Linear Prediction

Overview

- 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:

$$
\begin{aligned}
r(k) &= E[u(n)u(n+k)] && \text{– autocorrelation function} \\
R &= E[\underline{u}(n-1)\underline{u}^T(n-1)] = E\begin{bmatrix} u(n-1) \\ u(n-2) \\ u(n-3) \\ u(n-M) \end{bmatrix} \begin{bmatrix} u(n-1) & u(n-2) & u(n-3) & \ldots & u(n-M) \end{bmatrix} \\
&= \begin{bmatrix} Eu(n-1)u(n-1) & Eu(n-1)u(n-2) & \ldots & Eu(n-1)u(n-M) \\ Eu(n-2)u(n-1) & Eu(n-2)u(n-2) & \ldots & Eu(n-2)u(n-M) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ Eu(n-M)u(n-1) & Eu(n-M)u(n-2) & \ldots & Eu(n-M)u(n-M) \end{bmatrix} = \\
R &= \begin{bmatrix} r(0) & r(1) & \ldots & r(M-1) \\ r(1) & r(0) & \ldots & r(M-2) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ r(M-1) & r(M-2) & \ldots & r(0) \end{bmatrix} && \text{– autocorrelation matrix} \\
\underline{r} &= E[\underline{u}(n-1)u(n)] = \begin{bmatrix} r(1) & r(2) & r(3) & \ldots & r(M) \end{bmatrix}^T && \text{– autocorrelation vector} \\
\underline{r}^B &= E[\underline{u}(n-1)u(n-M-1)] = \begin{bmatrix} r(M) & r(M-1) & r(M-2) & \ldots & r(1) \end{bmatrix}^T
\end{aligned}
$$

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

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

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:

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

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

- Two derivations of optimal solution

1. Transforming the criterion into a quadratic form

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

The matrix R is semi-positive 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

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

2. Derivation based on optimal Wiener filter design

   The optimal predictor evaluation can be rephrazed 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

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

  which can be written in a block matrix equation form

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

  But

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

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

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

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

  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 solution:

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

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

- 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

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

- Relations between Backward and Forward predictors

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

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:

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

The Forward and Backward optimal predictors are solutions of the systems

$$\begin{aligned}
R\underline{w}_o &= \underline{r} \\
R\underline{g}_o &= \underline{r}^B
\end{aligned}$$

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

and since R is supposed nonsingular, we have

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

- Augmented Wiener Hopf equations for Backward prediction 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

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

But

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

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

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

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

or

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

Levinson – Durbin algorithm

- Stressing the order of the filter

  All variables will receive a subscript expressing the order of the predictor: $R_m, \underline{r}_m, \underline{a}_m, \underline{w_o}_m$.

  Some order recursive equations can be written:

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

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

- Main recursions

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

where

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

Multipling 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

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

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

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

But since $\psi(1) = a_{m-1,0} = 1$, and 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

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

with the notation

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

Levinson – Durbin recursions

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

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

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

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

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

- 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 (L–D)

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 $\Gamma_m = a_{m,m}$

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

- 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 $\{P_M, a_{M,1}, a_{M,2}, \ldots, a_{M,}$ (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.

- Gram-Schmidt orthogonalization algorithm

$$
\begin{aligned}
b_0(n) &= u(n) \\
b_1(n) &= a_{1,1}u(n) + a_{1,0}u(n-1) \\
b_2(n) &= a_{2,2}u(n) + a_{2,1}u(n-1) + a_{2,0}u(n-2) \\
&. \\
&. \\
&. \\
b_M(n) &= a_{M,M}u(n) + a_{M,M-1}u(n-1) + \ldots + a_{M,0}u(n-M)
\end{aligned}
$$

Using the notations

$$
\begin{aligned}
\underline{u}(n) &= [u(n), u(n-1), \ldots, u(n-M)]^T \\
\underline{b}(n) &= [b_0(n), b_1(n), \ldots, b_M(n)]^T \\
L &= \begin{bmatrix} 1 & 0 & \ldots & 0 \\ a_{1,1} & 1 & \ldots & 0 \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ a_{M,M} & a_{M,M-1} & \ldots & 1 \end{bmatrix}
\end{aligned}
$$

we may write G–S orthogonalization procedure as

$$\underline{b}(n) = L\underline{u}(n)$$

- Orthogonality of Backward prediction errors

  The following orthogonality property holds:

$$E[b_i(n)b_j(n)] = \begin{cases} P_m, & i = j \\ 0, & i \neq j \end{cases}$$

Proof Suppose $j \geq i$

$$E[b_j(n) \times \ldots] \qquad\qquad b_i(n) = \sum_{k=0}^{i} a_{i,i-k} u(n-k)$$

$$Eb_j(n)b_i(n) = Eb_j(n) \sum_{k=0}^{i} a_{i,i-k} u(n-k) =$$

$$= \sum_{k=0}^{i} a_{i,i-k} Eb_j(n)u(n-k) = \begin{cases} Eb_i^2(n), & i = j \\ 0, & i \neq j \end{cases} = \begin{cases} P_m, & i = j \\ 0, & i \neq j \end{cases}$$

due to orthogonality of optimal Wiener filter error to the inputs involved in the computation of that error: $Eb_j(n)u(n-k) = 0$ for $k \leq j$.

In matrix form

$$E[\underline{b}(n)\underline{b}(n)^T] = \begin{bmatrix} P_0 & 0 & \ldots & 0 \\ 0 & P_1 & \ldots & 0 \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ 0 & 0 & \ldots & P_M \end{bmatrix} = D$$

Substituting $\underline{b}(n) = L\underline{u}(n)$

$$E[\underline{b}(n)\underline{b}(n)^T] = LE[\underline{u}(n)\underline{u}(n)^T]L^T = LRL^T = D$$

which can be use to factorize the matrices R and R^{-1} as

$$\begin{aligned} R &= L^{-1}DL^{-T} \\ R^{-1} &= (L^{-1}DL^{-T})^{-1} = L^TD^{-1}L = L^TD^{-1/2}D^{-1/2}L = (D^{-1/2}L)^TD^{-1/2}L \end{aligned} \tag{3}$$

Equation (3) provides the Cholesky factorization factorization of R^{-1}.