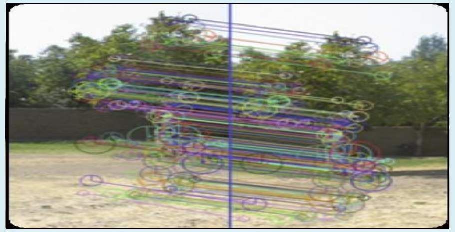
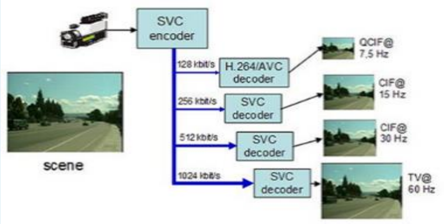
Adaptive System and Signal Processing Theory
(#8: Wiener Filter and Linear Prediction)

2023 Spring

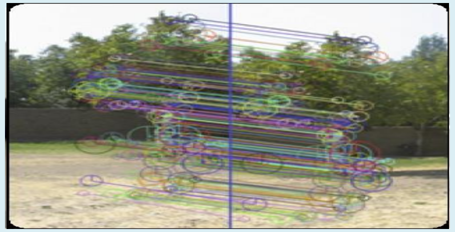
Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
http://ivpl.sookmyung.ac.kr
Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@ivpl.sookmyung.ac.kr

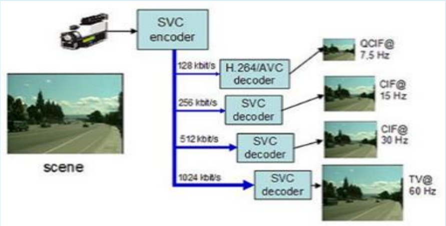




Contents

- Summary of the Previous Lecture
- Wiener Filter
- Linear Prediction
- Levinson-Durbin Algorithm





Contents

- **Summary of the Previous Lecture**
- **Wiener Filter**
- **Linear Prediction**
- **Levinson-Durbin Algorithm**

Remind (1): Stationary Process and Models and Wiener Filter

- Stochastic Models
  - Model: Any hypothesis that may be applied to explain or describe the hidden laws that are supposed to govern or constrain the generation of physical data of interest. → Yule (1927).
- Three Popular Types of Stochastic Models
  - AR (autoregressive) : Not use past input of the model.
  - MA (moving average) : Not use past output of the model.
  - ARMA (autoregressive-moving average): both are available.
- <u>Considerations</u>
  - In terms of computation, AR model is usually better easy.
    - A system (set) of linear equations → Yule-Walker equations
    - ARMA/MA model: very complex to solve and so many nonlinear cases.

Remind (2): Stationary Process and Models and Wiener Filter

- Yule-Walker Eq.

For any AR process $u[n]$,

$$a_0^* r(l) + a_1^* r(l-1) + a_2^* r(l-2) + \cdots + a_M^* r(l-M) = 0,$$

$$\therefore \ r(l) = -a_1^* r(l-1) - a_2^* r(l-2) - \cdots - a_M^* r(l-M).$$

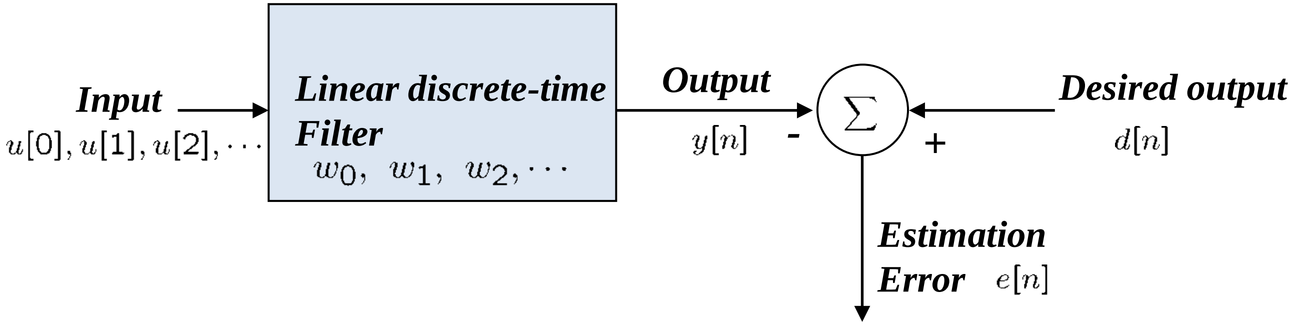
Matrix Form:

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(M) \end{bmatrix},$$

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

$$\therefore \ \mathbf{w} = \mathbf{R}^{-1}\mathbf{r}$$

- Linear Optimum Filtering
  - Main requirement:
    - Make $e[n]$ as smaller as possible in some statistical sense.



  - Factors for Filter Design
    - Whether the impulse response of the filter has finite or infinite duration.
    - What kind of statistical criterion used for the optimization.

