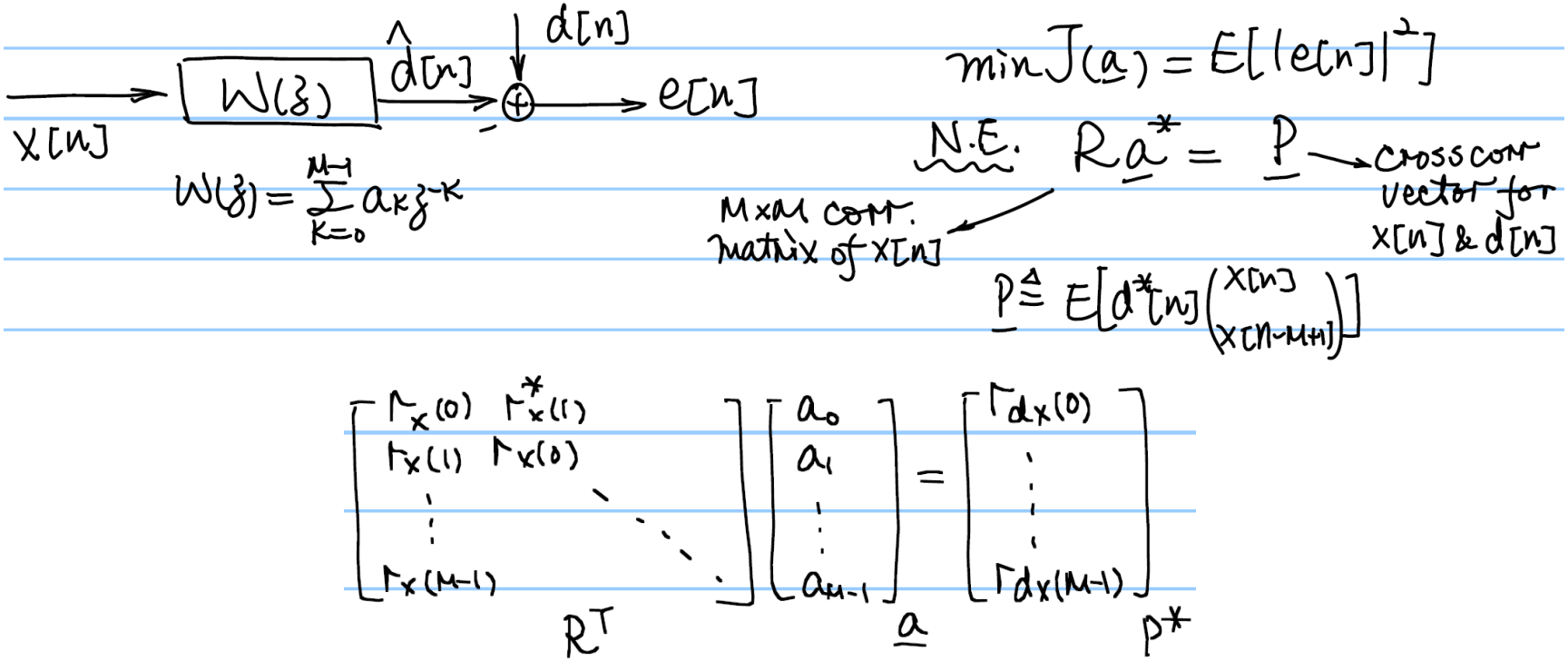
Part-II Parametric Signal Modeling and Linear Prediction Theory
3. Linear Prediction

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 11, 2012.

Review of Last Section: FIR Wiener Filtering



Two perspectives leading to the optimal filter's condition (NE):

1. write $J(\underline{a})$ to have a perfect square
2. $\frac{\partial}{\partial a_k^*} = 0 \Rightarrow$ principle of orthogonality $\mathbb{E}\left[e[n]x^*[n-k]\right] = 0$, $k = 0, \ldots M-1$.

Recap: Principle of Orthogonality

$\mathbb{E}\left[e[n]x^*[n-k]\right] = 0$ for $k = 0, \ldots M-1$.

