Parametric Signal Modeling and Linear Prediction Theory 5. Lattice Predictor

Electrical & Computer Engineering
University of Maryland, College Park

Acknowledgment: ENEE630 slides were based on class notes developed by Profs. K.J. Ray Liu and Min Wu. The LaTeX slides were made by Prof. Min Wu and Mr. Wei-Hong Chuang.

Contact: minwu@umd.edu. Updated: November 19, 2012.

Introduction

Recall: a forward or backward prediction-error filter can each be realized using a separate tapped-delay-line structure.

Lattice structure discussed in this section provides a powerful way to combine the FLP and BLP operations into a **single** structure.

Order Update for Prediction Errors

(Readings: Haykin §3.8)

Review:

1. signal vector $\underline{u}_{m+1}[n] = \begin{bmatrix} \underline{u}_m[n] \\ u[n-m] \end{bmatrix} = \begin{bmatrix} u[n] \\ \underline{u}_m[n-1] \end{bmatrix}$

2. Levinson-Durbin recursion:

$$\underline{a}_m = \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ \underline{a}^{B*}_{m-1} \end{bmatrix} \text{ (forward)}$$

$$\underline{a}^{B*}_m = \begin{bmatrix} 0 \\ \underline{a}^{B*}_{m-1} \end{bmatrix} + \Gamma^*_m \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} \text{ (backward)}$$

Recursive Relations for $f_m[n]$ and $b_m[n]$

$$f_m[n] = \underline{a}_m^H \underline{u}_{m+1}[n]; \; b_m[n] = \underline{a}_m^{BT} \underline{u}_{m+1}[n]$$

1. FLP:

$$f_m[n] = \left[\underline{a}_{m-1}^H \,\vdots\, 0 \right] \begin{bmatrix} \underline{u}_m[n] \\ u[n-m] \end{bmatrix} + \Gamma_m^* \left[0 \,\vdots\, \underline{a}_{m-1}^{BT} \right] \begin{bmatrix} u[n] \\ \underline{u}_m[n-1] \end{bmatrix}$$

(Details)

$$f_m[n] = f_{m-1}[n] + \Gamma_m^* b_{m-1}[n-1]$$

2. BLP:

$$b_m[n] = \left[0 \,\vdots\, \underline{a}_{m-1}^{BT} \right] \begin{bmatrix} u[n] \\ \underline{u}_m[n-1] \end{bmatrix} + \Gamma_m \left[\underline{a}_{m-1}^* \,\vdots\, 0 \right] \begin{bmatrix} \underline{u}_m[n] \\ u[n-m] \end{bmatrix}$$

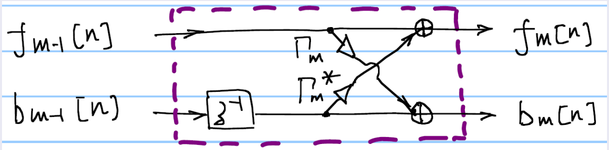
(Details)

$$b_m[n] = b_{m-1}[n-1] + \Gamma_m f_{m-1}[n]$$

Basic Lattice Structure

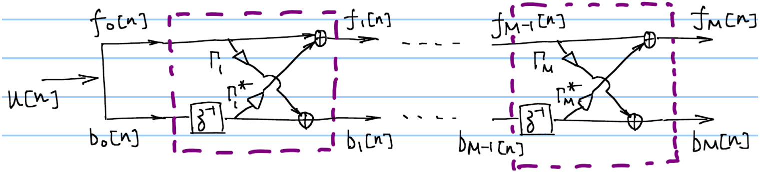
$$\begin{bmatrix} f_m[n] \\ b_m[n] \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m^* \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}[n] \\ b_{m-1}[n-1] \end{bmatrix}, \quad m = 1, 2, \ldots, M$$

Signal Flow Graph (SFG)



Modular Structure

Recall $f_0[n] = b_0[n] = u[n]$, thus



To increase the order, we simply add more stages and reuse the earlier computations.

Using a tapped delay line implementation, we need M separate filters to generate $b_1[n], b_2[n], \ldots, b_M[n]$.

In contrast, a single lattice structure can generate $b_1[n], \ldots, b_M[n]$ as well as $f_1[n], \ldots, f_M[n]$ at the same time.

