Statistical Digital Signal Processing

Algorithms and Structures for Optimum Linear Filters

Bisrat Derebssa, SECE, AAiT, AAU

Introduction

- The design and application of optimum filters involves
  - Determination of the optimum set of coefficients,
  - Evaluation of the cost function to determine whether the obtained parameters satisfy the design requirements, and
  - The implementation of the optimum filter.

- There are several important reasons to study the normal equations in greater detail in order to develop efficient, special-purpose algorithms for their solution.
  - The throughput of several real-time applications can only be served with algorithms that are obtained by exploiting the special structure of the correlation matrix.
  - We can develop order-recursive algorithms that help us to choose the correct filter order or to stop before numerical problems.
  - Some algorithms lead to intermediate sets of parameters that have physical meaning, provide easy tests, or are useful in special applications.
  - Sometimes there is a link between the algorithm for the solution of the normal equations and the structure for implementation.

Order-recursive algorithms

- Fixed-order algorithms
  - To solve the normal equations the order of the estimator should be known.
- When the order of the estimator becomes a design variable, fixed-order algorithms are not effective.
  - If order changes, the optimum coefficients have to be calculated again from scratch.

$$e_m(n) \triangleq y(n) - \hat{y}_m(n)$$

$$\hat{y}_m(n) \triangleq \mathbf{c}_m^H(n)\mathbf{x}_m(n)$$

$$\mathbf{c}_m(n) \triangleq [c_1^{(m)}(n)\ c_2^{(m)}(n)\ \cdots\ c_m^{(m)}(n)]^T$$

$$\mathbf{x}_m(n) \triangleq [x_1(n)\ x_2(n)\ \cdots\ x_m(n)]^T$$

- We would like to arrange the computations so that the results for order m, that is, $c_m(n)$ or $\hat{y}_m(n)$, can be used to compute the estimates for order m + 1, that is, $c_{m+1}(n)$ or $\hat{y}_{m+1}(n)$.
  - The resulting procedures are called order-recursive algorithms or order-updating relations.
- Similarly, procedures that compute $c_m(n + 1)$ from $c_m(n)$ or $\hat{y}_m(n+1)$ from $\hat{y}_m(n)$ are called time-recursive algorithms or time-updating relations.

Matrix Partitioning and Optimum Nesting

- If the order of the estimator increases from m to m+1, then the input data vector is augmented with one additional observation x_{m+1}.

$\mathbf{x}_{m+1}^{\lceil m \rceil}$ First m components of x_{m+1}

$\mathbf{x}_{m+1}^{\lfloor m \rfloor}$ Last m components of x_{m+1}

$\mathbf{R}_{m+1}^{\lceil m \rceil}$ The first mxm sub matrix of R_{m+1}

$\mathbf{R}_{m+1}^{\lfloor m \rfloor}$ The last mxm sub matrix of R_{m+1}

$\mathbf{R}_4^{\lceil 3 \rceil}$

$$\mathbf{R}_4 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}$$

$\mathbf{R}_4^{\lfloor 3 \rfloor}$

- Since $\mathbf{x}_{m+1}^{\lceil m \rceil} = \mathbf{x}_m$

$$\mathbf{R}_{m+1} = E\left\{\begin{bmatrix}\mathbf{x}_m \\ x_{m+1}\end{bmatrix}\begin{bmatrix}\mathbf{x}_m^H & x_{m+1}^*\end{bmatrix}\right\} = \begin{bmatrix}\mathbf{R}_m & \mathbf{r}_m^{\mathrm{b}} \\ \mathbf{r}_m^{\mathrm{b}H} & \rho_m^{\mathrm{b}}\end{bmatrix}$$

$$\mathbf{r}_m^{\mathrm{b}} \triangleq E\{\mathbf{x}_m x_{m+1}^*\}$$

$$\rho_m^{\mathrm{b}} \triangleq E\{|x_{m+1}|^2\}$$

$$\mathbf{x}_{m+1}^{\lceil m \rceil} = \mathbf{x}_m \Rightarrow \mathbf{R}_m = \mathbf{R}_{m+1}^{\lceil m \rceil} \qquad \mathbf{d}_m = \mathbf{d}_{m+1}^{\lceil m \rceil}$$

- This is known as the optimum nesting property and is instrumental in the development of order recursive algorithms.