$\Rightarrow \mathbb{E}\left[d[n]x^*[n-k]\right] = \sum_{\ell=0}^{M-1} a_\ell \cdot \mathbb{E}\left[x[n-\ell]x^*[n-k]\right]$

$\Rightarrow r_{dx}(k) = \sum_{\ell=0}^{M-1} a_\ell r_x(k-\ell) \Rightarrow$ Normal Equation $\underline{p}^* = \mathbf{R}^T \underline{a}$

$J_{min} = \mathrm{Var}(d[n]) - \mathrm{Var}(\hat{d}[n])$

where $\mathrm{Var}(\hat{d}[n]) = \mathbb{E}\left[\hat{d}[n]\hat{d}^*[n]\right] = \mathbb{E}\left[\underline{a}^T \underline{x}[n]\underline{x}^H[n]\underline{a}^*\right] = \underline{a}^T \mathbf{R}_x \underline{a}^*$

bring in N.E. for $\underline{a} \Rightarrow \mathrm{Var}(\hat{d}[n]) = \underline{a}^T \underline{p} = \underline{p}^H \mathbf{R}^{-1} \underline{p}$

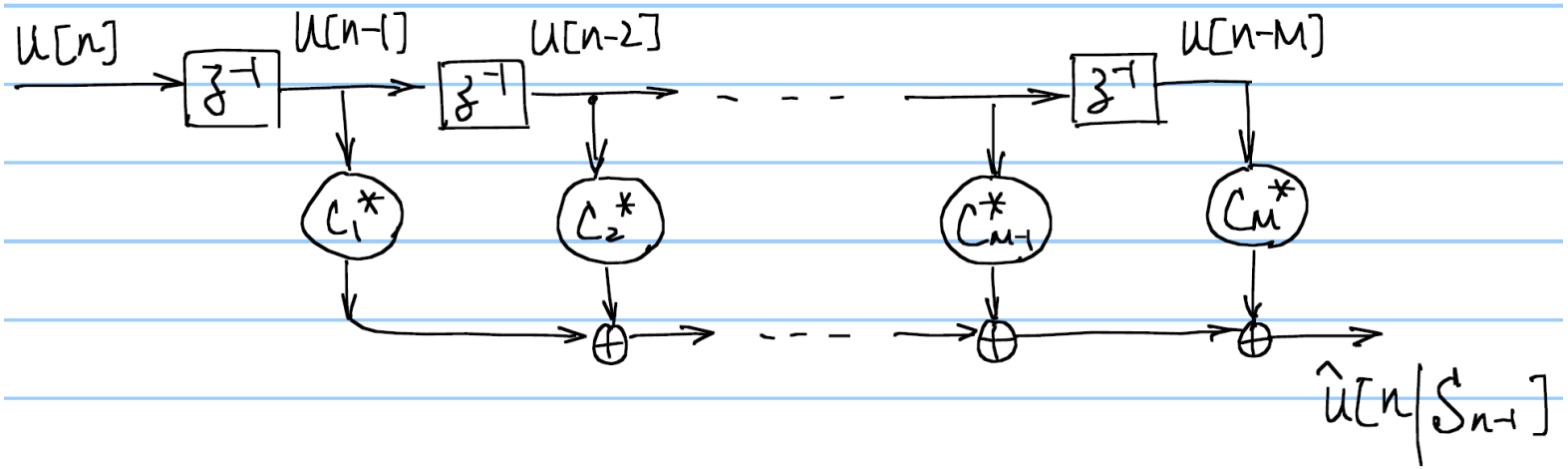
May also use the vector form to derive N.E.: set gradient $\bigtriangledown_{\underline{a}^*} J = 0$

Forward Linear Prediction

Recall last section: FIR Wiener filter $W(z) = \sum_{k=0}^{M-1} a_k z^{-k}$

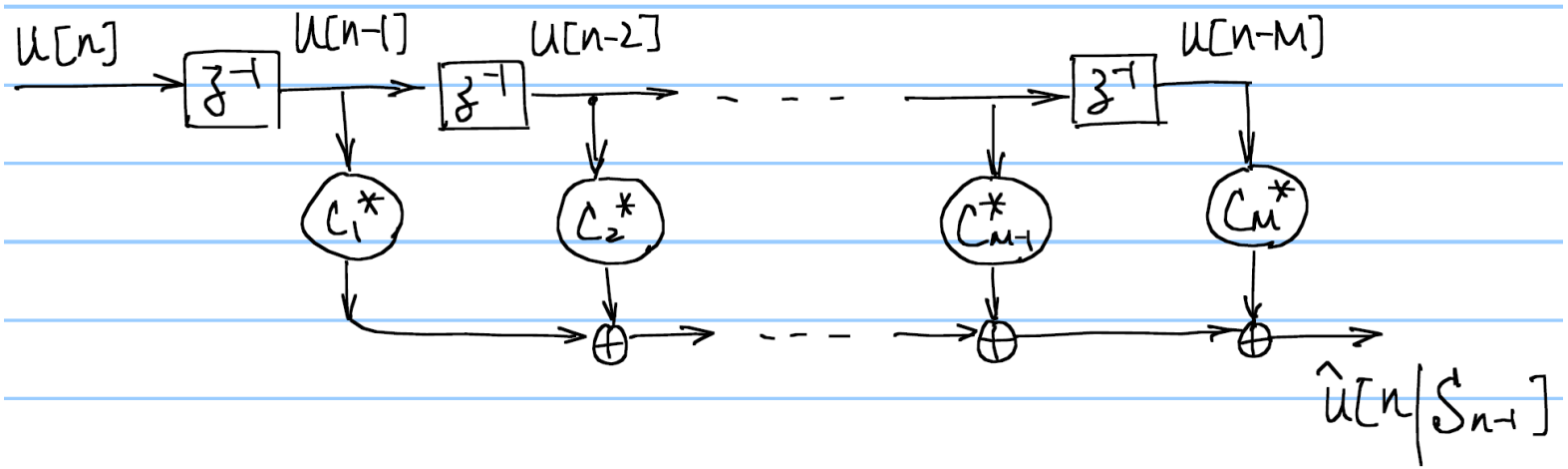
Let $c_k \triangleq a_k^*$ (i.e., c_k^* represents the filter coefficients and helps us to avoid many conjugates in the normal equation)

