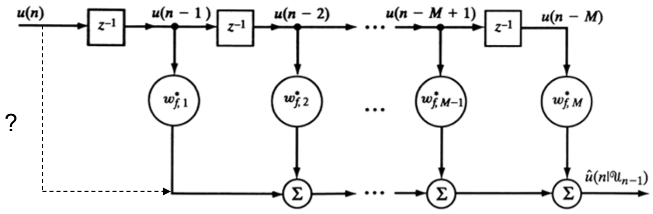
LINEAR PREDICTION

Linear Prediction

- Problem:
  - Forward Prediction
    - Observing
      $$[\ u(n-1),\quad u(n-2),\quad \cdots,\quad u(n-M)\ ]$$
    - Predict
      $$\hat{u}(n|\mathcal{U}_{n-1})$$
  - Backward Prediction
    - Observing
      $$[\ u(n),\quad u(n-1),\quad \cdots,\quad u(n-M+1)\ ]$$
    - Predict
      $$\hat{u}(n-M|\mathcal{U}_{n})$$

Forward Linear Prediction

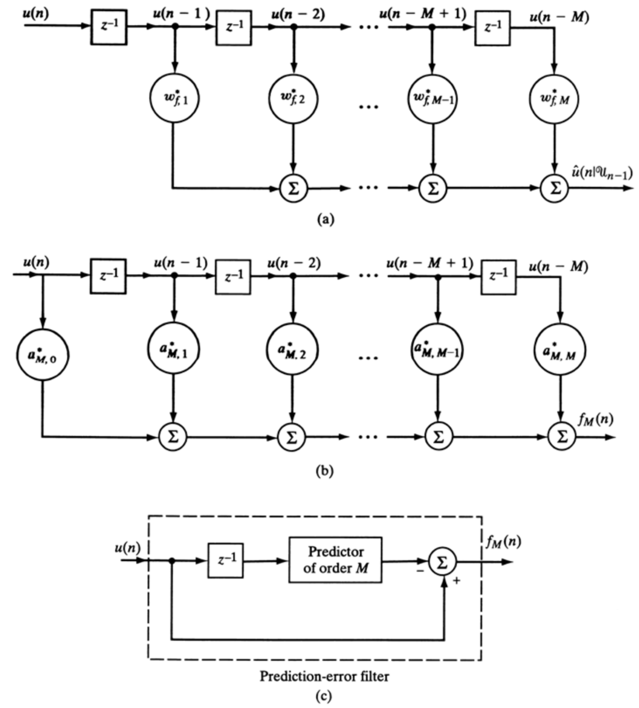
- Problem:
  - Forward Prediction
    - Observing the past
      $$[\ u(n-1),\quad u(n-2),\quad \cdots,\quad u(n-M)\ ]$$
    - Predict the future
      $$\hat{u}(n|\mathcal{U}_{n-1})$$
    - i.e. find the predictor filter taps $w_{f,1}$, $w_{f,2}$,...,$w_{f,M}$



$$\hat{u}(n|\mathcal{U}_{n-1}) = \sum_{k=1}^{M} w_{f,k}^{*} u(n-k)$$

Forward Linear Prediction

- Use Wiener filter theory to calculate $w_{f,k}$
- Desired signal
  $$d(n) = u(n)$$
- Then forward prediction error is (for predictor order M)
  $$f_M(n) = u(n) - \hat{u}(n|\mathcal{U}_{n-1})$$
- Let minimum mean-square prediction error be
  $$P_M = E\{|f_M(n)|^2\},\ \forall n$$



One-step predictor

Prediction-error filter

Forward Linear Prediction

- A structure similar to Wiener filter, same approach can be used.
- For the input vector

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

with the autocorrelation

$$\begin{aligned}\mathbf{R} &= E\{\mathbf{u}(n-1)\mathbf{u}^H(n-1)\} \\ &= \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix}\end{aligned}$$

- Find the filter taps

$$\mathbf{w}_f = \begin{bmatrix} w_{f,1} & w_{f,2} & \cdots & w_{f,M} \end{bmatrix}^T$$

where the cross-correlation bw. the filter input and the desired response is

$$\begin{aligned}\mathbf{r} &= E\{\mathbf{u}(n-1)u^*(n)\} \\ &= \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(M) \end{bmatrix} = \begin{bmatrix} r(-1) \\ r(-2) \\ \vdots \\ r(-M) \end{bmatrix}\end{aligned}$$