- The inverse of the m+1 autocorrelation matrix is given as the following.

$$\mathbf{R}_{m+1}^{-1} = \begin{bmatrix} \mathbf{R}_m & \mathbf{r}_m^{\mathrm{b}} \\ \mathbf{r}_m^{\mathrm{b}H} & \rho_m^{\mathrm{b}} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}_m^{-1} & \mathbf{0}_m \\ \mathbf{0}_m^H & 0 \end{bmatrix} + \frac{1}{\alpha_m^{\mathrm{b}}} \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H & 1 \end{bmatrix}$$

Where:

$$\mathbf{b}_m \triangleq [b_0^{(m)}\ b_1^{(m)}\ \cdots\ b_{m-1}^{(m)}]^T \triangleq -\mathbf{R}_m^{-1}\mathbf{r}_m^{\mathrm{b}}$$

$$\alpha_m^{\mathrm{b}} \triangleq \rho_m^{\mathrm{b}} - \mathbf{r}_m^{\mathrm{b}H}\mathbf{R}_m^{-1}\mathbf{r}_m^{\mathrm{b}} = \rho_m^{\mathrm{b}} + \mathbf{r}_m^{\mathrm{b}H}\mathbf{b}_m$$

Alternatively:

$$\alpha_m^{\mathrm{b}} = \frac{\det \mathbf{R}_{m+1}}{\det \mathbf{R}_m}$$

- Note that:
  - The inverse R_{m+1} of the *m+1* autocorrelation matrix is obtained directly from the inverse R_m.
  - The vector b_m is the MMSE estimator of observation x_{m+1} from data vector x_m.
  - The inverse matrix does not have the optimum nesting property.

- The inverse of the lower right corner partitioned matrix

$$\mathbf{R}_{m+1}^{-1} \triangleq \begin{bmatrix} \rho_m^{\text{f}} & \mathbf{r}_m^{\text{f}\,H} \\ \mathbf{r}_m^{\text{f}} & \mathbf{R}_m^{\text{f}} \end{bmatrix}^{-1} = \begin{bmatrix} 0 & \mathbf{0}_m^H \\ \mathbf{0}_m & (\mathbf{R}_m^{\text{f}})^{-1} \end{bmatrix} + \frac{1}{\alpha_m^{\text{f}}} \begin{bmatrix} 1 \\ \mathbf{a}_m \end{bmatrix} \begin{bmatrix} 1 & \mathbf{a}_m^H \end{bmatrix}$$

where

$$\mathbf{a}_m \triangleq [a_1^{(m)}\ a_2^{(m)}\ \cdots\ a_m^{(m)}]^T \triangleq -(\mathbf{R}_m^{\text{f}})^{-1}\mathbf{r}_m^{\text{f}}$$

$$\alpha_m^{\text{f}} \triangleq \rho_m^{\text{f}} - \mathbf{r}_m^{\text{f}\,H}(\mathbf{R}_m^{\text{f}})^{-1}\mathbf{r}_m^{\text{f}} = \rho_m^{\text{f}} + \mathbf{r}_m^{\text{f}\,H}\mathbf{a}_m = \frac{\det \mathbf{R}_{m+1}}{\det \mathbf{R}_m^{\text{f}}}$$

Levinson Recursion for the Optimum Estimator

- Solving the m+1 normal equation

$$\mathbf{c}_{m+1} = \mathbf{R}_{m+1}^{-1}\mathbf{d}_{m+1}$$

$$= \begin{bmatrix} \mathbf{R}_m^{-1} & \mathbf{0}_m \\ \mathbf{0}_m^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_m \\ d_{m+1} \end{bmatrix} + \frac{1}{\alpha_m^{\mathrm{b}}} \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H & 1 \end{bmatrix} \begin{bmatrix} \mathbf{d}_m \\ d_{m+1} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{R}_m^{-1}\mathbf{d}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} \frac{\mathbf{b}_m^H \mathbf{d}_m + d_{m+1}}{\alpha_m^{\mathrm{b}}}$$

Where

$$\mathbf{c}_{m+1} = \begin{bmatrix} \mathbf{c}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m^c$$

$$k_m^c \triangleq \frac{\beta_m^c}{\alpha_m^{\mathrm{b}}}$$

$$\beta_m^c \triangleq \mathbf{b}_m^H \mathbf{d}_m + d_{m+1}$$

$$\mathbf{c}_{m+1} = \begin{bmatrix} \mathbf{c}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m^c$$

- Note that
  - Even though the equation is order-recursive, the parameter c_{m+1} does not have the optimum nesting property.