Given $u[n-1], u[n-2], \ldots, u[n-M]$, we are interested in estimating $u[n]$ with a linear predictor:



This structure is called "tapped delay line": individual outputs of each delay are tapped out and diverted into the multipliers of the filter/predictor.

Forward Linear Prediction



$$\hat{u}\,[n|\mathbb{S}_{n-1}] = \sum_{k=1}^{M} c_k^* u[n-k] = \underline{c}^H \underline{u}[n-1]$$

$\mathbb{S}_{n-1}$ denotes the M-dimensional space spanned by the samples $u[n-1], \ldots, u[n-M]$, and

$$\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{bmatrix}, \quad \underline{u}[n-1] = \begin{bmatrix} u[n-1] \\ u[n-2] \\ \vdots \\ u[n-M] \end{bmatrix}$$

$\underline{u}[n-1]$ is vector form for tap inputs and is $\underline{x}[n]$ from General Wiener

Forward Prediction Error

- The forward prediction error

  $f_M[n] = u[n] - \hat{u}\,[n|\mathbb{S}_{n-1}]$

  $e[n]$ $d[n]$ $\leftarrow$ From general Wiener filter notation

- The minimum mean-squared prediction error

  $P_M = \mathbb{E}\left[|f_M[n]|^2\right]$

Readings for LP: Haykin 4th Ed. 3.1-3.3

Optimal Weight Vector

To obtain optimal weight vector $\underline{c}$, apply Wiener filtering theory:

1. Obtain the correlation matrix:

$$\mathbf{R} = \mathbb{E}\left[\underline{u}[n-1]\underline{u}^H[n-1]\right]$$
$$= \mathbb{E}\left[\underline{u}[n]\underline{u}^H[n]\right] \text{ (by stationarity)}$$

where $\underline{u}[n] = \begin{bmatrix} u[n] \\ u[n-1] \\ \vdots \\ u[n-M+1] \end{bmatrix}$

2. Obtain the "cross correlation" vector between the tap inputs and the desired output $d[n] = u[n]$:

$$\mathbb{E}\left[\underline{u}[n-1]u^*[n]\right] = \begin{bmatrix} r(-1) \\ r(-2) \\ \vdots \\ r(-M) \end{bmatrix} \triangleq \underline{r}$$

Optimal Weight Vector

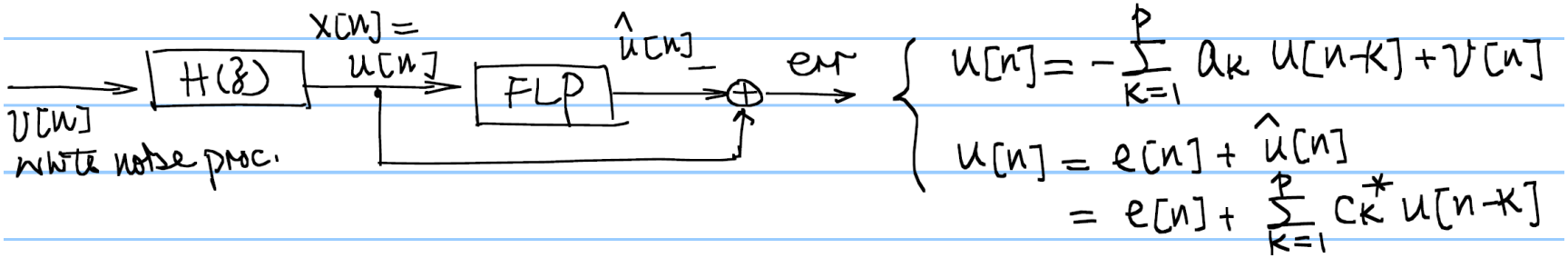
3. Thus the Normal Equation for FLP is

$$\mathbf{R}\underline{c} = \underline{r}$$

The prediction error is

$$P_M = r(0) - \underline{r}^H \underline{c}$$

Relation: N.E. for FLP vs. Yule-Walker eq. for AR



The Normal Equation for FLP is $\mathbf{R}\underline{c} = \underline{r}$

Yule-Walker Eq.