Forward Linear Prediction

- Solving the Wiener-Hopf equations, we obtain

$$\boxed{\mathbf{R}\mathbf{w}_f = \mathbf{r}}$$

- and the minimum forward-prediction error power becomes

$$\boxed{P_M = r(0) - \mathbf{r}^H \mathbf{w}_f}$$

- In summary,

TABLE 3.1 Summary of Wiener Filter Variables

Quantity	Wiener filter of Fig. 2.4	Forward predictor of Fig. 3.1(a)	Backward predictor of Fig. 3.2(a)
Tap-input vector	$\mathbf{u}(n)$	$\mathbf{u}(n-1)$	$\mathbf{u}(n)$
Desired response	$d(n)$	$u(n)$	$u(n-M)$
Tap-weight vector	$\mathbf{w}_o$	$\mathbf{w}_f$	$\mathbf{w}_b$
Estimation error	$e(n)$	$f_M(n)$	$b_M(n)$
Correlation matrix of tap inputs	$\mathbf{R}$	$\mathbf{R}$	$\mathbf{R}$
Cross-correlation vector between tap inputs and desired response	$\mathbf{p}$	$\mathbf{r}$	$\mathbf{r}^{B*}$
Minimum mean-square error	$J_{\min}$	P_M	P_M

Relation bw. Linear Prediction and AR Modelling

- Note that the Wiener-Hopf equations for a linear predictor is mathematically identical with the Yule-Walker equations for the model of an AR process.
- If AR model order M is known, model parameters can be found by using a forward linear predictor of order M.
- If the process is not AR, predictor provides an (AR) model approximation of order M of the process.

Forward Prediction-Error Filter

- We wrote that

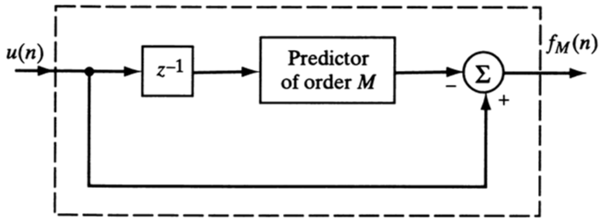
$$f_M(n) = u(n) - \hat{u}(n|\mathcal{U}_{n-1})$$

- Let

$$a_{M,k} = \begin{cases} 1, & k = 0 \\ -w_{f,k}, & k = 1, 2, \cdots, M \end{cases}$$

- Then

$$f_M(n) = \sum_{k=0}^{M} a_{M,k}^* u(n-k)$$



Prediction-error filter

Augmented Wiener-Hopf Eqn.s for Forward Prediction

- Let us combine the forward prediction filter and forward prediction-error power equations in a single matrix expression, i.e.

$$\mathbf{R}\mathbf{w}_f = \mathbf{r} \quad \text{and} \quad P_M = r(0) - \mathbf{r}^H \mathbf{w}_f$$

$$\begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_f \end{bmatrix} = \begin{bmatrix} P_M \\ \mathbf{0} \end{bmatrix}$$

- Define the forward prediction-error filter vector

$$\mathbf{a}_M = \begin{bmatrix} 1 \\ -\mathbf{w}_f \end{bmatrix}$$

- Then

Augmented Wiener-Hopf Eqn.s of a forward prediction-error filter of order M.

$$\mathbf{R}_{M+1}\mathbf{a}_M = \begin{bmatrix} P_M \\ \mathbf{0} \end{bmatrix} \quad \text{or} \quad \sum_{l=0}^{M} a_{M,l} r(l-i) = \begin{cases} P_M, & i = 0 \\ 0, & i = 1, 2, \cdots, M \end{cases}$$

Example – Forward Predictor (order M=1)

- For a forward predictor of order M=1

$$\begin{bmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{bmatrix} \begin{bmatrix} a_{1,0} \\ a_{1,1} \end{bmatrix} = \begin{bmatrix} P_1 \\ 0 \end{bmatrix}$$