$$\mathbf{c}_{m+1}^{\lceil m \rceil} \neq \mathbf{c}_m$$

  - If b_m is known, c_{m+1} can be calculated.
  - However, the calculation of b_m requires the inversion of R_m.
    - Minimal computational savings.

$$\mathbf{b}_m \triangleq [b_0^{(m)}\ b_1^{(m)}\ \cdots\ b_{m-1}^{(m)}]^T \triangleq -\mathbf{R}_m^{-1}\mathbf{r}_m^{\mathrm{b}}$$

Order-recursive computation of LDL^H Decomposition

- The m+1 autocorrelation matrix R can be written as

$$\mathbf{R}_{m+1} = \mathbf{L}_{m+1}\mathbf{D}_{m+1}\mathbf{L}_{m+1}^H$$

Where

$$\mathbf{L}_{m+1} = \begin{bmatrix} \mathbf{L}_m & \mathbf{0} \\ \mathbf{l}_m^H & 1 \end{bmatrix} \qquad \mathbf{D}_{m+1} = \begin{bmatrix} \mathbf{D}_m & \mathbf{0} \\ \mathbf{0}^H & \xi_{m+1} \end{bmatrix} \quad \mathbf{R}_m = \mathbf{L}_m\mathbf{D}_m\mathbf{L}_m^H$$

- Note that both matrices have optimum nesting property

$$\mathbf{L}_m = \mathbf{L}^{\lceil m \rceil}, \mathbf{D}_m = \mathbf{D}^{\lceil m \rceil}$$

- From LDL^H decomposition of linear MMSE

$$\mathbf{L}_m \mathbf{D}_m \mathbf{k}_m \triangleq \mathbf{d}_m$$

$$\mathbf{L}_m^H \mathbf{c}_m = \mathbf{k}_m$$

- Since L_m is lower triangular, k_m has the optimum nesting property

$$\mathbf{k}_m = \mathbf{k}^{\lceil m \rceil}$$

- However, since $L_m{}^H$ is not lower triangular, c_m does not satisfy the optimum nesting property.
- The MMSE also has the optimum nesting property

$$P_m = P_y - \mathbf{c}_m^H \mathbf{d}_m = P_y - \mathbf{k}_m^H \mathbf{D}_m \mathbf{k}_m$$

Order-Recursive Computation of the Optimum Estimate

- The computation of the optimum linear estimate using a linear combiner requires m multiplications and m-1 additions.
  - To compute the estimate for $1 \leq m \leq M$, we need $M(M + 1)/2$ operations.
- From LDL^H decomposition,

$$\hat{y}_m = \mathbf{c}_m^H \mathbf{x}_m = (\mathbf{k}_m^H \mathbf{L}_m^{-1}) \mathbf{x}_m = \mathbf{k}_m^H (\mathbf{L}_m^{-1} \mathbf{x}_m)$$

- Define a new vector w_m called innovation as

$$\mathbf{L}_m \mathbf{w}_m \triangleq \mathbf{x}_m$$

- Then the estimate is given as

$$\hat{y}_m = \mathbf{k}_m^H \mathbf{w}_m = \sum_{i=1}^{m} k_i^* w_i$$

- Since both k_m^H and w_m satisfy the optimum nesting property, the estimate also has optimum nesting property.
- Therefore,

$$\hat{y}_m = \hat{y}_{m-1} + k_m^* w_m$$

$$e_m = e_{m-1} - k_m^* w_m$$

$$w_m = x_m - \sum_{i=1}^{m-1} l_{i-1}^{(m-1)} w_i$$

- Note that:
  - The correlation of w_m is

$$E\{\mathbf{w}_m \mathbf{w}_m^H\} = \mathbf{L}_m^{-1} E\{\mathbf{x}_m \mathbf{x}_m^H\} \mathbf{L}_m^{-H} = \mathbf{D}_m$$

  - Therefore the components of w_m are uncorrelated.
  - The transformation from x_m to w_m removes all the redundant correlation among components of **x**.
  - Therefore each w_i adds new information or innovation.

- The estimation equation shows that the improvement in the estimate when an additional observation is included is proportional to the innovation w_{m+1} contained in x_{m+1}.
- Therefore, L_{m-1} acts as a decorrelator.
- k_m^H acts a linear combiner.
- LDL^H decomposition can be seen as the matrix equivalent of spectral factorization.