$$\begin{bmatrix} r(0) & r(-1) & \cdots & r(-p+1) \\ r(1) & & & \\ \vdots & & \ddots & \vdots \\ r(p-1) & & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a(1) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} -r_x(1) \\ \vdots \\ -r_x(p) \end{bmatrix}$$

Recall the $\underline{a}$ in AR process $\quad x[n] = -\sum_{k=1}^{p} x[n-k]a[k] + v[n]$

$$\Rightarrow R^T \underline{a} = -\begin{bmatrix} r_x(1) \\ \vdots \\ r_x(M) \end{bmatrix} \Longleftrightarrow \text{conjugate} \quad R^H(-\underline{a}^*) = \begin{bmatrix} r_x^*(1) \\ \vdots \\ r_x^*(M) \end{bmatrix} = \begin{bmatrix} r(-1) \\ \vdots \\ r(-M) \end{bmatrix}$$

for AR(M); R = our $\underline{c}$; $= r$

$\Rightarrow$ N.E. is in the same form as the Yule-Walker equation for AR

Relation: N.E. for FLP vs. Yule-Walker eq. for AR

If the forward linear prediction is applied to an AR process of known model order M and optimized in MSE sense, its tap weights in theory take on the same values as the corresponding parameter of the AR process.

- Not surprising: the equation defining the forward prediction and the difference equation defining the AR process have the same mathematical form.
- When $u[n]$ process is not AR, the predictor provides only an approximation of the process.

  $\Rightarrow$ This provide a way to test if u[n] is an AR process (through examining the whiteness of prediction error $e[n]$); and if so, determine its order and AR parameters.

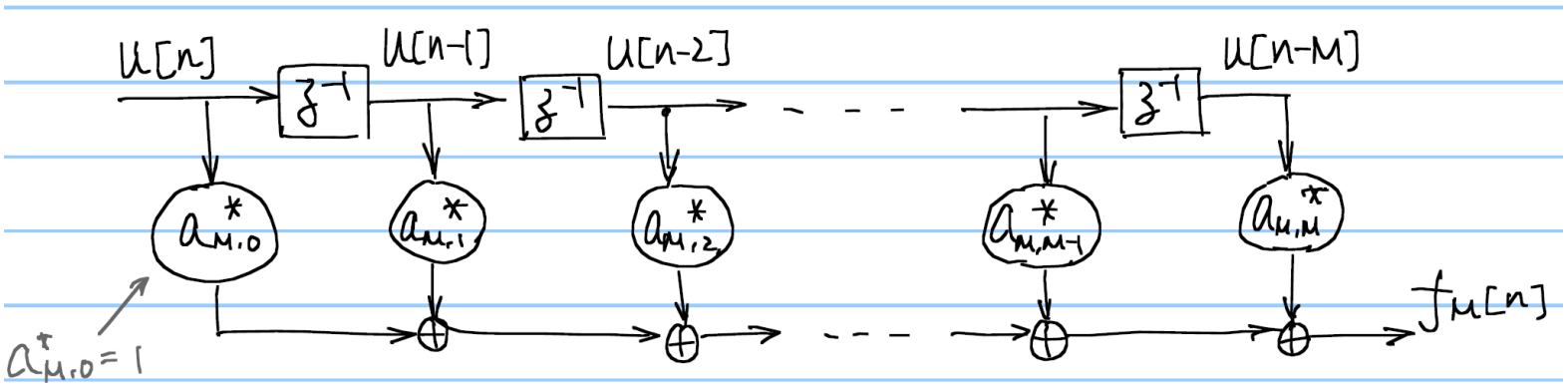
Question: Optimal predictor for $\{u[n]\}$=AR(p) when $p < M$?