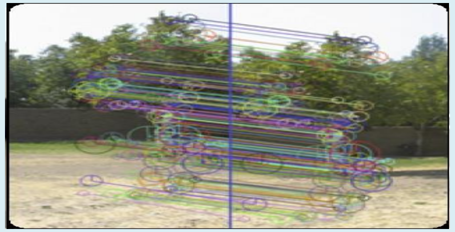
Cost function or index of performance

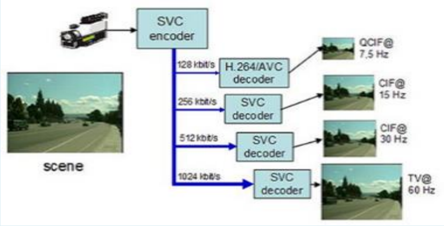
- What kinds of Cost Functions:
  - Mean-square value of the estimation error.
  - Expectation of the absolute value of the estimation error.
  - Expectation of the third or higher order of ~ .

- How to Solve Mathematically the statistical Optimization Problem:
  - Principle of orthogonality.
    - Gradient operation:
      - Used in the context of finding the stationary points of a function of interest.
    - Input and error signal $\therefore\ E\left[u[n-k]e_o^*[n]\right] = 0,\ k = 0, 1, \cdots.$
    - Output and error signal $\therefore\ E\left[y_o[n]e_o^*[n]\right] = 0.$
    - Minimum Mean-Square Error (MSE)

$$\therefore\ J_{min} = \sigma_d^2 \ - \ \sigma_{\hat{d}}^2 \text{ in MSE sense.}$$

    - Winer-Hopf Equation : $\therefore\ \sum_{i=0}^{\infty} w_{oi} r(i-k) = p(-k),\ \text{for } k = 0, 1, \cdots.$
  - Error-performance surface on the filter's coefficients.





Contents

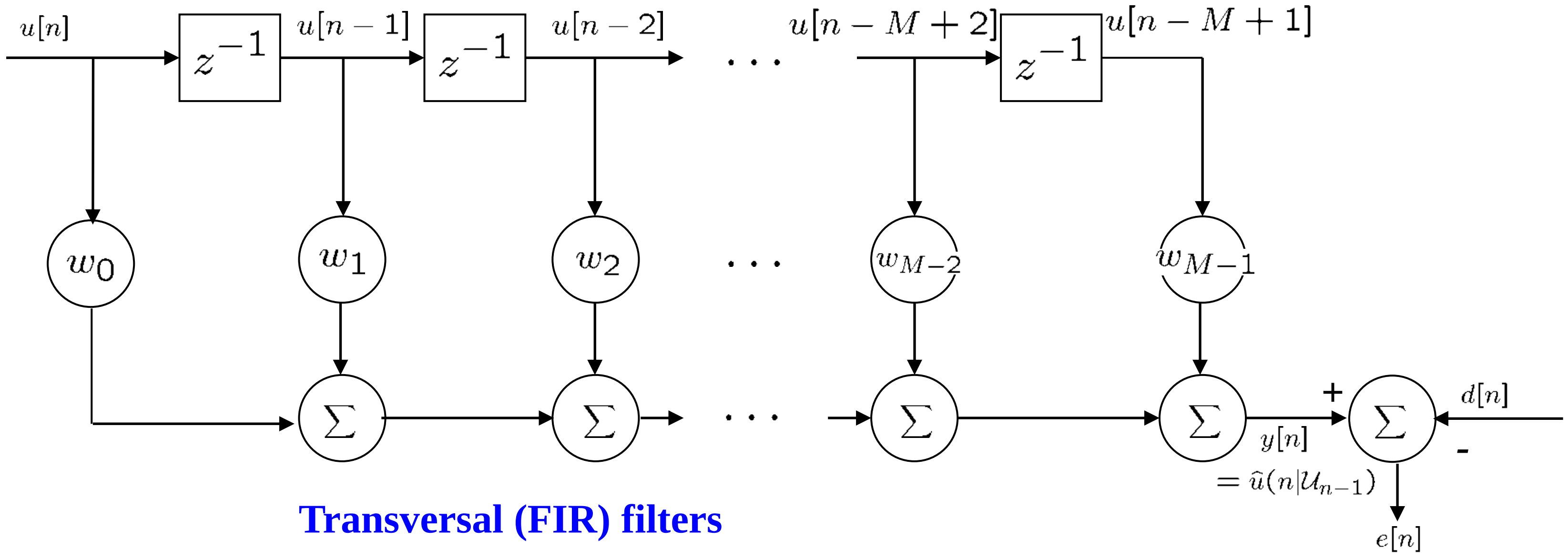
- Summary of the Previous Lecture
- **Wiener Filter**
- Linear Prediction
- Levinson-Durbin Algorithm

- Solution of Wiener-Hopf Equations for Linear Transversal Filter

  If the filter is in the optimum condition,

$$\therefore \sum_{i=0}^{M} w_{oi} r(i-k) = p(-k), \text{ for } k = 0, 1, \cdots, M-1.$$

  Then we can expand this equation for all finite number *M* like as:



Transversal (FIR) filters

Wiener Filters: Linear Optimum Filters (12) : Wiener-Hopf Equations (3)

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} p(0) \\ p(-1) \\ \vdots \\ p(1-M) \end{bmatrix},$$

where $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$ and $\mathbf{p} = E[\mathbf{u}(n)d(n)]$.

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}$$

If $\mathbf{R}$ is nonsingular,

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$$

- The Correlation matrix $\mathbf{R}$,
- The cross-correlation matrix $\mathbf{p}$.