- Then

$$a_{1,0} = \frac{P_1}{\Delta_r} r(0) \text{ and } a_{1,1} = \frac{P_1}{\Delta_r} r^*(1)$$

where

$$\Delta_r = \det(\mathbf{R}) = r^2(0) - |r(1)|^2$$

- But $a_{1,0}$=1, then

$$P_1 = \frac{\Delta_r}{r(0)}$$
$$a_{1,1} = -\frac{r^*(1)}{r(0)}$$

Backward Linear Prediction

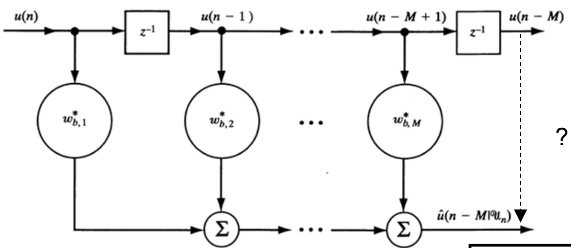
- Problem:
  - Forward Prediction
    - Observing the future

$$[\ u(n), \quad u(n-1), \quad \cdots, \quad u(n-(M-1))\ ]$$

    - Predict the past

$$\hat{u}(n-M|\mathcal{U}_n)$$

    - i.e. find the predictor filter taps $w_{b,1}, w_{b,2},\ldots,w_{b,M}$



$$\hat{u}(n-M|\mathcal{U}_n) = \sum_{k=1}^{M} w_{b,k}^* u(n-k+1)$$

Backward Linear Prediction

- Desired signal

$$d(n) = u(n-M)$$

- Then backward prediction error is (for predictor order M)

$$b_M(n) = u(n-M) - \hat{u}(n-M|\mathcal{U}_n)$$

- Let minimum-mean square prediction error be

$$P_M = E\{|b_M(n)|^2\}, \forall n$$

Backward Linear Prediction

- Problem:
- For the input vector

$$\mathbf{u}(n) = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(n-(M-1)) \end{bmatrix}^T$$

with the autocorrelation

$$\begin{aligned}\mathbf{R} &= E\{\mathbf{u}(n-1)\mathbf{u}^H(n-1)\} \\ &= \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix}\end{aligned}$$

- Find the filter taps

$$\mathbf{w}_b = \begin{bmatrix} w_{b,1} & w_{b,2} & \cdots & w_{b,M} \end{bmatrix}^T$$

where the cross-correlation bw. the filter input and the desired response is

$$\begin{aligned}\mathbf{r}^{B*} &= E\{\mathbf{u}(n)u^*(n-M)\} \\ &= \begin{bmatrix} r(M) \\ r(M-1) \\ \vdots \\ r(1) \end{bmatrix}\end{aligned}$$

Backward Linear Prediction

- Solving the Wiener-Hopf equations, we obtain
- $$\boxed{\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*}}$$
- and the minimum forward-prediction error power becomes

$$\boxed{P_M = r(0) - \mathbf{r}^{BT}\mathbf{w}_b}$$

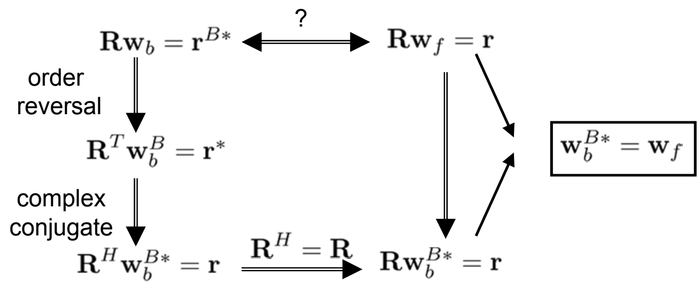
- In summary,

TABLE 3.1 Summary of Wiener Filter Variables

Quantity	Wiener filter of Fig. 2.4	Forward predictor of Fig. 3.1(a)	Backward predictor of Fig. 3.2(a)
Tap-input vector	$\mathbf{u}(n)$	$\mathbf{u}(n-1)$	$\mathbf{u}(n)$
Desired response	$d(n)$	$u(n)$	$u(n-M)$
Tap-weight vector	$\mathbf{w}_o$	$\mathbf{w}_f$	$\mathbf{w}_b$
Estimation error	$e(n)$	$f_M(n)$	$b_M(n)$
Correlation matrix of tap inputs	$\mathbf{R}$	$\mathbf{R}$	$\mathbf{R}$
Cross-correlation vector between tap inputs and desired response	$\mathbf{p}$	$\mathbf{r}$	$\mathbf{r}^{B*}$
Minimum mean-square error	$J_{\min}$	P_M	P_M