Correlation Properties

	Given from a zero-mean w.s.s. process:		Predict
(FLP)	$\{u[n-1], \ldots, u[n-M]\}$	$\Rightarrow$	$u[n]$
(BLP)	$\{u[n], u[n-1], \ldots, u[n-M+1]\}$	$\Rightarrow$	$u[n-M]$

1. Principle of Orthogonality

	i.e., conceptually
$\mathbb{E}\left[f_m[n]u^*[n-k]\right] = 0 \; (1 \le k \le m)$	$f_m[n] \perp \underline{u}_m[n-1]$
$\mathbb{E}\left[b_m[n]u^*[n-k]\right] = 0 \; (0 \le k \le m-1)$	$b_m[n] \perp \underline{u}_m[n]$

2. $\mathbb{E}\left[f_m[n]u^*[n]\right] = \mathbb{E}\left[b_m[n]u^*[n-m]\right] = P_m$

<u>Proof</u> : (Details)

Correlation Properties

3. Correlations of error signals across orders:

(BLP) $$\mathbb{E}\left[b_m[n]b_i^*[n]\right] = \begin{cases} P_m & i = m \\ 0 & i < m \text{ i.e., } b_m[n] \perp b_i[n] \end{cases}$$

(FLP) $$\mathbb{E}\left[f_m[n]f_i^*[n]\right] = P_m \text{ for } i \leq m$$

Proof : (Details) (can obtain the case $i > m$ by conjugation)

Remark : The generation of $\{b_0[n], b_1[n], \ldots,\}$ is like a **Gram-Schmidt** orthogonalization process on $\{u[n], u[n-1], \ldots,\}$.

As a result, $\{b_i[n]\}_{i=0,1,\ldots}$ is a new, **uncorrelated** representation of $\{u[n]\}$ containing exactly the **same information**.

Correlation Properties

4. Correlations of error signals across orders and time:

$$\mathbb{E}\left[f_m[n]f_i^*[n-\ell]\right] = \mathbb{E}\left[f_m[n+\ell]f_i^*[n]\right] = 0 \ (1 \leq \ell \leq m-i, i < m)$$

$$\mathbb{E}\left[b_m[n]b_i^*[n-\ell]\right] = \mathbb{E}\left[b_m[n+\ell]b_i^*[n]\right] = 0 \ (0 \leq \ell \leq m-i-1, i < m)$$

Proof : (Details)

5. Correlations of error signals across orders and time:

$$\mathbb{E}\left[f_m[n+m]f_i^*[n+i]\right] = \begin{cases} P_m & i = m \\ 0 & i < m \end{cases}$$

$$\mathbb{E}\left[b_m[n+m]b_i^*[n+i]\right] = P_m \qquad i \leq m$$

Proof : (Details)

Correlation Properties

6. Cross-correlations of FLP and BLP error signals:

$$\mathbb{E}\left[f_m[n]b_i^*[n]\right] = \begin{cases} \Gamma_i^* P_m & i \le m \\ 0 & i > m \end{cases}$$

Proof : (Details)

Joint Process Estimator: Motivation

(Readings: Haykin §3.10; Hayes §7.2.4, §9.2.8)

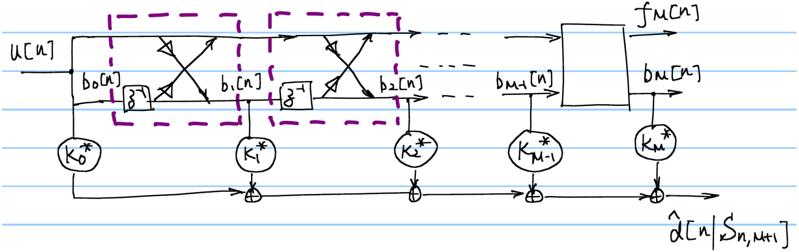
In (general) Wiener filtering theory, we use $\{x[n]\}$ process to estimate a desired response $\{d[n]\}$.

Solving the normal equation may require inverting the correlation matrix $\mathbf{R}_x$.

We now use the lattice structure to obtain a backward prediction error process $\{b_i[n]\}$ as an equivalent, uncorrelated representation of $\{u[n]\}$ that contains exactly the same information.

We then apply an optimal filter on $\{b_i[n]\}$ to estimate $\{d[n]\}$.