Wiener Filters: Linear Optimum Filters (13) : Error-Performance Surface (1)

- Cost Function
  - Dependent on the weights of the linear filter.

$$J = F(\mathbf{w}).$$

From the original definition of J,

$$J = E\Big[e[n]e^*[n]\Big],$$

$$e[n] = d[n] - y[n],$$
$$= d[n] - \sum_{k=0}^{M-1} w_k u[n-k].$$

$$= E\Big[|d[n]|^2\Big] - \sum w_k^* E\Big[u[n-k]d^*[n]\Big] - \sum w_k E\Big[u^*[n-k]d[n]\Big] + \sum_k \sum_i w_k^* w_i E\Big[u^*[n-k]u[n-i]\Big].$$

$E[|d[n]|^2] = \sigma_d^2$
witn zero mean.

$$E[u[n-k]d^*[n]] = p(-k)$$

$$E[u^*[n-k]d[n]] = p^*(-k) \quad E[u^*[n-k]u[n-i]] = r(i-k)$$

Therefore,

(*M*+*1*) dim. Surface: error-performance surface

$$J = F(\mathbf{w}),$$
$$= \sigma_d^2 - \sum w_k^* p(-k) - \sum w_k^* p^*(-k) + \sum_k \sum_i w_k^* w_i r(i-k).$$

- Bottom point (minimum condition):

  By using the gradient operator for the defined cost function,

$$\nabla_k J = \frac{\partial J}{\partial a_k} + j\frac{\partial J}{\partial b_k} = 0, \;\; k = 0, 1, 2, \cdots .$$

$$\Big\downarrow \text{since } w_k = a_k + jb_k, \;\; k = 0, 1, 2, \cdots$$

$$\nabla_k J = -2p(-k) + 2\sum_{i=0}^{M-1} w_i r(i-k) = 0. \;\; k = 0, 1, 2, \cdots .$$

For the optimum point:

$$\therefore \;\; \underline{\sum_{i=0}^{M-1} w_i r(i-k) = p(-k)}, \;\; k = 0, 1, 2, \cdots .$$

Wiener-Hopf Equation

- Minimum-mean squared error:

Let $\hat{d}[n|\mathcal{U}_n]$ denote the estimate of the desired response $d[n]$.

$$\hat{d}[n|\mathcal{U}_n] = \sum_{k=0}^{M-1} w_{ok}^* u[n-k]$$
$$= \mathbf{w}_o^H \mathbf{u}(n).$$

To evaluate the variance of $\hat{d}[n|\mathcal{U}_n]$,

$$\sigma_{\hat{d}}^2 = E\left[\hat{d}[n|\mathcal{U}_n]\hat{d}^*[n|\mathcal{U}_n]\right],$$
$$= E\left[\mathbf{w}_o^H \mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}_o\right],$$
$$= \mathbf{w}_o^H \boxed{E\left[\mathbf{u}(n)\mathbf{u}^H(n)\right]} \mathbf{w}_o,$$
$$= \mathbf{w}_o^H \mathbf{R} \mathbf{w}_o, \longleftarrow \mathbf{R}\mathbf{w}_o = \mathbf{p}$$

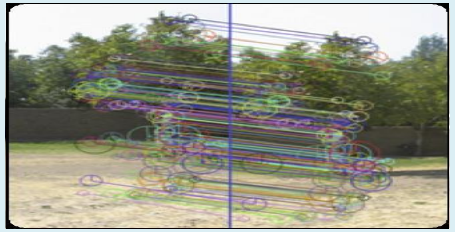
Then

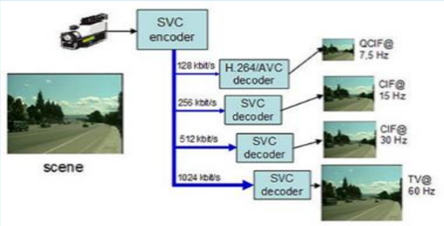
$$\sigma_{\hat{d}}^2 = \mathbf{w}_o^H \mathbf{p},$$
$$= \mathbf{p}^H \mathbf{w}_o.$$

IVPL
Intelligent Vision Processing Lab

- Minimum-mean squared error:

$$
\begin{aligned}
J_{min} &= \sigma_d^2 - \sigma_{\hat{d}}^2, \\
&= \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o, \\
&= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p}.
\end{aligned}
$$





