}^B_m \\ (\underline{r}^B_m)^T & r(0) \end{bmatrix} \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}}{P_{m-1}} \begin{bmatrix} r(0) & \underline{r}^T_m \\ \underline{r}_m & R_m \end{bmatrix} \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix}$$

$$= \begin{bmatrix} R_m \underline{a}_{m-1} \\ (\underline{r}^B_m)^T \underline{a}_{m-1} \end{bmatrix} - \frac{\Delta_{m-1}}{P_{m-1}} \begin{bmatrix} \underline{r}^T_m \underline{a}^B_{m-1} \\ R_m \underline{a}^B_{m-1} \end{bmatrix} = \begin{bmatrix} R_m \underline{a}_{m-1} \\ \Delta_{m-1} \end{bmatrix} - \frac{\Delta_{m-1}}{P_{m-1}} \begin{bmatrix} \Delta_{m-1} \\ R_m \underline{a}^B_{m-1} \end{bmatrix}$$

$$= \begin{bmatrix} P_{m-1} \\ \underline{0}_{m-1} \\ \Delta_{m-1} \end{bmatrix} - \frac{\Delta_{m-1}}{P_{m-1}} \begin{bmatrix} \Delta_{m-1} \\ \underline{0}_{m-1} \\ P_{m-1} \end{bmatrix} = \begin{bmatrix} P_{m-1} - \frac{\Delta^2_{m-1}}{P_{m-1}} \\ \underline{0}_{m-1} \\ \Delta_{m-1} - \Delta_{m-1} \end{bmatrix} = \begin{bmatrix} P_{m-1} - \frac{\Delta^2_{m-1}}{P_{m-1}} \\ \underline{0}_m \end{bmatrix}$$

LD recursion proof:

Recall the augmented WH equation

$$R_{M+1}\underline{a}_M = \begin{bmatrix} P_M \\ 0 \end{bmatrix}$$

Now using $\psi(1) = a_{m-1,0} = 1$, and since we suppose R_m nonsingular, the unique solution of

$$R_{m+1}\underline{\psi} = \begin{bmatrix} P_{m-1} - \frac{\Delta^2_{m-1}}{P_{m-1}} \\ 0_m \end{bmatrix}$$

provides the optimal predictor $\underline{a}_m = \underline{\psi}$ with the recursion (2) and the optimal prediction error power

$$P_m = P_{m-1} - \frac{\Delta^2_{m-1}}{P_{m-1}} = P_{m-1}(1 - \frac{\Delta^2_{m-1}}{P^2_{m-1}}) = P_{m-1}(1 - \Gamma^2_m)$$

with the notation

$$\Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}}$$

Levinson – Durbin recursions

Levinson – Durbin recursions

$$\underline{a}_m = \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix}$$ Vector form of L – D recursions

$$a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}, \qquad k = 0, 1, \ldots, m$$ Scalar form of L – D recursions

$$\Delta_{m-1} = \underline{a}^T_{m-1} \underline{r}^B_m = r(m) + \sum_{k=1}^{m-1} a_{m-1,k} r(m-k)$$

$$\Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}}$$

$$P_m = P_{m-1}(1 - \Gamma^2_m)$$

LD recursion variables:

Interpretation of Δ_m and Γ_m

1. $\Delta_{m-1} = E[f_{m-1}(n)b_{m-1}(n-1)]$
   Proof (Solution of Problem 9 page 238 in [Haykin91])

$$\begin{aligned} E[f_{m-1}(n)b_{m-1}(n-1)] &= E[\underline{a}_{m-1}^T \underline{u}(n)][\underline{u}(n-1)^T \underline{a}_{m-1}^B] = \underline{a}_{m-1}^T \begin{bmatrix} & \underline{r}_m^T \\ R_{m-1} & \underline{r}_{m-1}^B \end{bmatrix} \begin{bmatrix} -\underline{w}_{m-1}^B \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\underline{w}_{m-1}^T \end{bmatrix} \begin{bmatrix} \underline{r}_m^T \underline{a}_{m-1}^B \\ 0_{m-1} \end{bmatrix} = \underline{r}_m^T \underline{a}_{m-1}^B = \Delta_{m-1} \end{aligned}$$