Joint Process Estimator: Structure



$$\hat{d}\,[n|\mathbb{S}_n] = \underline{k}^H \underline{b}_{M+1}[n], \text{ where } \underline{k} = [k_0, k_1, \ldots, k_M]^T$$

Joint Process Estimator: Result

To determine the optimal weight to minimize MSE of estimation:

1. Denote D as the $(M+1) \times (M+1)$ correlation matrix of $\underline{b}[n]$

   $$D = \mathbb{E}\left[\underline{b}[n]\underline{b}^H[n]\right] = \underset{\sim}{\text{diag}}(P_0, P_1, \ldots, P_M)$$

   $\because \{b_k[n]\}_{k=0}^M$ are uncorrelated

2. Let $\underline{s}$ be the crosscorrelation vector

   $$\underline{s} \triangleq [s_0, \ldots, s_M \ldots]^T = \mathbb{E}\left[\underline{b}[n]d^*[n]\right]$$

3. The normal equation for the optimal weight vector is:

   $$D\underline{k}_{\text{opt}} = \underline{s}$$

   $$\Rightarrow \underline{k}_{\text{opt}} = D^{-1}\underline{s} = \text{diag}(P_0^{-1}, P_1^{-1}, \ldots, P_M^{-1})\underline{s}$$

   i.e., $k_i = P_i^{-1}s_i, \; i = 0, \ldots, M$

Joint Process Estimator: Summary

The name "joint process estimation" refers to the system's structure that performs two optimal estimation jointly:

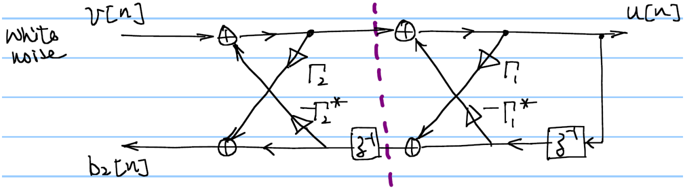
- One is a **lattice predictor** (characterized by $\Gamma_1, \ldots, \Gamma_M$) transforming a sequence of correlated samples $u[n]$, $u[n-1], \ldots, u[n-M]$ into a sequence of uncorrelated samples $b_0[n], b_1[n], \ldots, b_M[n]$.
- The other is called a **multiple regression filter** (characterized by $\underline{k}$), which uses $b_0[n], \ldots, b_M[n]$ to produce an estimate of $d[n]$.

Inverse Filtering

The lattice predictor discussed just now can be viewed as an analyzer, i.e., to represent an (approximately) AR process $\{u[n]\}$ using $\{\Gamma_m\}$.

With some reconfiguration, we can obtain an inverse filter or a synthesizer, i.e., we can reproduce an AR process by applying white noise $\{v[n]\}$ as the input to the filter.

A 2-stage Inverse Filtering



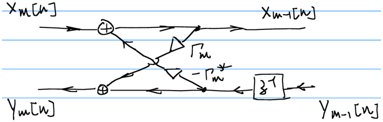
$$u[n] = v[n] - \Gamma_1^* u[n-1] - \Gamma_2^*(\Gamma_1 u[n-1] + u[n-2])$$

$$= v[n] - \underbrace{(\Gamma_1^* + \Gamma_1 \Gamma_2^*)}_{a_{2,1}^*} u[n-1] - \underbrace{\Gamma_2^*}_{a_{2,2}^*} u[n-2]$$

$$\therefore u[n] + a_{2,1}^* u[n-1] + a_{2,2}^* u[n-2] = v[n]$$

$\Rightarrow \{u[n]\}$ is an **AR(2)** process.

Basic Building Block for All-pole Filtering



$$\begin{cases} x_{m-1}[n] = x_m[n] - \Gamma_m^* y_{m-1}[n-1] \\ y_m[n] = \Gamma_m x_{m-1}[n] + y_{m-1}[n-1] \\ \qquad\quad = \Gamma_m x_m[n] + (1 - |\Gamma_m|^2) y_{m-1}[n-1] \end{cases}$$

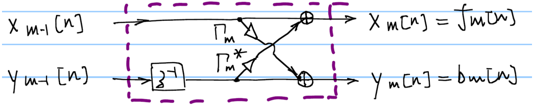
$$\Rightarrow \begin{cases} x_m[n] = x_{m-1}[n] + \Gamma_m^* y_{m-1}[n-1] \\ y_m[n] = \Gamma_m x_{m-1}[n] + y_{m-1}[n-1] \end{cases}$$

$$\therefore \begin{bmatrix} x_m[n] \\ y_m[n] \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m^* \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} x_{m-1}[n] \\ y_{m-1}[n-1] \end{bmatrix}$$

All-pole Filter via Inverse Filtering

$$\begin{bmatrix} x_m[n] \\ y_m[n] \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m^* \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} x_{m-1}[n] \\ y_{m-1}[n-1] \end{bmatrix}$$

This gives basically the same relation as the forward lattice module:



$\Rightarrow u[n] = -a_{2,1}^* u[n-1] - a_{2,2}^* u[n-2] + v[n]$ $\quad v[n]$: white noise