Contents

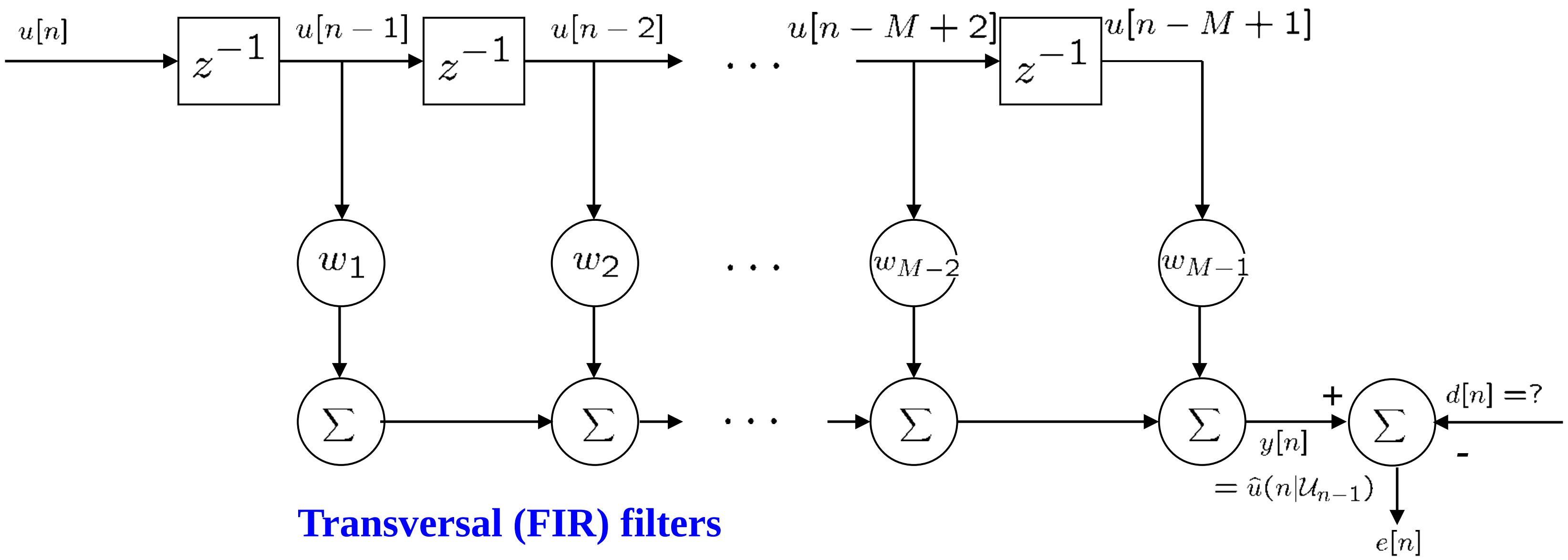
- Summary of the Previous Lecture
- Wiener Filter
- **Linear Prediction**
- Levinson-Durbin Algorithm

Linear Prediction (1) (Chap. 6)

- What is "Prediction"?
  - One of the most celebrated problems.
  - Want to know a future value of a stationary discrete-time stochastic process, given a set of past samples of the process.
- Some Notations
  - $\mathcal{U}_{n-1}$: M-dim. Space spanned by the samples $u[n-1], u[n-2], \cdots, u[n-M]$.
  - $\hat{u}(n|\mathcal{U}_{n-1})$: predicted value of $u[n]$ given the past samples.
- Linear Prediction
  - As linear combination of the given samples $u[n-1], u[n-2], \cdots, u[n-M]$.
  - One step prediction: Forward/Backward linear prediction (FLP/BLP)
    - One step forward prediction: $u[n-1], u[n-2], \cdots, u[n-M] \longrightarrow \hat{u}(n|\mathcal{U}_{n-1})$
    - One step backward prediction: $u[n], u[n-1], \cdots, u[n-(M-1)] \longrightarrow \hat{u}(n-M|\mathcal{U}_n)$

- The Purpose of This Section
  - To optimize the design of the FLP/BLP using Winer Filter Theory in the sense of MSE.
- <u>Forward Linear Prediction</u>

  Let's start with transversal filter of the order of M and M tap-weights with wide-sense stationary stochastic process of zero mean.



Transversal (FIR) filters

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(M) \end{bmatrix},$$

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

So we can solve this as if $\mathbf{R}$ is nonsingular,

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

AR coefficients

- The variance of the white noise:

Let l = 0, then $E[\nu[n]u^*[n]] = E[\sum_{k=0}^{M} a_k^* u[n-k]u * [n]]$

$$\therefore \quad \sum_{k=0}^{M} a_k^* r(k) = \sigma_\nu^2.$$

Linear Prediction (3): Forward Linear Prediction

- The predicted value

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

Since the desired signal is the current input,

- Forward prediction error: $f_M(n) = d[n] - \hat{u}(n|\mathcal{U}_{n-1}) = u[n] - \hat{u}(n|\mathcal{U}_{n-1}).$

To change into the form of Winer-Hopf Eq.,

$$\mathbf{w}_f = [w_{f1}, w_{f2}, \cdots, w_{fM}]^T$$

$$\mathbf{R} = E\left[\mathbf{u}[n-1]\mathbf{u}^H[n-1]\right]$$

$$\mathbf{p} = E\left[\mathbf{u}[n-1]d^*[n]\right]$$

$\Rightarrow$ $\mathbf{R}\mathbf{w}_f = \mathbf{p}$ or

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

IVPL Intelligent Vision Processing Lab

Linear Prediction (4): Forward Linear Prediction

- Forward prediction-error power (P_M)

$$P_M = r(0) - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p}, \longleftarrow \quad = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p}.$$

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

- Relationship betw. Linear prediction and AR modeling
  - Winer-Hopf equation
  - Yule-Walker equation

For AR model

These two system of simultaneous equations are of exactly same mathematical form..

For the case of AR process, when a forward predictor is optimized in the MSE, in theory, its tap-weights take on the same values as the corresponding parameters of the stochastic process.