Relations bw. Forward and Backward Predictors

- Compare the Wiener-Hopf eqn.s for both cases (**R** and **r** are same)



$$P_M = r(0) - \mathbf{r}^{BT}\mathbf{w}_b \Longrightarrow P_M = r(0) - \mathbf{r}^{H}\mathbf{w}_b^{B*} \Longrightarrow P_M = r(0) - \mathbf{r}^{H}\mathbf{w}_f$$

Backward Prediction-Error Filter

- We wrote that

$$b_M(n) = u(n-M) - \hat{u}(n-M|\mathcal{U}_n)$$

- Let

$$c_{M,k} = \begin{cases} -w_{b,k+1}, & k = 0, 1, \cdots, M-1 \\ 1, & k = M \end{cases}$$

- Then

$$b_M(n) = \sum_{k=0}^{M} c_{M,k}^* u(n-k)$$

but we found that

$$\mathbf{w}_b^{B*} = \mathbf{w}_f \Longrightarrow \begin{array}{l} w_{b,M-k+1}^* = w_{f,k},\ k = 1, 2, \cdots, M \\ \text{or} \\ w_{b,k} = w_{f,M-k+1}^*,\ k = 1, 2, \cdots, M \end{array}$$

$$c_{M,k} = \begin{cases} -w_{f,M-k}^*, & k = 0, 1, \cdots, M-1 \\ 1, & k = M \end{cases} = a_{M,M-k}^*,\ k = 0, 1, \cdots, M$$

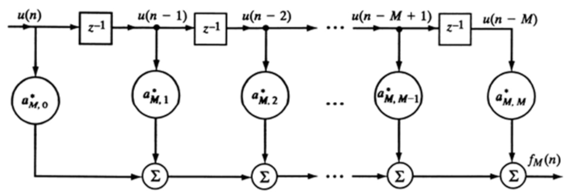
Then

$$\boxed{b_M(n) = \sum_{k=0}^{M} a_{M,M-k} u(n-k)}$$

Backward Prediction-Error Filter

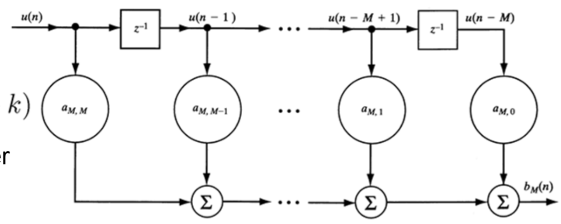
$$f_M(n) = \sum_{k=0}^{M} a_{M,k}^* u(n-k)$$

forward prediction-error filter



$$b_M(n) = \sum_{k=0}^{M} a_{M,M-k} u(n-k)$$

backward prediction-error filter



- For stationary inputs, we may change a forward prediction-error filter into the corresponding backward prediction-error filter by reversing the order of the sequence and taking the complex conjugation of them.

Augmented Wiener-Hopf Eqn.s for Backward Prediction

- Let us combine the backward prediction filter and backward prediction-error power equations in a single matrix expression, i.e.
- $\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*} \qquad P_M = r(0) - \mathbf{r}^{BT}\mathbf{w}_b$

$$\begin{bmatrix} \mathbf{R} & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix}$$

- With the definition

$$\mathbf{a}_M^{B*} = \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix}$$

- Then

Augmented Wiener-Hopf Eqn.s of a backward prediction-error filter of order M.

- $\mathbf{R}_{M+1}\mathbf{a}_M^{B*} = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix}$ $\quad \sum_{l=0}^{M} a_{M,M-l}^{*} r(l-i) = \begin{cases} 0, & i = 0, \cdots, M-1 \\ P_M, & i = M \end{cases}$

Levinson-Durbin Algorithm

- Solve the following Wiener-Hopf eqn.s to find the predictor coef.s

$$\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*} \qquad \mathbf{R}\mathbf{w}_f = \mathbf{r}$$

- One-shot solution can have high computation complexity.
- Instead, use an (order)-recursive algorithm
  - Levinson-Durbin Algorithm.
  - Start with a first-order (m=1) predictor and at each iteration increase the order of the predictor by one up to (m=M).
  - Huge savings in computational complexity and storage.