2. $\Delta_0 = E[f_0(n)b_0(n-1)] = E[u(n)u(n-1)] = r(1)$
3. Iterating $P_m = P_{m-1}(1 - \Gamma_m^2)$ we obtain

$$P_m = P_0 \prod_{k=1}^{m} (1 - \Gamma_k^2)$$

4. Since the power of prediction error must be positive for all orders, the reflection coefficients are less than unit in absolute value:

$$|\Gamma_m| \leq 1 \quad \forall m = 0, \ldots, M$$

5. Reflection coefficients equal last autoregressive coefficient, for each order m:

$$\Gamma_m = a_{m,m}, \qquad \forall m = M, M-1, \ldots, 1$$

Algorithm Levinson-Durbin

Given $r(0), r(1), r(2), \ldots, r(M)$
(for example, estimated from data $u(1), u(2), u(3), \ldots, u(T)$ using $r(k) = \frac{1}{T} \sum_{n=k+1}^{T} u(n)u(n-k)$)

1. Initialize $\Delta_0 = r(1), \quad P_0 = r(0)$
2. For $m = 1, \ldots, M$

   2.1 $\Gamma_m = -\dfrac{\Delta_{m-1}}{P_{m-1}}$

   2.2 $a_{m,0} = 1$

   2.3 $a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}, \qquad k = 1, \ldots, m$

   2.4 $\Delta_m = r(m+1) + \sum_{k=1}^{m} a_{m,k} r(m+1-k)$

   2.5 $P_m = P_{m-1}(1 - \Gamma_m^2)$

Computational complexity:
For the m–th iteration of Step 2: $2m+2$ multiplications, $2m+2$ additions, 1 division
The overall computational complexity: $\mathcal{O}(M^2)$ operations

Algorithm (L–D) Second form

Given $r(0)$ and $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$

1. Initialize $P_0 = r(0)$
2. For $m = 1, \ldots, M$
   - 2.1 $a_{m,0} = 1$
   - 2.2 $a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}, \qquad k = 1, \ldots, m$
   - 2.3 $P_m = P_{m-1}(1 - \Gamma_m^2)$

Inverse Levinson – Durbin algorithm

$$a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}$$
$$a_{m,m-k} = a_{m-1,m-k} + \Gamma_m a_{m-1,k}$$

$$\begin{bmatrix} a_{m,k} \\ a_{m,m-k} \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} a_{m-1,m-k} \\ a_{m-1,k} \end{bmatrix}$$

and using the identity $\Gamma_m = a_{m,m}$

$$\boxed{a_{m-1,k} = \frac{a_{m,k} - a_{m,m} a_{m,m-k}}{1 - (a_{m,m})^2} \qquad k = 1, \ldots, m}$$

Inverse Levinson – Durbin algorithm

The second order properties of the AR process are perfectly described by the set of reflection coefficients

This immediately follows from the following property:
The sets $\{P_0, \Gamma_1, \Gamma_2, \ldots, \Gamma_M\}$ and $\{r(0), r(1), \ldots, r(M)\}$ are in one-to-one correspondence
Proof

(a) $\{r(0), r(1), \ldots, r(M)\} \overset{\text{(Algorithm L – D)}}{\Longrightarrow} \{P_0, \Gamma_1, \Gamma_2, \ldots, \Gamma_M, \}$
(b) From

$$\Gamma_{m+1} = -\frac{\Delta_m}{P_m} = -\frac{r(m+1) + \sum_{k=1}^{m} a_{m,k} r(m+1-k)}{P_m}$$

we can obtain immediately

$$r_{m+1} = -\Gamma_{m+1} P_m - \sum_{k=1}^{m} a_{m,k} r(m+1-k)$$

which can be iterated together with Algorithm L–D form 2, to obtain all $r(1), \ldots, r(M)$.