Linear Prediction (5): Forward Prediction-Error Filter(1)

- Forward prediction-error filter
  - Output: forward prediction-error (FPE)
  - Forward prediction-error

$$f_M(n) = d[n] - \hat{u}(n|\mathcal{U}_{n-1}) = u[n] - \hat{u}(n|\mathcal{U}_{n-1}). \longleftarrow \hat{u}(n|\mathcal{U}_{n-1}) = \sum_{k=1}^{M} w_{f,k} u[n-k]$$

Let $a_{M,k}(k = 0, 1, ..., M)$ denote the tap-weights of new filter as the following:

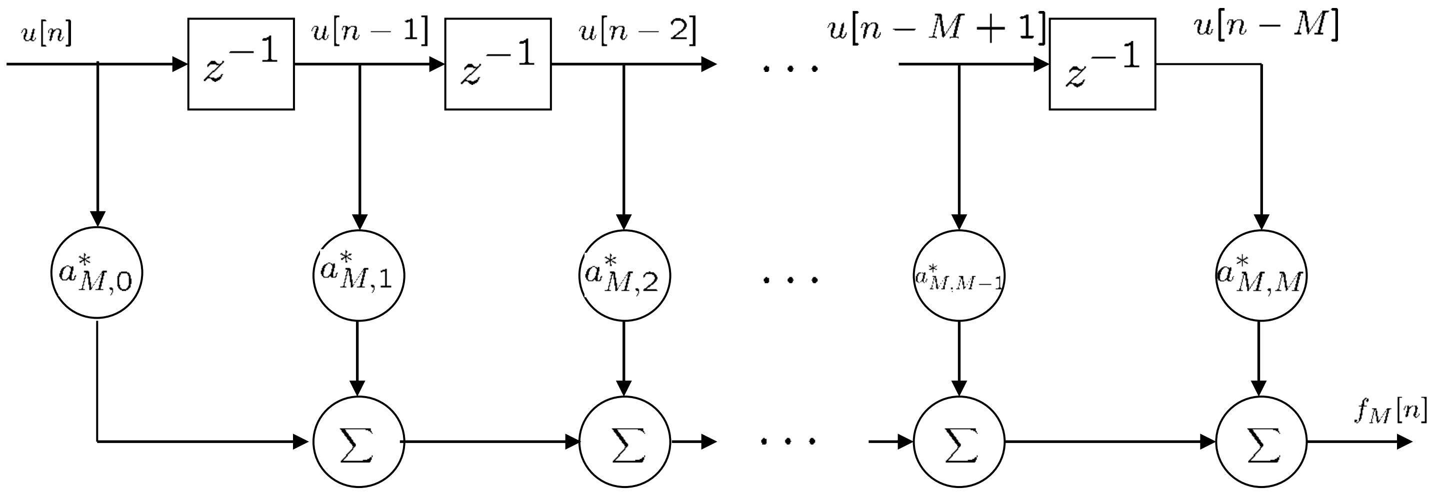
$$a_{M,k} = \begin{cases} 1, & k = 0, \\ -w_{f,k}, & k = 1, 2, .., M. \end{cases}$$

Then,

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

IVPL
Intelligent Vision Processing Lab

Linear Prediction (6): Forward Prediction-Error Filter(2)



FPE filters

- Augmented Winer-Hopf Equations for Forward Prediction

  Using Eqs. of $P_M = r(0) - \mathbf{r}^H \mathbf{w}_f$ and $\mathbf{R}\mathbf{w}_f = \mathbf{r}$,

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

Linear Prediction (7): Forward Prediction-Error Filter(3)

From the Eq., if we let $\begin{bmatrix} 1 \\ -\mathbf{w}_f \end{bmatrix} = \mathbf{a}_M$ then we can rewrite as a system of (M+1) linear equations:

$$\sum_{l=0}^{M} a_{M,l} r[l-i] = \begin{cases} P_M, & i = 0, \\ 0, & i = 1, 2, \cdots, M. \end{cases}$$

or

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

***Augmented Winer-Hopf equations** for forward prediction-error filter*

Linear Prediction (8): Backward Linear Prediction (1)