Levinson-Durbin Algorithm

- Let the forward prediction error filter of order m be represented by the (m+1)x1 $\mathbf{a}_M$

and its order reversed and complex conjugated version (backward prediction error filter) be $\mathbf{a}_M^{B*}$

- The forward-prediction error filter can be order-updated by

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \qquad a_{m,l} = a_{m-1,l} + \kappa_m a_{m-1,m-l}^{*},\ l = 0, 1, \cdots, m$$

- The backward-prediction error filter can be order-updated by

$$\mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + \kappa_m \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} \qquad a_{m,m-l}^{*} = a_{m-1,m-l}^{*} + \kappa_m^{*} a_{m-1,l},\ l = 0, 1, \cdots, m$$

where the constant κ_m is called the reflection coefficient.

Levinson-Durbin Recursion

- How to calculate $\mathbf{a}_m$ and κ_m?
- Start with the relation bw. correlation matrix $\mathbf{R}_{m+1}$ and the forward-error prediction filter $\mathbf{a}_m$.

indicates order

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

indicates dim. of matrix/vector

- We have seen how to partition the correlation matrix

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

Levinson-Durbin Recursion

- Multiply the order-update eqn. by $\mathbf{R}_{m+1}$ from the left

$$\mathbf{R}_{m+1}\mathbf{a}_m = \underbrace{\mathbf{R}_{m+1}\begin{bmatrix}\mathbf{a}_{m-1}\\0\end{bmatrix}}_{①} + \kappa_m\underbrace{\mathbf{R}_{m+1}\begin{bmatrix}0\\\mathbf{a}_{m-1}^{B*}\end{bmatrix}}_{②}$$

- Term 1:

$$\mathbf{R}_{m+1}\begin{bmatrix}\mathbf{a}_{m-1}\\0\end{bmatrix} = \begin{bmatrix}\mathbf{R}_m & \mathbf{r}_m^{B*}\\\mathbf{r}_m^{BT} & r(0)\end{bmatrix}\begin{bmatrix}\mathbf{a}_{m-1}\\0\end{bmatrix} = \begin{bmatrix}\mathbf{R}_m\mathbf{a}_{m-1}\\\mathbf{r}_m^{BT}\mathbf{a}_{m-1}\end{bmatrix}$$

but we know that (augmented Wiener-Hopf eqn.s)

$$\mathbf{R}_m\mathbf{a}_{m-1} = \begin{bmatrix}P_{m-1}\\\mathbf{0}_{m-1}\end{bmatrix}$$

- Then

$$\mathbf{R}_{m+1}\begin{bmatrix}\mathbf{a}_{m-1}\\0\end{bmatrix} = \begin{bmatrix}P_{m-1}\\\mathbf{0}_{m-1}\\\Delta_{m-1}\end{bmatrix} \text{ where } \Delta_{m-1} = \mathbf{r}_m^{BT}\mathbf{a}_{m-1}$$

Levinson-Durbin Recursion

- Term 2:

$$\mathbf{R}_{m+1}\begin{bmatrix}0\\\mathbf{a}_{m-1}^{B*}\end{bmatrix} = \begin{bmatrix}r(0) & \mathbf{r}_m^{H}\\\mathbf{r}_m & \mathbf{R}_m\end{bmatrix}\begin{bmatrix}0\\\mathbf{a}_{m-1}^{B*}\end{bmatrix} = \begin{bmatrix}\mathbf{r}_m^H\mathbf{a}_{m-1}^{B*}\\\mathbf{R}_m\mathbf{a}_{m-1}^{B*}\end{bmatrix}$$

but we know that (augmented Wiener-Hopf eqn.s)

$$\mathbf{R}_m\mathbf{a}_{m-1}^{B*} = \begin{bmatrix}\mathbf{0}_{m-1}\\P_{m-1}\end{bmatrix}$$

- Then

$$\mathbf{R}_{m+1}\begin{bmatrix}0\\\mathbf{a}_{m-1}^{B*}\end{bmatrix} = \begin{bmatrix}\Delta_{m-1}^{*}\\\mathbf{0}_{m-1}\\P_{m-1}\end{bmatrix} \text{ where } \Delta_{m-1} = \mathbf{r}_m^{BT}\mathbf{a}_{m-1}$$