Forward-Prediction-Error Filter

$$f_M[n] = u[n] - \underline{c}^H \underline{u}[n-1]$$

$$\text{Let } a_{M,k} = \begin{cases} 1 & k = 0 \\ -c_k & k = 1, 2, \ldots, M \end{cases}, \text{ i.e., } \underline{a}_M \triangleq \begin{bmatrix} a_{M,0} \\ \vdots \\ a_{M,M} \end{bmatrix}$$

$$\Rightarrow f_M[n] = \sum_{k=0}^{M} a_{M,k}^* u[n-k] = \underline{a}_M^H \begin{bmatrix} \underline{u}[n] \\ u[n-M] \end{bmatrix}$$



Augmented Normal Equation for FLP

From the above results:

$$\begin{cases} \mathbf{R}\underline{c} = \underline{r} & \text{Normal Equation or Wiener-Hopf Equation} \\ P_M = r(0) - \underline{r}^H \underline{c} & \text{prediction error} \end{cases}$$

Put together:

$$\underbrace{\begin{bmatrix} r(0) & \underline{r}^H \\ \underline{r} & \mathbf{R}_M \end{bmatrix}}_{\mathbf{R}_{M+1}} \begin{bmatrix} 1 \\ -\underline{c} \end{bmatrix} = \begin{bmatrix} P_M \\ \underline{0} \end{bmatrix}$$

Augmented N.E. for FLP

$$\mathbf{R}_{M+1}\underline{a}_M = \begin{bmatrix} P_M \\ \underline{0} \end{bmatrix}$$

Summary of Forward Linear Prediction

	General Wiener	Forward LP	Backward LP
Tap input			
Desired response			
(conj) Weight vector			
Estimated sig			
Estimation error			
Correlation matrix			
Cross-corr vector			
MMSE			
Normal Equation			
Augmented N.E.			

(detail)

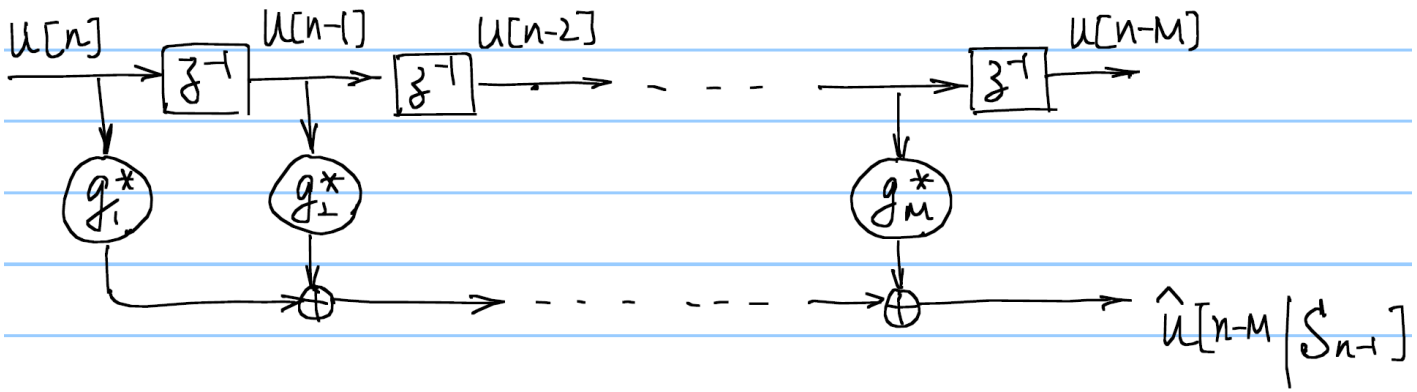
Backward Linear Prediction

Given $u[n], u[n-1], \ldots, u[n-M+1]$, we are interested in estimating $u[n-M]$.

Backward prediction error $b_M[n] = u[n-M] - \hat{u}\,[n-M|\mathbb{S}_n]$

- $\mathbb{S}_n$: span $\{u[n], u[n-1], \ldots, u[n-M+1]\}$

Minimize mean-square prediction error $P_{M,\text{BLP}} = \mathbb{E}\left[|b_M[n]|^2\right]$



Backward Linear Prediction

Let g denote the optimal weight vector (conjugate) of the BLP: i.e., $\hat{u}[n-M] = \sum_{k=1}^{M} g_k^* u[n+1-k]$.

To solve for $\underline{g}$, we need

1. Correlation matrix $\mathbf{R} = \mathbb{E}\left[\underline{u}[n]\underline{u}^H[n]\right]$
2. Crosscorrelation vector

$$\mathbb{E}\left[\underline{u}[n]u^*[n-M]\right] = \begin{bmatrix} r(M) \\ r(M-1) \\ \vdots \\ r(1) \end{bmatrix} \triangleq \underline{r}^{B*}$$