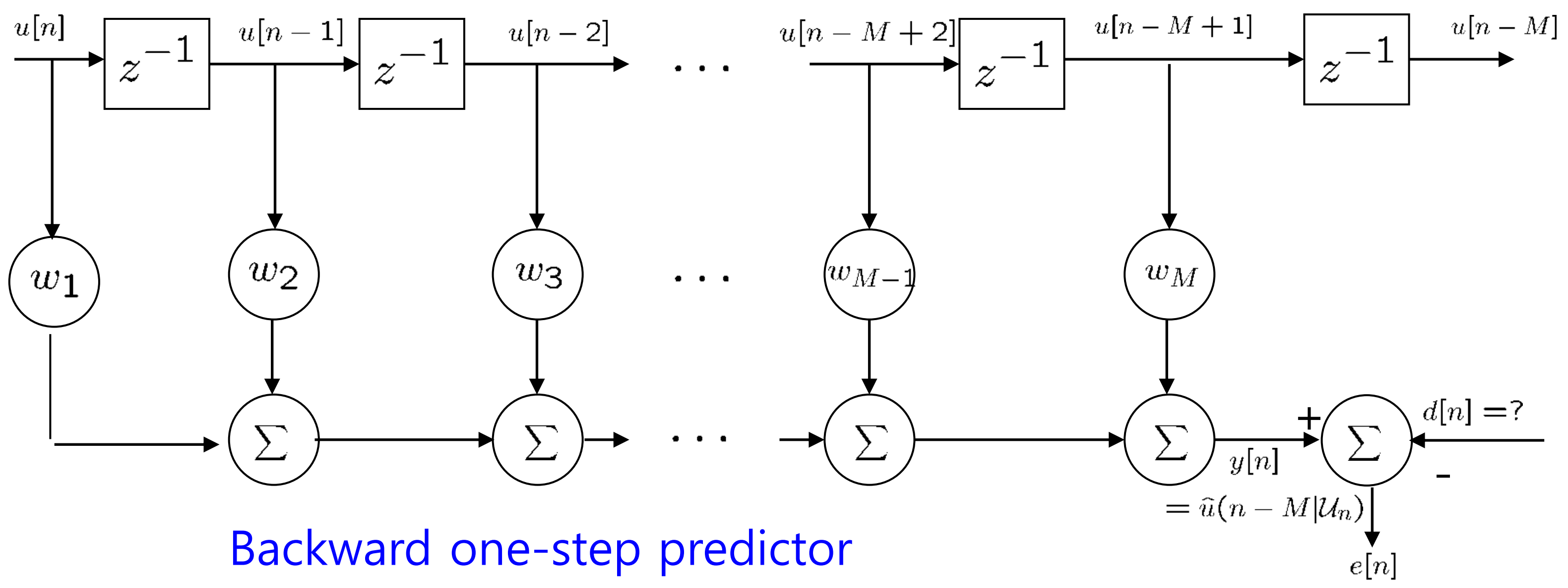
❖ Backward Linear Prediction

For the given time series $u[n], u[n-1], \cdots, u[n-(M-1)] \longrightarrow \hat{u}(n-M|\mathcal{U}_n)$

- How to make a prediction of $u(n-M)$?

Let $\mathcal{U}_n$ denote M-dim. Space spanned by $u[n], u[n-1], \cdots, u[n-M+1]$,

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



Backward one-step predictor

Linear Prediction (9): Backward Linear Prediction (2)

- The desired signal: $d[n] = u[n-M]$
- The prediction error: $b_M(n) = u[n-M] - \hat{u}(n-M|\mathcal{U}_n).$

Let P_M denote the minimum mean-square prediction error,

$$P_M = E\left[|b_M(n)|^2\right] = E\left[b_M(n)b_M^*(n)\right]$$

Herein, $\mathbf{w}_b =$ the optimum tap-weight vector of the backward prediction. To solve the Winer-Hopf equation for $\mathbf{w}_b$, we need

- Correlation matrix
- Cross-correlation matrix
- The variance of $u[n-M] = r(0).$ (Since assumed to zero mean.)

$$\mathbf{R}\mathbf{w}_b = \mathbf{p} \quad \text{or} \quad \mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*}$$

and $$P_M = r(0) - \mathbf{p}^H\mathbf{R}^{-1}\mathbf{p} = r(0) - \mathbf{r}^{BT}\mathbf{w}_b.$$

IVPL Intelligent Vision Processing Lab

Linear Prediction (10): Backward Linear Prediction (3)

- Relationship betw. Backward and Forward predictors

  In terms of $\mathbf{r}$ in Winer-Hopf Eqs,
  - It's elements are arranged in backward.
  - They are complex conjugated.

- Aspect of tap-weights: With $\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*}$,

Complex conjugate

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

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

Since $\mathbf{R}^H = \mathbf{R}$

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

$$\therefore \ \mathbf{w}_b^{B*} = \mathbf{w}_f.$$

$$\therefore \ \mathbf{R}\mathbf{w}_f = \mathbf{r}.$$

Winer-Hopf Eq. of FLP

Linear Prediction (11): Backward Linear Prediction (4)

- Aspect of Ensemble-averaged error power:

With $P_M = r(0) - \mathbf{r}^{BT}\mathbf{w}_b.$ ← reordering

$$P_M = r(0) - \mathbf{r}^{T}\mathbf{w}_b^{B}.$$ ← Complex conjugate

$$P_M = r(0) - \mathbf{r}^{T*}\mathbf{w}_b^{B*},$$
$$= r(0) - \mathbf{r}^{H}\mathbf{w}_b^{B*}.$$

If we compare with that of the FLP case,

Error power of FLP → $P_M = r(0) - \mathbf{r}^{H}\mathbf{w}_f.$

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

$$\therefore\ \mathbf{w}_b^{B*} = \mathbf{w}_f.$$

We may modify a backward predictor into a forward predictor by reversing the sequence in which its tap-weights are positioned and also complex-conjugating them.