Whitening property of prediction – error filters

- In theory, a prediction – error filter is capable of whitening a stationary discrete-time stochastic process applied to its input, if the order of the filter is high enough.
- Then all information in the original stochastic process $u(n)$ is represented by the parameters $\{P_M, a_{M,1}, a_{M,2}, \ldots, a_{M,M}\}$ (or, equivalently, by $\{P_0, \Gamma_1, \Gamma_2, \ldots, \Gamma_M\}$).
- A signal equivalent (as second order properties) can be generated starting from $\{P_M, a_{M,1}, a_{M,2}, \ldots, a_{M,M}\}$ using the autoregressive difference equation model.
- These "analyze and generate" paradigms combine to provide the basic principle of vocoders.

Lattice Predictors

- **Order -Update Recursions for Prediction errors**
  Since the predictors obey the recursive–in–order equations

$$\underline{a}_m = \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix}$$

$$\underline{a}^B_m = \begin{bmatrix} 0 \\ \underline{a}^B_{m-1} \end{bmatrix} + \Gamma_m \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix}$$

it is natural that prediction errors can be expressed in recursive–in–order forms. These forms results considering the recursions for the vector $\underline{u}_{m+1}(n)$

$$\underline{u}_{m+1}(n) = \begin{bmatrix} \underline{u}_m(n) \\ u(n-m) \end{bmatrix}$$

$$\underline{u}_{m+1}(n) = \begin{bmatrix} u(n) \\ \underline{u}_m(n-1) \end{bmatrix}$$

Combining the equations we obtain

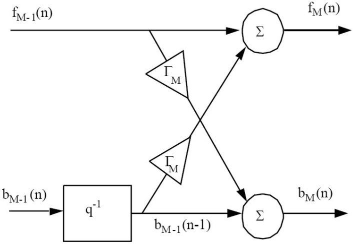
$$\begin{aligned} f_m(n) &= \underline{a}_m^T \underline{u}_{m+1}(n) = \begin{bmatrix} \underline{a}^T_{m-1} & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_m(n) \\ u(n-m) \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 & (\underline{a}^B_{m-1})^T \end{bmatrix} \begin{bmatrix} u(n) \\ \underline{u}_m(n-1) \end{bmatrix} = \\ &= \underline{a}^T_{m-1} \underline{u}_m(n) + \Gamma_m (\underline{a}^B_{m-1})^T \underline{u}_m(n-1) = \\ &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \end{aligned}$$

Lattice Predictors

$$\begin{aligned}
b_m(n) &= (\underline{a}_m^B)^T \underline{u}_{m+1}(n) = \begin{bmatrix} 0 & (\underline{a}_{m-1}^B)^T \end{bmatrix} \begin{bmatrix} u(n) \\ \underline{u}_m(n-1) \end{bmatrix} + \Gamma_m \begin{bmatrix} (\underline{a}_{m-1})^T & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_m(n) \\ u(n-m) \end{bmatrix} = \\
&= (\underline{a}_{m-1}^B)^T \underline{u}_m(n-1) + \Gamma_m (\underline{a}_{m-1})^T \underline{u}_m(n) \\
&= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n)
\end{aligned}$$

The order recursions of the errors can be represented as

$$\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n-1) \end{bmatrix}$$



Lattice Predictors

$$
\begin{aligned}
f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\
b_m(n) &= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n)
\end{aligned}
$$

Using the time shifting operator q^{-1}, the prediction error recursions are given by