Levinson-Durbin Recursion

$$\mathbf{R}_{m+1}\mathbf{a}_m = \mathbf{R}_{m+1}\begin{bmatrix}\mathbf{a}_{m-1}\\0\end{bmatrix} + \kappa_m \mathbf{R}_{m+1}\begin{bmatrix}0\\ \mathbf{a}^{B*}_{m-1}\end{bmatrix}$$

$$\boxed{\begin{bmatrix}P_m\\ \mathbf{0}_m\end{bmatrix} = \begin{bmatrix}P_{m-1}\\ \mathbf{0}_{m-1}\\ \Delta_{m-1}\end{bmatrix} + \kappa_m \begin{bmatrix}\Delta^*_{m-1}\\ \mathbf{0}_{m-1}\\ P_{m-1}\end{bmatrix}}$$

- Then we have
  - from the first line

    $$P_m = P_{m-1} + \kappa_m \Delta^*_{m-1}$$

  - from the last line

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

$$P_m = P_{m-1}(1-|\kappa_m|^2)$$

As iterations increase P_m decreases

$$P_0 = r(0)$$

$$0 \le P_m \le P_{m-1},\ m \ge 1$$

Levinson-Durbin Recursion - Interpretations

$$P_m = P_{m-1}(1-|\kappa_m|^2) \longrightarrow \boxed{P_M = P_0 \prod_{m=1}^{M}(1-|\kappa_m|^2)}$$

final value of the prediction error power

- κ_m: reflection coef.s due to the analogy with the reflection coef.s corresponding to the boundary bw. two sections in transmission lines

  $$|\kappa_m| \le 1,\ \forall m \text{ and } \boxed{\kappa_m = a_{m,m}}$$

- The parameter Δ_m represents the crosscorrelation bw. the forward prediction error and the *delayed* backward prediction error

  $$\Delta_{m-1} = E\{b_{m-1}(n-1)f^*_{m-1}(n)\}$$

  HW: Prove this!

- Since $f_0(n)$= $b_0(n)$= u(n)

$$\begin{aligned}\Delta_0 &= E\{b_0(n-1)f^*_0(n)\}\\ &= E\{u(n-1)u^*(n)\}\\ &= r(-1) = r^*(1)\end{aligned}$$

Application of the Levinson-Durbin Algorithm

- Find the forward prediction error filter coef.s $a_{m,k}$, given the autocorrelation sequence {r(0), r(1), r(2)}

$$a_{m,0} = 1$$
$$\Delta_{m-1} = \mathbf{r}_m^{BT}\mathbf{a}_{m-1}$$
$$\kappa_m = -\frac{\Delta_{m-1}}{P_{m-1}} = a_{m,m}$$
$$a_{m,l} = a_{m-1,l} + \kappa_m a^*_{m-1,m-l}$$
$$P_m = P_{m-1}(1 - |\kappa_m|^2)$$

- m=0

$$a_{0,0} = 1,\ P_0 = r(0)$$

- m=1

$$a_{1,0} = 1$$
$$\Delta_0 = r(-1)$$
$$\kappa_1 = -\frac{\Delta_0}{P_0} = -\frac{r(-1)}{r(0)} = a_{1,1}$$
$$P_1 = P_0(1 - |\kappa_1|^2) = r(0)\left(1 - \frac{r^2(-1)}{r^2(0)}\right)$$

- m=M=2

$$a_{2,0} = 1,$$
$$\begin{aligned}\Delta_1 &= a_{1,0}r(-2) + a_{1,1}r(-1)\\ &= r(-2) + \kappa_1 r(-1)\\ &= r(-2) - \frac{r^2(-1)}{r(0)}\end{aligned}$$
$$\kappa_2 = -\frac{\Delta_1}{P_1} = a_{2,2}$$
$$P_2 = P_0\prod_{k=1}^{2}(1 - |\kappa_k|^2)$$

Properties of the prediction error filters

- **Property 1:** There is a one-to-one correspondence bw. the two sets of quantities {P_0, κ_1, κ_2, ... ,κ_M} and {r(0), r(1), ..., r(M)}.
  - If one set is known the other can directly be computed by:

$$\kappa_m = -\frac{1}{P_{m-1}}\sum_{k=0}^{m-1} a_{m-1,k}r(k-m)$$
and
$$r(m) = -\kappa_m^* P_{m-1} - \sum_{k=1}^{m-1} a_{m-1,k}r(m-k)$$