Linear Prediction (12): Backward Prediction-Error Filter(1)

- Backward prediction-error filter
  - Output: backward prediction-error (FPE)
  - Backward prediction-error

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

Let $c_{M,k}^*(k=0,1,...,M)$ denote the tap-weights of new filter as the following:

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

Then,

$$\therefore \; b_M(n) = \sum_{k=0}^{M} c_{M,k}^* u[n-k].$$

IVPL
Intelligent Vision Processing Lab

Linear Prediction (13): Backward Prediction-Error Filter(2)

- In aspect of tap-weights of the forward prediction-error filter, we can express as:

$$w^*_{b,M-k+1} = w_{f,k} \text{ or } w_{b,k} = w^*_{f,M-k+1} \quad k = 1, \cdots, M.$$

Then, we can get the following:

$$c_{M,k} = \begin{cases} 1, & k = M, \\ -w^*_{f,M-k}, & k = 0, 1, .., M-1. \end{cases}$$

Therefore, if $c_{M,k} = a^*_{M,M-k} \quad (k = 0, 1, ..., M)$,

$$\boxed{\therefore \; b_M(n) = \sum_{k=0}^{M} \boxed{a_{M,M-k}} u[n-k].}$$

BP filter can be obtained by reversing the sequence in which its tap-weights are positioned and also complex-conjugating them of FP filter.

Linear Prediction (14): Backward Prediction-Error Filter(3)

- Augmented Winer-Hopf Equations for Backward Prediction

Using Eqs. of $P_M = r(0) - \mathbf{r}^H \mathbf{w}_b^{B*}$ and $\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*} (-\mathbf{R}\mathbf{w}_b + \mathbf{r}^{B*} = \mathbf{0})$,

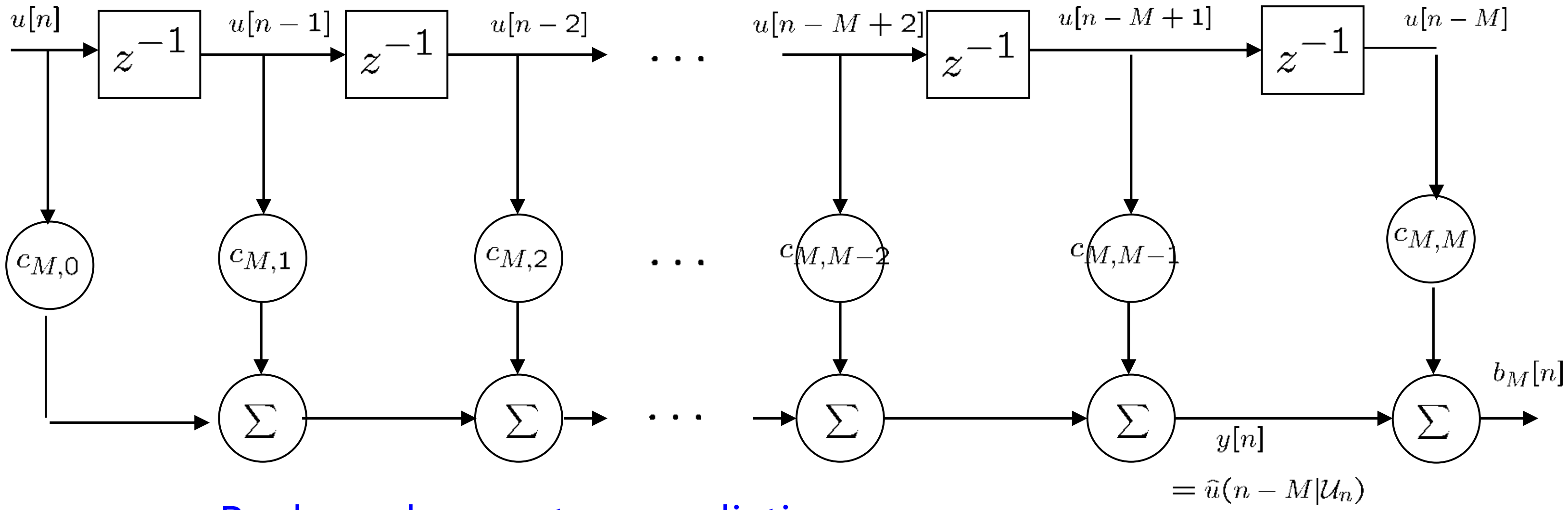
$$\begin{bmatrix} \mathbf{R} & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix}$$

From the Eq., if we let $\begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} = \mathbf{a}_M$ then we can rewrite as a system of (M+1) linear equations:

$$\sum_{l=0}^{M} a_{M,M-l} r[l-i] = \begin{cases} P_M, & i = M, \\ 0, & i = 1, 2, \cdots, M-1. \end{cases}$$

Augmented Winer-Hopf equations for backward prediction-error filter

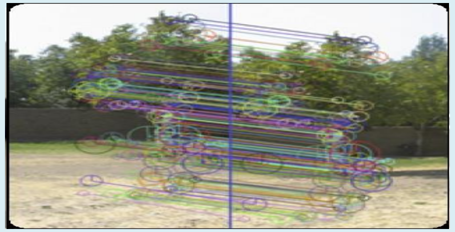
IVPL Intelligent Vision Processing Lab

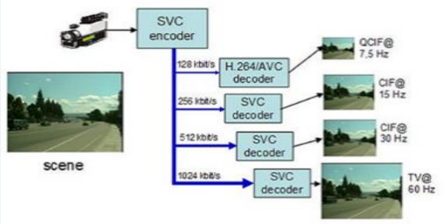


Backward one-step prediction error filter

❖ Levinson-Durbin Algorithm

- Direct method for computing the prediction-error filter coeffs. and error power(P_M) using the augmented Wiener-Hopf equation.
- Recursion in nature.