ORDER-RECURSIVE ALGORITHMS FOR OPTIMUM FIR FILTERS

- The key difference between a linear combiner and an FIR filter is the nature of the input data vector.
- The input data vector for FIR filters consists of consecutive samples from the same discrete-time stochastic process.
- Taking the shift invariance of the input data

$$\mathbf{x}_{m+1}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-m+1) \\ x(n-m) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_m(n) \\ x(n-m) \end{bmatrix} = \begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix}$$

- The correlation matrix $R_{m+1}(n)$ can be shown to be

$$\mathbf{R}_{m+1}(n) = E\{\mathbf{x}_{m+1}(n)\mathbf{x}_{m+1}^H(n)\}$$

$$\mathbf{R}_{m+1}(n) = \begin{bmatrix} \mathbf{R}_m(n) & \mathbf{r}_m^{\mathrm{b}}(n) \\ \mathbf{r}_m^{\mathrm{b}H}(n) & P_x(n-m) \end{bmatrix} \qquad \mathbf{R}_{m+1}(n) = \begin{bmatrix} P_x(n) & \mathbf{r}_m^{\mathrm{f}H}(n) \\ \mathbf{r}_m^{\mathrm{f}}(n) & \mathbf{R}_m(n-1) \end{bmatrix}$$

$$\mathbf{r}_m^{\mathrm{b}}(n) = E\{\mathbf{x}_m(n)x^*(n-m)\}$$

$$\mathbf{r}_m^{\mathrm{f}}(n) = E\{\mathbf{x}_m(n-1)x^*(n)\}$$

$$P_x(n) = E\{|x(n)|^2\}$$

- Note that

$$\mathbf{R}_{m+1}^{\lfloor m \rfloor}(n) = \mathbf{R}_m(n-1)$$

- If the optimum m FIR filter coefficients are known at time n, the m+1 time coefficients can be calculated as

$$\mathbf{c}_{m+1}(n) = \mathbf{R}_{m+1}^{-1}(n)\,\mathbf{d}_{m+1}(n)$$

$$\mathbf{R}_{m+1}^{-1}(n) = \begin{bmatrix} \mathbf{R}_m^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{P_m^{\mathrm{b}}(n)} \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H(n) & 1 \end{bmatrix}$$

$$\mathbf{b}_m(n) = -\mathbf{R}_m^{-1}(n)\mathbf{r}_m^{\mathrm{b}}(n)$$

$$\mathbf{d}_{m+1}(n) = E\left\{ \begin{bmatrix} \mathbf{x}_m(n) \\ x(n-m) \end{bmatrix} y^*(n) \right\} = \begin{bmatrix} \mathbf{d}_m(n) \\ d_{m+1}(n) \end{bmatrix}$$

- By substitution,

$$\mathbf{c}_{m+1}(n) = \begin{bmatrix} \mathbf{c}_m(n) \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} k_m^c(n)$$

$$k_m^c(n) \triangleq \frac{\beta_m^c(n)}{P_m^{\mathrm{b}}(n)}$$

$$\beta_m^c(n) \triangleq \mathbf{b}_m^H(n)\mathbf{d}_m(n) + d_{m+1}(n)$$

- This is called the Levinson order recursion.

- For this order recursion to be useful, we need an order recursion for the backward linear prediction (BLP) $b_m(n)$.
- This is possible if $b_m(n)$ has optimum nesting.

$$\mathbf{R}_m(n)\mathbf{b}_m(n) = -\mathbf{r}_m^{\mathrm{b}}(n)$$

$$\mathbf{R}_{m+1}(n)\mathbf{b}_{m+1}(n) = -\mathbf{r}_{m+1}^{\mathrm{b}}(n)$$

- The right side vectors are not nested if we use upper partitioning.

- If we use lower-upper partitioning

$$\mathbf{r}^{\mathrm{b}}_{m+1}(n) = E\left\{\begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix} x^*(n-m-1)\right\} \triangleq \begin{bmatrix} r^{\mathrm{b}}_{m+1}(n) \\ \mathbf{r}^{\mathrm{b}}_m(n-1) \end{bmatrix}$$