Normal Equation for BLP

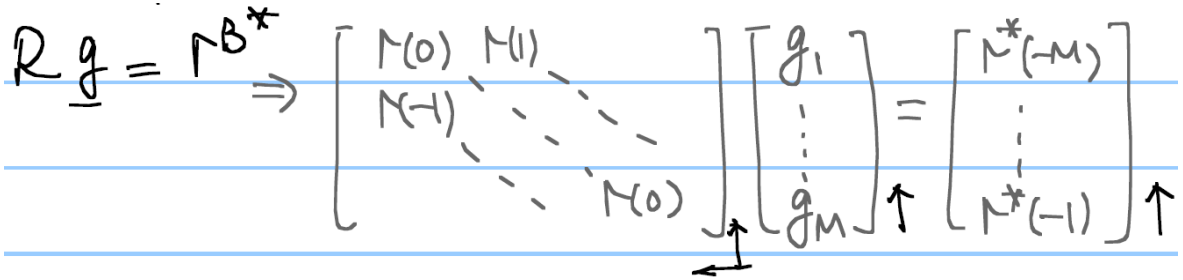
$$\mathbf{R}\underline{g} = \underline{r}^{B*}$$

The BLP prediction error: $P_{M,\text{BLP}} = r(0) - (\underline{r}^B)^T \underline{g}$

Relations between FLP and BLP

Recall the NE for FLP: $\mathbf{R}\underline{c} = \underline{r}$

Rearrange the NE for BLP backward: $\Rightarrow \mathbf{R}^T \underline{g}^B = \underline{r}^*$



$$\text{Conjugate} \Rightarrow \mathbf{R}^H \underline{g}^{B^*} = \underline{r} \Rightarrow \mathbf{R}\underline{g}^{B^*} = \underline{r}$$

$\therefore$ optimal predictors of FLP: $\underline{c} = \underline{g}^{B^*}$, or equivalently $\underline{g} = \underline{c}^{B^*}$

By reversing the order & complex conjugating $\underline{c}$, we obtain $\underline{g}$.

Relations between FLP and BLP

$$P_{M,\text{BLP}} = r(0) - (\underline{r}^B)^T \underline{g} = r(0) - (\underline{r}^B)^T \underline{c}^{B^*} = r(0) - \underbrace{\left[\underline{r}^H \underline{c}\right]}_{\text{real, scalar}}^{B*}$$

$$= r(0) - \underline{r}^H \underline{c} = P_{M,\text{FLP}}$$

This relation is not surprising:
the process is w.s.s. (s.t. $r(k) = r^*(-k)$), and the optimal prediction error depends only on the process' statistical property.

⁂ Recall from Wiener filtering: $J_{\min} = \sigma_d^2 - \underline{p}^H \mathbf{R}^{-1} \underline{p}$

(FLP) $\underline{r}^H \mathbf{R}^{-1} \underline{r}$

(BLP) $\underline{r}^{B*H} \mathbf{R}^{-1} \underline{r}^{B^*} = (\underline{r}^H \mathbf{R}^{T*^{-1}} \underline{r})^{B^*} = \underline{r}^H \mathbf{R}^{-1} \underline{r}$

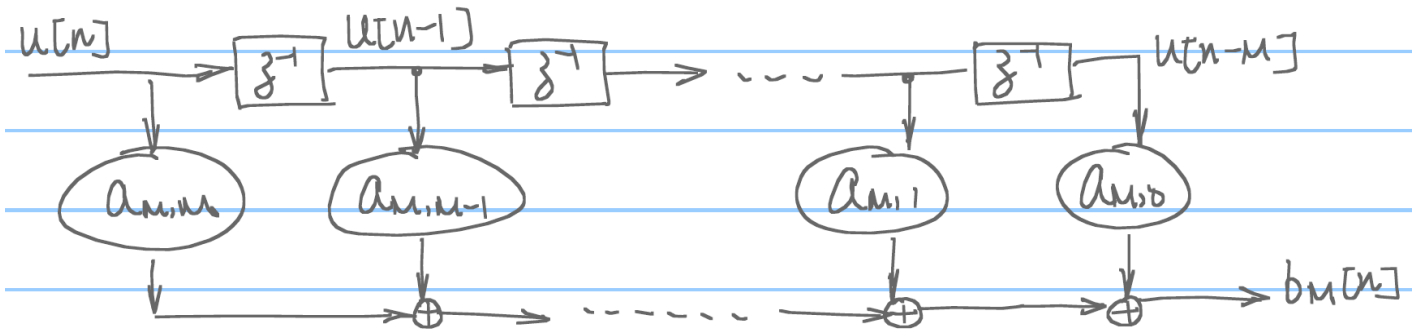
Backward-Prediction-Error Filter

$$b_M[n] = u[n-M] - \sum_{k=1}^{M} g_k^* u[n+1-k]$$

Using the $a_{i,j}$ notation defined earlier and $g_k = -a_{M,M+1-k}^*$:

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

$$= \underline{a}_M^{BT} \begin{bmatrix} \underline{u}[n] \\ u[n-M] \end{bmatrix}, \text{ where } \underline{a}_M = \begin{bmatrix} a_{M,0} \\ \vdots \\ a_{M,M} \end{bmatrix}$$