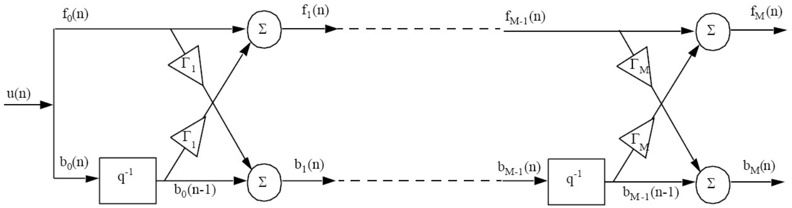
$$
\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m q^{-1} \\ \Gamma_m & q^{-1} \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n) \end{bmatrix}
$$

which can now be iterated for $m = 1, 2, \ldots, M$ to obtain

$$
\begin{aligned}
\begin{bmatrix} f_M(n) \\ b_M(n) \end{bmatrix} &= \begin{bmatrix} 1 & \Gamma_M q^{-1} \\ \Gamma_M & q^{-1} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{M-1} q^{-1} \\ \Gamma_{M-1} & q^{-1} \end{bmatrix} \cdots \begin{bmatrix} 1 & \Gamma_1 q^{-1} \\ \Gamma_1 & q^{-1} \end{bmatrix} \begin{bmatrix} f_0(n) \\ b_0(n) \end{bmatrix} \\
&= \begin{bmatrix} 1 & \Gamma_M q^{-1} \\ \Gamma_M & q^{-1} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{M-1} q^{-1} \\ \Gamma_{M-1} & q^{-1} \end{bmatrix} \cdots \begin{bmatrix} 1 & \Gamma_1 q^{-1} \\ \Gamma_1 & q^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(n)
\end{aligned}
$$

Having available the reflexion coefficients, all prediction errors of order $m = 1, \ldots, M$ can be computed using the Lattice predictor, in $2M$ additions and $2M$ multiplications.

Lattice Predictors



LATTICE PREDICTOR OF ORDER M

Some characteristics of the Lattice predictor:

1. It is the most efficient structure for generating simultaneously the forward and backward prediction errors.
2. The lattice structure is modular: increasing the order of the filter requires adding only one extra module, leaving all other modules the same.
3. The various stages of a lattice are decoupled from each other in the following sense: The memory of the lattice (storing $b_0(n-1), \ldots, b_{M-1}(n-1)$) contains orthogonal variables, thus the information contained in $u(n)$ is split in M pieces, which reduces gradually the redundancy of the signal.
4. The similar structure of the lattice filter stages makes the filter suitable for VLSI implementation.

Lattice Inverse filters

- The basic equations for one stage of the lattice are

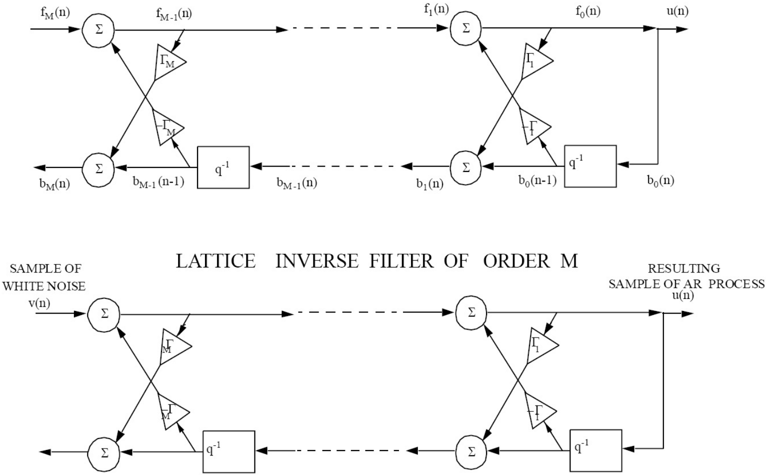
$$\begin{aligned} f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= \Gamma_m f_{m-1}(n) + b_{m-1}(n-1) \end{aligned}$$

and simply rewriting the first equation

$$\begin{aligned} f_{m-1}(n) &= f_m(n) - \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= \Gamma_m f_{m-1}(n) + b_{m-1}(n-1) \end{aligned}$$

we obtain the basic stage of the Lattice inverse filter representation.