- By using lower-upper partitioning of R_{m+1}

$$\mathbf{R}^{-1}_{m+1}(n) = \begin{bmatrix} 0 & \mathbf{0}^H \\ \mathbf{0} & \mathbf{R}^{-1}_m(n-1) \end{bmatrix} + \frac{1}{P^{\mathrm{f}}(n)} \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} \begin{bmatrix} 1 & \mathbf{a}^H_m(n) \end{bmatrix}$$

$$\mathbf{a}_m(n) \triangleq -\mathbf{R}^{-1}_m(n-1)\mathbf{r}^{\mathrm{f}}_m(n)$$

Forward linear prediction

$$P^{\mathrm{f}}_m(n) = \frac{\det \mathbf{R}_{m+1}(n)}{\det \mathbf{R}_m(n-1)} = P_x(n) + \mathbf{r}^{\mathrm{f}\,H}_m(n)\mathbf{a}_m(n)$$

- By substitution

$$\mathbf{b}_{m+1}(n) = -\mathbf{R}_{m+1}^{-1}(n)\mathbf{r}_{m+1}^{\mathrm{b}}(n)$$

$$\mathbf{b}_{m+1}(n) = \begin{bmatrix} 0 \\ \mathbf{b}_m(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} k_m^{\mathrm{b}}(n)$$

$$k_m^{\mathrm{b}}(n) \triangleq -\frac{\beta_m^{\mathrm{b}}(n)}{P_m^{\mathrm{f}}(n)}$$

$$\beta_m^{\mathrm{b}}(n) \triangleq r_{m+1}^{\mathrm{b}}(n) + \mathbf{a}_m^H(n)\mathbf{r}_m^{\mathrm{b}}(n-1)$$

- Similarly $a_m(n)$ does not have optimum nesting.

- Order recursion for FLP

$$\mathbf{a}_{m+1}(n) = \begin{bmatrix} \mathbf{a}_m(n) \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m(n-1) \\ 1 \end{bmatrix} k_m^{\mathrm{f}}(n)$$

$$k_m^{\mathrm{f}}(n) \triangleq -\frac{\beta_m^{\mathrm{f}}(n)}{P_m^{\mathrm{b}}(n-1)}$$

$$\beta_m^{\mathrm{f}}(n) \triangleq \mathbf{b}_m^H(n-1)\mathbf{r}_m^{\mathrm{f}}(n) + r_{m+1}^{\mathrm{f}}(n)$$

- Clearly, a_m does not have the optimum nesting property.

Simplification for Stationary Stochastic Processes

- When x(n) and y(n) are jointly wide-sense stationary (WSS), the optimum estimators are time-invariant and we have the following simplifications:
  - All quantities are independent of
    - no time recursion necessary.
  - $b_m = J a_m^*$,   J simply reverses the order of the vector elements

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \qquad \mathbf{J}^H\mathbf{J} = \mathbf{J}\mathbf{J}^H = \mathbf{I}$$

- This is due to the Toeplitz structure of the autocorrelation matrix.

- Therefore, R_{m+1} can be partitioned as

$$\mathbf{R}_{m+1}(n) = \begin{bmatrix} \mathbf{R}_m & \mathbf{J}\mathbf{r}_m \\ \mathbf{r}_m^H\mathbf{J} & r(0) \end{bmatrix} = \begin{bmatrix} r(0) & \mathbf{r}_m^T \\ \mathbf{r}_m^* & \mathbf{R}_m \end{bmatrix}$$

$$\mathbf{r}_m \triangleq [r(1)\ r(2)\ \cdots\ r(m)]^T$$

- It can be shown that

$$\mathbf{a}_{m+1} = \begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m$$

$$\mathbf{b}_m = \mathbf{J}\mathbf{a}_m^*$$

- Where

$$k_m \triangleq k_m^{\mathrm{f}} = k_m^{\mathrm{b}*} = -\frac{\beta_m}{P_m}$$

$$\beta_m \triangleq \beta_m^{\mathrm{f}} = \beta_m^{\mathrm{b}*} = \mathbf{b}_m^H \mathbf{r}_m^* + r^*(m+1)$$

$$P_m \triangleq P_m^{\mathrm{b}} = P_m^{\mathrm{f}} = P_{m-1} + \beta_{m-1}^* k_{m-1} = P_{m-1} + \beta_{m-1} k_{m-1}^*$$