Augmented Normal Equation for BLP

$$\text{Bring together } \begin{cases} \mathbf{R}\underline{g} = \underline{r}^{B*} \\ P_M = r(0) - (\underline{r}^B)^T \underline{g} \end{cases}$$

$$\Rightarrow \underbrace{\begin{bmatrix} \mathbf{R} & \underline{r}^{B*} \\ (\underline{r}^B)^T & r(0) \end{bmatrix}}_{\mathbf{R}_{M+1}} \begin{bmatrix} -\underline{g} \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ P_M \end{bmatrix}$$

Augmented N.E. for BLP

$$\mathbf{R}_{M+1}\underline{a}_M^{B*} = \begin{bmatrix} \underline{0} \\ P_M \end{bmatrix}$$

Summary of Backward Linear Prediction

	General Wiener	Forward LP	Backward LP
Tap input			
Desired response			
(conj) Weight vector			
Estimated sig			
Estimation error			
Correlation matrix			
Cross-corr vector			
MMSE			
Normal Equation			
Augmented N.E.			

(detail)

Whitening Property of Linear Prediction

(Ref: Haykin 4th Ed. §3.4 (5) Property)

Conceptually: The best predictor tries to explore the predictable traces from a set of (past) given values onto the future value, leaving only the unforeseeable parts as the prediction error.

Also recall the principle of orthogonality: the prediction error is statistically uncorrelated with the samples used in the prediction.

As we increase the order of the prediction-error filter,
the correlation between its adjacent outputs is reduced.
If the order is high enough, the output errors become approximately a white process (i.e., be "whitened").

Analysis and Synthesis

From forward prediction results on the $\{u[n]\}$ process:

$$\begin{cases} u[n] + a^*_{M,1} u[n-1] + \ldots + a^*_{M,M} u[n-M] = f_M[n] & \text{Analysis} \\ \hat{u}[n] = -a^*_{M,1} u[n-1] - \ldots - a^*_{M,M} u[n-M] + v[n] & \text{Synthesis} \end{cases}$$

Here $v[n]$ may be quantized version of $f_M[n]$, or regenerated from white noise

If $\{u[n]\}$ sequence has high correlation between adjacent samples, then $f_M[n]$ will have a much smaller dynamic range than $u[n]$.

Compression tool #3: Predictive Coding

Recall two compression tools from Part-1:
(1) lossless: decimate a bandlimited signal; (2) lossy: quantization.

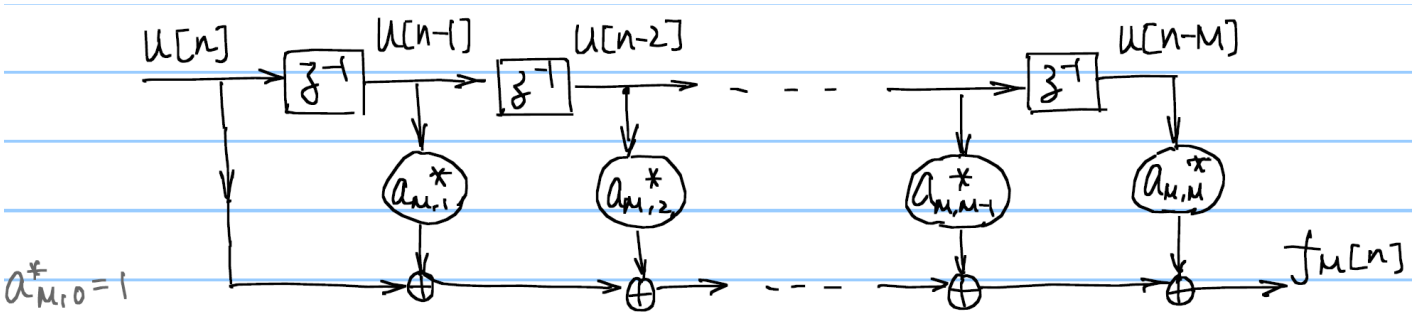
Tool #3: Linear Prediction. we can first figure out the best predictor for a chunk of approximately stationary samples, encode the first sample, then do prediction and encode the prediction residues (as well as the prediction parameters).

The structures of analysis and synthesis of linear prediction form a matched pair.

This is the basic principle behind Linear Prediction Coding (LPC) for transmission and reconstruction of digital speech signals.