Lattice Inverse filter and its use as a Synthesis filter



Burg estimation algorithm (not for exam)

- The optimum design of the lattice filter is a decoupled problem.
  At stage m the optimality criterion is:

$$J_m = E[f_m^2(n)] + E[b_m^2(n)]$$

and using the stage m equations

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n) \end{aligned}$$

$$\begin{aligned} J_m &= E[f_m^2(n)] + E[b_m^2(n)] = E[(f_{m-1}(n) + \Gamma_m b_{m-1}(n-1))^2] + E[(b_{m-1}(n-1) + \Gamma_m f_{m-1}(n))^2] \\ &= E[(f_{m-1}^2(n) + b_{m-1}^2(n-1)](1 + \Gamma_m^2) + 4\Gamma_m E[b_{m-1}(n-1) f_{m-1}(n)] \end{aligned}$$

Taking now the derivative with respect to Γ_m of the above criterion we obtain

$$\frac{d(J_m)}{d\Gamma_m} = 2E[(f_{m-1}^2(n) + b_{m-1}^2(n-1)]\Gamma_m + 4E[b_{m-1}(n-1) f_{m-1}(n)] = 0$$

and therefore

$$\Gamma_m^* = -\frac{2E[b_{m-1}(n-1) f_{m-1}(n)]}{E[(f_{m-1}^2(n)] + E[b_{m-1}^2(n-1)]}$$

Burg estimation algorithm (not for exam)

- Replacing the expectation operator E with time average operator $\frac{1}{N}\sum_{n=1}^{N}$ we obtain one direct way to estimate the parameters of the lattice filter, starting from the data available in lattice filter:

$$\Gamma_m = -\frac{2\sum_{n=1}^{N} b_{m-1}(n-1)f_{m-1}(n)}{\sum_{n=1}^{N}[(f_{m-1}^2(n) + b_{m-1}^2(n-1)]}$$

The parameters $\Gamma_1, \dots, \Gamma_M$ can be found solving first for Γ_1, then using Γ_1 to filter the data $u(n)$ and obtain $f_1(n)$ and $b_1(n)$, then find the estimate of Γ_2 ....
There are other possible estimators, but Burg estimator ensures the condition $|\Gamma| < 1$ which is required for the stability of the lattice filter.

Gradient Adaptive Lattice Filters (not for exam)

- Imposing the same optimality criterion as in Burg method

$$J_m = E[f_m^2(n)] + E[b_m^2(n)]$$

the gradient method applied to the lattice filter parameter at stage m is

$$\frac{d(J_m)}{d\Gamma_m} = 2E[f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n)]$$

and can be approximated (as usually in LMS algorithms) by

$$\hat{\nabla} J_m \approx 2[f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n)]$$

We obtain the updating equation for the parameter Γ_m

$$\Gamma_m(n+1) = \Gamma_m(n) - \frac{1}{2}\mu_m(n)\hat{\nabla} J_m = \Gamma_m(n) - \mu_m(n)(f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n))$$

In order to normalize the adaptation step, the following value of $\mu_m(n)$ was suggested

$$\mu_m(n) = \frac{1}{\xi_{m-1}(n)}$$

where

$$\xi_{m-1}(N) = \sum_{i=1}^{N}[(f_{m-1}^2(i) + b_{m-1}^2(i-1)] = \xi_{m-1}(N-1) + f_{m-1}^2(N) + b_{m-1}^2(N-1)$$

represents the total energy of forward and backward prediction errors.

Gradient Adaptive Lattice Filters

- We can introduce a forgetting factor using

$$\xi_{m-1}(n) = \beta \xi_{m-1}(n-1) + (1-\beta)[f_{m-1}^2(n) + b_{m-1}^2(n-1)]$$

with the forgetting factor close to 1, but $0 < \beta < 1$ allowing to forget the old history, which may be irrelevant if the filtered signal is nonstationary.