- The optimum coefficients are

$$\mathbf{c}_{m+1} = \begin{bmatrix} \mathbf{c}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{J}\mathbf{a}_m \\ 1 \end{bmatrix} k_m^c$$

$$k_m^c \triangleq \frac{\beta_m^c}{P_m}$$

$$\beta_m^c = \mathbf{b}_m^H \mathbf{d}_m + d_{m+1}$$

Levinson-Durbin Algorithm

- For stationary RP, the Toeplitz structure of the autocorrelation matrix can be used to come up with efficient order recursive algorithms.
- Suppose that c_m is known

$$\mathbf{c}_m = \mathbf{R}_m^{-1}\mathbf{d}_m$$

and we wish to determine

$$\mathbf{c}_{m+1} = \mathbf{R}_{m+1}^{-1}\mathbf{d}_{m+1}$$

- Since R_{m+1} and d_{m+1} can be partitioned as follows

$$\mathbf{R}_{m+1} = \left[\begin{array}{cccc|c} r(0) & \cdots & r(m-1) & & r(m) \\ \vdots & \ddots & \vdots & & \vdots \\ r^*(m-1) & \cdots & r(0) & & r(1) \\ \hline r^*(m) & \cdots & r^*(1) & & r(0) \end{array}\right] = \begin{bmatrix} \mathbf{R}_m & \mathbf{J}\mathbf{r}_m \\ \mathbf{r}_m^H \mathbf{J} & r(0) \end{bmatrix}$$

$$\mathbf{d}_{m+1} = \begin{bmatrix} \mathbf{d}_m \\ d_{m+1} \end{bmatrix}$$

$$\mathbf{b}_m = -\mathbf{R}_m^{-1} \mathbf{J}\mathbf{r}_m$$

$$P_m^{\mathrm{b}} = r(0) + \mathbf{r}_m^H \mathbf{J}\mathbf{b}_m$$

$$\mathbf{c}_{m+1} = \begin{bmatrix} \mathbf{c}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m^c$$

$$k_m^c \triangleq \frac{\beta_m^c}{P_m^{\mathrm{b}}}$$

$$\beta_m^c \triangleq \mathbf{b}_m^H \mathbf{d}_m + d_{m+1} = -\mathbf{c}_m^H \mathbf{J}\mathbf{r}_m + d_{m+1}$$

- By utilizing the Toeplitz structure of R_m,

$$\mathbf{b}_m = \mathbf{J}\mathbf{a}_m^*$$

$$P_m \triangleq P_m^{\mathrm{b}} = P_m^{\mathrm{f}}$$

- To avoid the use of lower right corner partitioning, FLP recursion can be used to obtain a_m

$$\mathbf{a}_{m+1} = -\mathbf{R}_{m+1}^{-1}\mathbf{r}_{m+1}^*$$

- This leads to the Levinson recursion

$$\mathbf{a}_{m+1} = \begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m$$

$$k_m \triangleq -\frac{\beta_m}{P_m}$$

$$\beta_m \triangleq \mathbf{b}_m^H \mathbf{r}_m^* + r^*(m+1) = \mathbf{a}_m^T \mathbf{J} \mathbf{r}_m^* + r^*(m+1)$$

$$P_m = r(0) + \mathbf{r}_m^H \mathbf{a}_m^* = r(0) + \mathbf{a}_m^T \mathbf{r}_m$$

- Levinson recursion consists of two parts:
  - A set of recursion to compute the FLP or BLP a_m or b_m,
  - A set of recursion to compute the optimum filter from a_m or b_m.

TABLE 7.2

Summary of the Levinson-Durbin algorithm.

1. **Input:** $r(0), r(1), r(2), \ldots, r(M)$
2. **Initialization**
   (a) $P_0 = r(0), \beta_0 = r^*(1)$
   (b) $k_0 = -r^*(1)/r(0), a_1^{(1)} = k_0$
3. **For** $m = 1, 2, \ldots, M-1$
   (a) $P_m = P_{m-1} + \beta_{m-1} k_{m-1}^*$
   (b) $\mathbf{r}_m = [r(1)\ r(2)\ \cdots\ r(m)]^T$
   (c) $\beta_m = \mathbf{a}_m^T \mathbf{J} \mathbf{r}_m^* + r^*(m+1)$
   (d) $k_m = -\dfrac{\beta_m}{P_m}$
   (e) $\mathbf{a}_{m+1} = \begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{J}\mathbf{a}_m^* \\ 1 \end{bmatrix} k_m$
4. $P_M = P_{M-1} + \beta_M k_M^*$
5. **Output:** $\mathbf{a}_M, \{k_m\}_0^{M-1}, \{P_m\}_1^M$

- If required to obtain the coefficients c.