Linear Prediction: Analysis

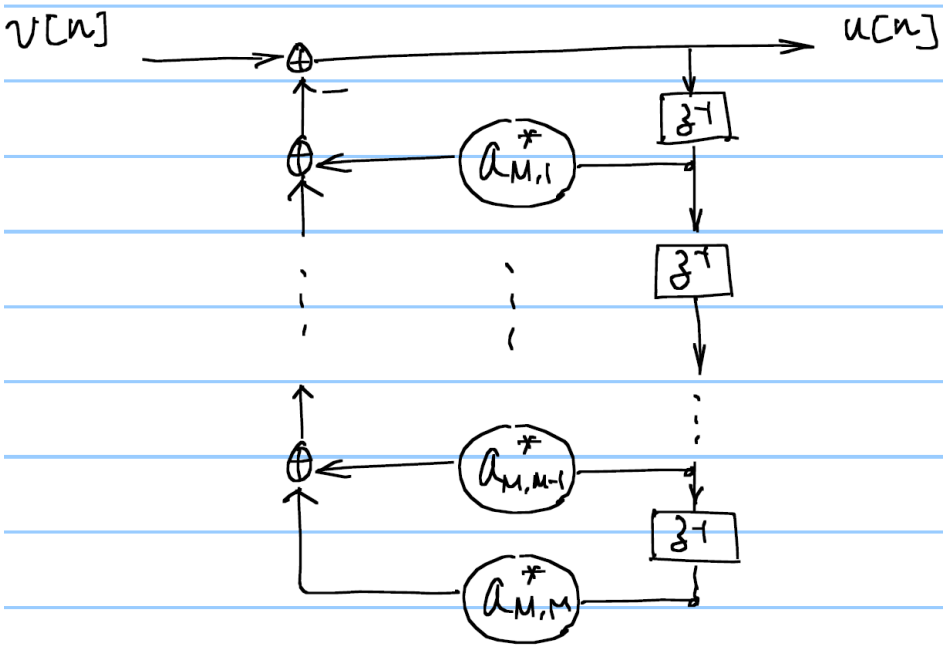
$$u[n] + a^*_{M,1} u[n-1] + \ldots + a^*_{M,M} u[n-M] = f_M[n]$$



If $\{f_M[n]\}$ is white (i.e., the correlation among $\{u[n], u[n-1], \ldots\}$ values have been completely explored), then the process $\{u[n]\}$ can be statistically characterized by $\underline{a}_M$ vector, plus the mean and variance of $f_M[n]$.

Linear Prediction: Synthesis

$$\hat{u}[n] = -a^*_{M,1}u[n-1] - \ldots - a^*_{M,M}u[n-M] + v[n]$$



If $\{v[n]\}$ is a white noise process, the synthesis output $\{u[n]\}$ using linear prediction is an AR process with parameters $\{a_{M,k}\}$.

LPC Encoding of Speech Signals

- Partition speech signal into frames s.t. within a frame it is approximately stationary
- Analyze a frame to obtain a compact representation of the linear prediction parameters, and some parameters characterizing the prediction residue $f_M[n]$

  (if more b.w. is available and higher quality is desirable, we may also include some coarse representation of $f_M[n]$ by quantization)
- This gives much more compact representation than simple digitization (PCM coding): e.g., 64kbps $\rightarrow$ 2.4k-4.8kbps
- A decoder uses the synthesis structure to reconstruct the speech signal, with a suitable driving sequence
  (periodic impulse train for voiced sound & white noise for fricative sound; quantized $f_M[n]$ if b.w. allowed)

Review: Recursive Relation of Correlation Matrix

$$R_{M+1} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r^*(1) & r(0) & \cdots & r(M-1) \\ r^*(2) & r^*(1) & \ddots & \vdots \\ r^*(M) & r^*(M-1) & & r(0) \end{bmatrix} = \left[\begin{array}{c|c} r(0) & \underline{r}^H \\ \hline \underline{r} & R_M \end{array}\right] = \left[\begin{array}{c|c} R_M & (\underline{r}^B)^* \\ \hline (\underline{r}^B)^T & r(0) \end{array}\right]$$

where $\underline{r} = \begin{bmatrix} r^*(1) \\ \vdots \\ r^*(M) \end{bmatrix} = \begin{bmatrix} r(-1) \\ \vdots \\ r(-M) \end{bmatrix}$, $\underline{r}^B = \begin{bmatrix} r^*(M) \\ \vdots \\ r^*(1) \end{bmatrix}$

Matrix Inversion Lemma for Homework

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Special case: $(A + \underset{B}{\underline{u}}\,\underline{v}^H)^{-1} = A^{-1} - \dfrac{A^{-1}\underline{u}\,\underline{v}^H A^{-1}}{1 + \underline{v}^H A^{-1}\underline{u}}$

$$(I + \underline{u}\,\underline{v}^H)^{-1} = I - \frac{\underline{u}\,\underline{v}^H}{1 + \underline{v}^H\underline{u}}$$