Properties of the prediction error filters

- **Property 2a:** Transfer function of a forward prediction error filter
$$\{a_{m,k}^*\} \rightarrow H_{f,m}(z) = \sum_{k=0}^{m} a_{m,k}^* z^{-k}$$
- Utilizing Levinson-Durbin recursion $a_{m,l} = a_{m-1,l} + \kappa_m a_{m-1,m-l}^*$
$$\begin{aligned} H_{f,m}(z) &= \sum_{k=0}^{m} a_{m-1,k}^* z^{-k} + \kappa_m^* \sum_{k=0}^{m} a_{m-1,m-k} z^{-k} \\ &= \underline{\sum_{k=0}^{m-1} a_{m-1,k}^* z^{-k}} + \kappa_m^* \underline{\sum_{k=0}^{m-1} a_{m-1,m-1-k} z^{-k}} \end{aligned}$$
  but we also have
$$H_{f,m-1}(z) = \sum_{k=0}^{m} a_{m-1,k}^* z^{-k} \qquad H_{b,m-1}(z) = \sum_{k=0}^{m} a_{m-1,m-1-k} z^{-k}$$
- Then
$$\boxed{H_{f,m}(z) = H_{f,m-1}(z) + \kappa_m^* z^{-1} H_{b,m-1}(z)}$$

Properties of the prediction error filters

- **Property 2b:** Transfer function of a backward prediction error filter
$$\{a_{m,m-k}^*\} \rightarrow H_{b,m}(z) = \sum_{k=0}^{m} a_{m,m-k}^* z^{-k}$$
- Utilizing Levinson-Durbin recursion $a_{m,m-l}^* = a_{m-1,m-l}^* + \kappa_m^* a_{m-1,l}$
$$\boxed{H_{b,m}(z) = z^{-1} H_{b,m-1}(z) + \kappa_m^* H_{f,m-1}(z)}$$
- Given the reflection coef.s κ_m and the transfer functions of the forward and backward prediction-error filters of order m-1, we can uniquely calculate the corresponding transfer functions for the forward and backward prediction error filters of order m.

Properties of the prediction error filters

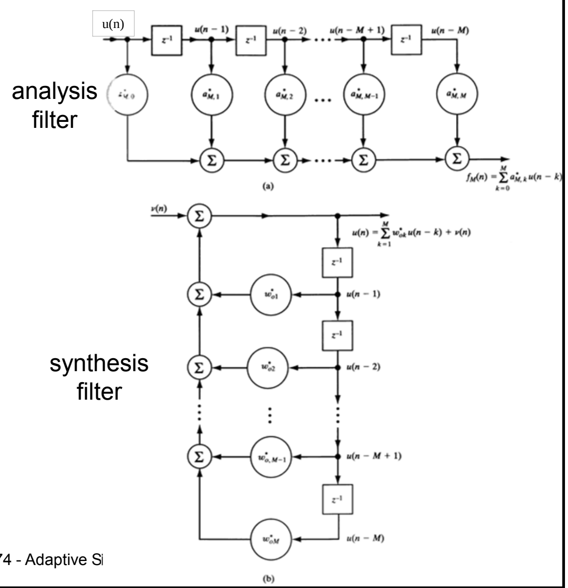
- **Property 3:** Both the forward and backward prediction error filters have the same magnitude response

$$|H_{f,m}(z)| = |H_{b,m}(z)|,\; z = e^{j\omega}$$

- **Property 4:** Forward prediction-error filter is minimum-phase.
  - causal and has stable inverse.
- **Property 5:** Backward prediction-error filter is maximum-phase.
  - non-causal and has unstable inverse.

Properties of the prediction error filters

- **Property 6**: Forward prediction-error filter is a whitening filter.
  - We have seen that a forward prediction-error filter can estimate an AR model (analysis filter).



Properties of the prediction error filters

- **Property 7:** Backward prediction errors are orthogonal to each other.

$$E\{b_m(n)b_i^*(n)\} = \begin{cases} P_m, & i = m \\ 0, & i \leq m \end{cases}$$

($b_i(n), \forall i$ are white)

- Proof: Comes from principle of orthogonality, i.e.:

$$E\{b_m(n)u^*(n-k)\} = 0, \; 0 \leq k \leq m-1$$

(HW: continue the proof)

Lattice Predictors

- A very efficient structure to implement the forward/backward predictors.
- Rewrite the prediction error filter coef.s

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}$$

$$\mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + \kappa_m \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix}$$