$$(f) \quad \beta_m^c = -\mathbf{c}_m^H \mathbf{J} \mathbf{r}_m + d_{m+1}$$

$$(g) \quad k_m^c = \frac{\beta_m^c}{P_m}$$

$$(h) \quad \mathbf{c}_{m+1} = \begin{bmatrix} \mathbf{c}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{J}\mathbf{a}_m^* \\ 1 \end{bmatrix} k_m^c$$

$$(i) \quad P_{m+1}^c = P_m^c + \beta_m^c k_m^{c*}$$

4. **Output:** $\mathbf{a}_M, \mathbf{c}_M, \{k_m, k_m^c\}_0^{M-1}, \{P_m, P_m^c\}_0^M$

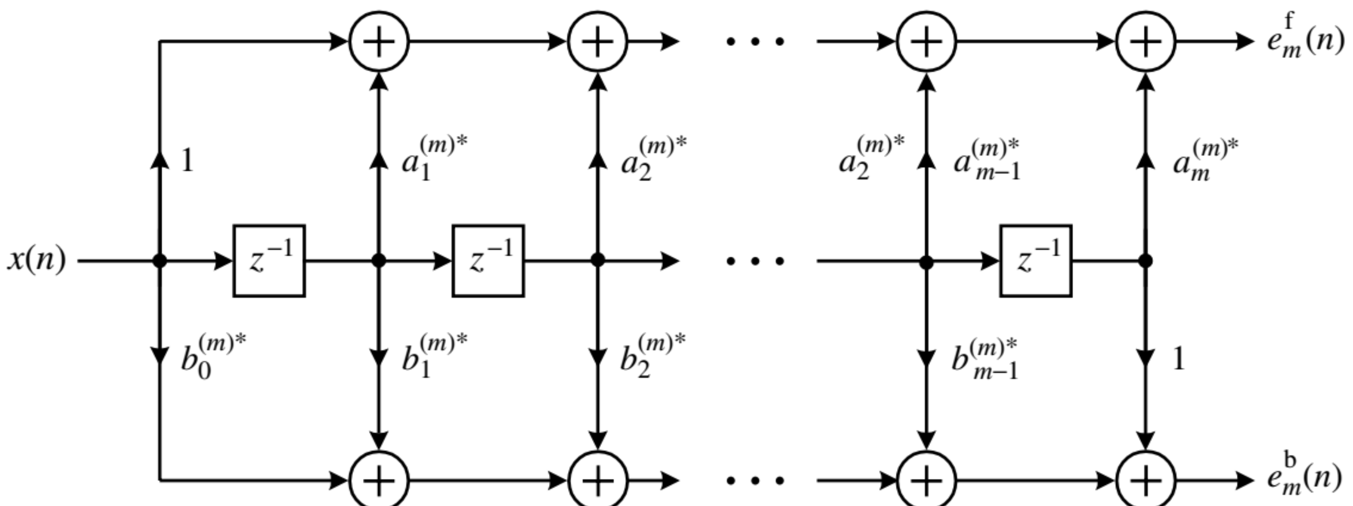
LATTICE STRUCTURES FOR OPTIMUM FIR FILTERS

- To compute the FLP error and BLP error

$$e_m^{\mathrm{f}}(n) = x(n) + \mathbf{a}_m^H \mathbf{x}_m(n-1) = x(n) + \sum_{k=1}^{m} a_k^{(m)*} x(n-k)$$

$$e_m^{\mathrm{b}}(n) = x(n-m) + \mathbf{b}_m^H \mathbf{x}_m(n) = x(n-m) + \sum_{k=0}^{m-1} b_k^{(m)*} x(n-k)$$

- Using direct-form filter structure



- Since a_m and b_m do not have the optimum nesting property, we cannot obtain order-recursive direct-form structures for the computation of the prediction errors.
- By partitioning x,

$$\begin{aligned}\mathbf{x}_{m+1}(n) &= [x(n)\ x(n-1)\ \cdots\ x(n-m+1)\ x(n-m)]^T \\ &= [\mathbf{x}_m^T(n)\ x(n-m)]^T \\ &= [x(n)\ \mathbf{x}_m^T(n-1)]^T\end{aligned}$$