Contents

- Summary of the Previous Lecture
- Wiener Filter
- Linear Prediction
- **Levinson-Durbin Algorithm**

Linear Prediction (16): Levinson-Durbin Algorithm (1)

❖ Notation

- $\mathbf{a}_m$: tap weight vector of FPE filter with the order of m ((m+1)x1).
- $\mathbf{a}_m^{B*}$: tap weight vector of BPE filter with the order of m ((m+1)x1).
- $\mathbf{a}_{m-1}$: tap weight vector of FPE filter with the order of m-1 (mx1).
- $\mathbf{a}_{m-1}^{B*}$: tap weight vector of BPE filter with the order of m-1 (mx1).

❖ Then Levinson-Durbin recursion may be stated as:

- The ordered update of the tap-weight vector of FPE filter as,

$$a_{m,l} = a_{m-1,l} + \kappa_m a^*_{m-1,m-l} \text{ for } l = 0, 1, \cdots, m.$$

where $a_{m,l}$ is the l-th tap weight of forward prediction-error filter of order m and $a_{m-1,0} = 0,\ a_{m-1,m} = 0$, and κ_m is a constant.

- The ordered update of the tap-weight vector of BPE filter as,

$$a^*_{m,m-l} = a^*_{m-1,m-l} + \kappa^*_m a_{m-1,l} \text{ for } l = 0, 1, \cdots, m.$$

where $a^*_{m,m-l}$ is the l-th tap weight of backward prediction-error filter of order m and others are same.

Linear Prediction (17): Levinson-Durbin Algorithm (2)

❖ Error Power Recursion

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

Reflection coefficient

where $\kappa_m = a_{m,m}.$

- As the order m of the prediction-error filter increases, the corresponding value of the prediction-error power P_M normally decreases or else remains the same.
- For the order of zero $(M = 0)$, $P_0 = r(0).$

❖ Application of the Levinson-Durbin algorithm

- The main goal is to get filter coefficients and prediction-error power.
  - When we have explicit knowledge of the autocorrelation function of the input process: In actual, we can estimate by means of the time-average formula,

$$\hat{r}(k) = \frac{1}{N} \sum_{n=1+k}^{N} u[n]u*[n-k], \quad k = 0, 1, \cdots, M.$$

where N is the total length of the input time series, with N>>M.

Linear Prediction (18): Levinson-Durbin Algorithm (3)

With $\{r(0), r(1), r(2), \cdots, r(M)\}$,

compute $\Delta_{m-1} = \sum_{l=0}^{m-1} r(l-m) a_{m-1,l}$,

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

Recursion: m=0 $\quad P_0 = r(0), \quad \Delta_0 = r^*(1).$

m=M $\quad$ Computation stop.

– When we know the reflection coefficients (κ_m) and the autocorrelation function $r(0)$ for a lag zero. Then we only need the pair of relations:

$$a_{m,l} = a_{m-1,l} + \kappa_m a^*_{m-1,m-l} \quad \text{for } l = 0, 1, \cdots, m.$$

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

Plz, see an Example 2 on p. 260 for illustrating the second method.

- Inverse Levinson-Durbin Algorithm
  - When we need to compute κ_m , given the tap-weights of the filter.
  - Inverse recursion:

With

$$a_{m,l} = a_{m-1,l} + \kappa_m a^*_{m-1,m-l} \quad \text{for } l = 0, 1, \cdots, m.$$

and

$$a^*_{m,m-l} = a^*_{m-1,m-l} + \kappa^*_m a_{m-1,l} \quad \text{for } l = 0, 1, \cdots, m,$$

$$\longrightarrow \begin{bmatrix} 1 & \kappa_m \\ \kappa^*_m & 1 \end{bmatrix} \begin{bmatrix} a_{m-1,l} \\ a^*_{m-1,m-l} \end{bmatrix} = \begin{bmatrix} \boxed{a_{m-1,l}} \\ a^*_{m,m-l} \end{bmatrix}$$

where the order $m = 1, 2, 3, \cdots, M.$

$$a_{m-1,l} = \frac{a_{m-l} - \kappa_m a^*_{m,m-l}}{1 - |\kappa_m|^2}, \quad \longleftarrow \text{Since } \kappa_m = a_{m,m},$$

$$= \frac{a_{m-l} - a_{m,m} a^*_{m,m-l}}{1 - |a_{m,m}|^2}.$$

IVPL Intelligent Vision Processing Lab

We can compute

$$\kappa_m = a_{m,m}.$$

Plz, see an Example 3 on p. 261 for illustrating the second method.

In Next Class?

- We will talk about the Kalman Filter.
  - Introduction to Kalman filter (if possible).

HW#5- Linear Prediction

- Solve the following Problems:
  - (Chap. 6) P. 4, P. 6 , P. 11

- Due date: ~ to the next week.

Thank you for your attention.!!!
QnA

http://ivpl.sookmyung.ac.kr