- The input signal to the predictors {u(n), n(n-1),...,u(n-M)} can be stacked into a vector

$$\mathbf{u}_{m+1}(n) = \left[\frac{\mathbf{u}_m(n)}{u(n-m)}\right] = \left[\frac{u(n)}{\mathbf{u}_m(n-1)}\right]$$

- Then the output of the predictors are

$$f_m(n) = \mathbf{a}_m^H \mathbf{u}_{m+1}(n) \qquad b_m(n) = \mathbf{a}_m^{B*} \mathbf{u}_{m+1}(n)$$

(forward) (backward)

Lattice Predictors

- Forward prediction-error filter

$$f_m(n) = \mathbf{a}_m^H \mathbf{u}_{m+1}(n) \Longleftarrow \mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}$$

- First term

$$\begin{aligned} \begin{bmatrix} \mathbf{a}_{m-1}^H \mid 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_m(n) \\ \hline u(n-m) \end{bmatrix} &= \mathbf{a}_{m-1}^H \mathbf{u}_m(n) \\ &= f_{m-1}(n) \end{aligned}$$

- Second term

$$\begin{aligned} \begin{bmatrix} 0 \mid \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} u(n) \\ \hline \mathbf{u}_m(n-1) \end{bmatrix} &= \mathbf{a}_{m-1}^{B*} \mathbf{u}_m(n-1) \\ &= b_{m-1}(n-1) \end{aligned}$$

- Combine both terms

$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1)$$

Lattice Predictors

- Similarly, Backward prediction-error filter

$$b_m(n) = \mathbf{a}_m^{BT} \mathbf{u}_{m+1}(n) \Longleftarrow \mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + \kappa_m \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix}$$

- First term

$$\begin{aligned} \begin{bmatrix} 0 \mid \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} u(n) \\ \hline \mathbf{u}_m(n-1) \end{bmatrix} &= \mathbf{a}_{m-1}^{B*} \mathbf{u}_m(n-1) \\ &= b_{m-1}(n-1) \end{aligned}$$

- Second term

$$\begin{aligned} \begin{bmatrix} \mathbf{a}_{m-1}^H \mid 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_m(n) \\ \hline u(n-m) \end{bmatrix} &= \mathbf{a}_{m-1}^H \mathbf{u}_m(n) \\ &= f_{m-1}(n) \end{aligned}$$

- Combine both terms

$$b_m(n) = b_{m-1}(n-1) + \kappa_m f_{m-1}(n)$$

Lattice Predictors

- Forward and backward prediction-error filters

$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1)$$

$$b_m(n) = b_{m-1}(n-1) + \kappa_m f_{m-1}(n)$$

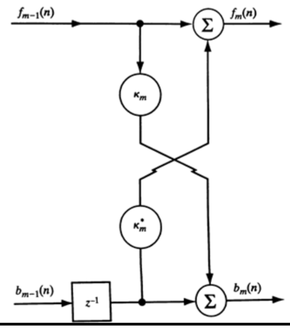
in matrix form

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

and

$$b_{m-1}(n-1) = z^{-1} b_{m-1}(n)$$

Last two equations define the m-th stage of the lattice predictor

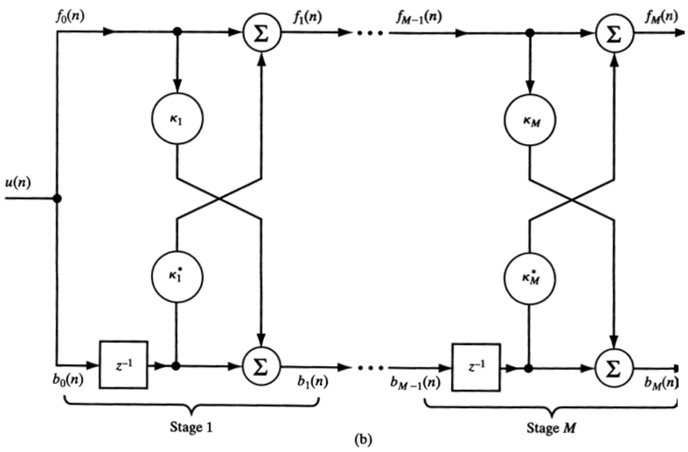


Lattice Predictors

- For m=0 we have $f_0(n) = b_0(n) = u(n)$, hence for M stages

$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1)$$

$$b_m(n) = b_{m-1}(n-1) + \kappa_m f_{m-1}(n)$$



(b)