- FLP errors are

$$e^{\mathrm{f}}_{m+1}(n) = x(n) + \left\{ \underbrace{\begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_m \\ 1 \end{bmatrix} k_m}_{\mathbf{a}_{m+1}} \right\}^H \begin{bmatrix} \mathbf{x}_m(n-1) \\ x(n-m-1) \end{bmatrix}$$

$$= x(n) + \mathbf{a}_m^H \mathbf{x}_m(n-1) + k_m^* [\mathbf{b}_m^H \mathbf{x}_m(n-1) + x(n-1-m)]$$

$$e^{\mathrm{f}}_{m+1}(n) = e^{\mathrm{f}}_m(n) + k_m^* e^{\mathrm{b}}_m(n-1)$$

- BLP errors are

$$e^{\mathrm{b}}_{m+1}(n) = x(n-m-1) + \left\{ \begin{bmatrix} 0 \\ \mathbf{b}_m \end{bmatrix} + \begin{bmatrix} 1 \\ \mathbf{a}_m \end{bmatrix} k_m^* \right\}^H \begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix}$$

$$= x(n-m-1) + \mathbf{b}_m^H \mathbf{x}_m(n-1) + k_m [x(n) + \mathbf{a}_m^H \mathbf{x}_m(n-1)]$$

$$e^{\mathrm{b}}_{m+1}(n) = e^{\mathrm{b}}_m(n-1) + k_m e^{\mathrm{f}}_m(n)$$

- These equations can be computed for m=0,1,… ,*M*-1 given initial conditions

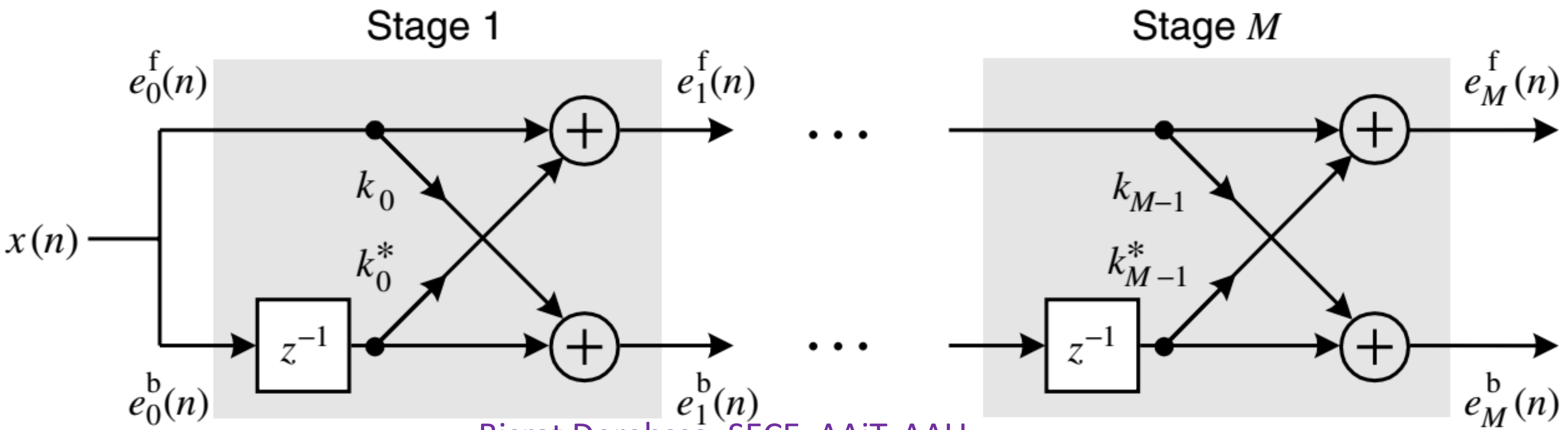
$$e_0^{\mathrm{f}}(n) = e_0^{\mathrm{b}}(n) = x(n)$$

$$e_m^{\mathrm{f}}(n) = e_{m-1}^{\mathrm{f}}(n) + k_{m-1}^* e_{m-1}^{\mathrm{b}}(n-1) \qquad m = 1, 2, \ldots, M$$

$$e_m^{\mathrm{b}}(n) = k_{m-1} e_{m-1}^{\mathrm{f}}(n) + e_{m-1}^{\mathrm{b}}(n-1) \qquad m = 1, 2, \ldots, M$$

$$e(n) = e_M^{\mathrm{f}}(n)$$

- Implementation



- Given that

$$e_{m+1}(n) = e_m(n) - k_m^{c*}(n)e_m^{\mathrm{b}}(n)$$

- The optimum filtering error can be computed from the BLP error.