Parametric Signal Modeling and Linear Prediction Theory
1. Discrete-time Stochastic Processes

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: October 27, 2011.

Outline of Part-2

1. Discrete-time Stochastic Processes
2. Discrete Wiener Filtering
3. Linear Prediction

Outline of Section 1

- Basic Properties and Characterization
  - 1st and 2nd moment function; ergodicity
  - correlation matrix; power-spectrum density
- The Rational Transfer Function Model
  - ARMA, AR, MA processes
  - Wold Decomposition Theorem
  - ARMA, AR, and MA models and properties
  - asymptotic stationarity of AR process

Readings for §1.1: Haykin 4th Ed. 1.1-1.3, 1.12, 1.14;
see also Hayes 3.3, 3.4, and background reviews 2.2, 2.3, 3.2

Stochastic Processes

- To describe the time evolution of a statistical phenomenon according to probabilistic laws.

  Example random processes: speech signals, image, noise, temperature and other spatial/temporal measurements, etc.

- Discrete-time Stochastic Process $\{u[n]\}$
  - Focus on the stochastic process that is defined / observed at discrete and uniformly spaced instants of time
  - View it as an ordered sequence of random variables that are related in some statistical way:
    $\{\ldots u[n-M], \ldots, u[n], u[n+1], \ldots\}$
  - A random process is not just a single function of time; it may have an **infinite** number of different realizations

Parametric Signal Modeling

- A general way to completely characterize a random process is by joint probability density functions for all possible subsets of the r.v. in it:

$$\text{Probability of } \{u[n_1], u[n_2], \dots, u[n_k]\}$$

- **Question:** How to use only a few parameters to describe a process?

  Determine a model and then the model parameters

  $\Rightarrow$ This part of the course studies the signal modeling (including models, applicable conditions, how to determine the parameters, etc)

(1) Partial Characterization by 1st and 2nd moments

It is often difficult to determine and efficiently describe the joint p.d.f. for a general random process.

As a compromise, we consider partial characterization of the process by specifying its 1st and 2nd moments.

Consider a stochastic time series $\{u[n]\}$, where $u[n], u[n-1], \ldots$ may be complex valued. We define the following functions:

- **mean-value function**: $m[n] = \mathbb{E}\left[u[n]\right]$, $n \in \mathbb{Z}$
- **autocorrelation function**: $r(n, n-k) = \mathbb{E}\left[u[n]u^*[n-k]\right]$
- **autocovariance function**: $c(n, n-k) = \mathbb{E}\left[(u[n] - m[n])(u[n-k] - m[n-k])^*\right]$

Without loss of generality, we often consider zero-men random process $\mathbb{E}\left[u[n]\right] = 0 \; \forall n$, since we can always subtract the mean in preprocessing.

Now the autocorrelation and autocovariance functions become identical.

Wide-Sense Stationary (w.s.s.)

Wide-Sense Stationarity

If $\forall n$, $m[n] = m$ and $r(n, n-k) = r(k)$ (or $c(n, n-k) = c(k)$), then the sequence $u[n]$ is said to be wide-sense stationary (w.s.s.), or also called stationary to the second order.

- The strict stationarity requires the entire statistical property (characterized by joint probability density or mass function) to be invariant to time shifts.
- The partial characterization using 1st and 2nd moments offers two important advantages:
  1. reflect practical measurements;
  2. well suited for linear operations of random processes

(2) Ensemble Average vs. Time Average

- Statistical expectation $\mathbb{E}(\cdot)$ as an ensemble average: take average across (different realizations of) the process
- Time-average: take average along the process.

  This is what we can rather easily measure from one realization of the random process.

Question: Are these two average the same?

Answer: No in general. (Examples/discussions from ENEE620.)

Consider two special cases of correlations between signal samples:

1. $u[n], u[n-1], \cdots$ i.i.d.
2. $u[n] = u[n-1] = \cdots$ (i.e. all samples are exact copies)

Mean Ergodicity

For a w.s.s. process, we may use the time average

$$\hat{m}(N) = \frac{1}{N} \sum_{n=0}^{N-1} u[n]$$

to estimate the mean m.

- $\hat{m}(N)$ is an unbiased estimator of the mean of the process.

$\because \mathbb{E}\left[\hat{m}(N)\right] = m \ \ \forall N$.

- **Question:** How much does $\hat{m}(N)$ from one observation deviate from the true mean?

Mean Ergodic

A w.s.s. process $\{u[n]\}$ is mean ergodic in the mean square error sense if $\lim_{N\to\infty} \mathbb{E}\left[|m - \hat{m}(N)|^2\right] = 0$

Mean Ergodicity

A w.s.s. process $\{u[n]\}$ is mean ergodic in the mean square error sense if $\lim_{N\to\infty} \mathbb{E}\left[|m - \hat{m}(N)|^2\right] = 0$

Question: under what condition will this be satisfied?

(Details)

$\Rightarrow$ (nece.& suff.) $\lim_{N\to\infty} \frac{1}{N} \sum_{\ell=-N+1}^{N-1} (1 - \frac{|\ell|}{N}) c(\ell) = 0$

Mean ergodicity suggests that $c(\ell)$ is asymptotically decaying s.t. $\{u[n]\}$ is asymptotically uncorrelated.

Correlation Ergodicity

Similarly, let the autocorrelation estimator be

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

The w.s.s. process $\{u[n]\}$ is said to be correlation ergodic in the MSE sense if the mean squared difference between $r(k)$ and $\hat{r}(k, N)$ approaches zero as $N \to \infty$.

(3) Correlation Matrix

Given an observation vector $\underline{u}[n]$ of a w.s.s. process, the correlation matrix $\mathbf{R}$ is defined as $\mathbf{R} \triangleq \mathbb{E}\left[\underline{u}[n]\underline{u}^H[n]\right]$

where H denotes Hermitian transposition (i.e., conjugate transpose).

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

Each entry in $\mathbf{R}$ is

$$[\mathbf{R}]_{i,j} = \mathbb{E}\left[u[n-i]u^*[n-j]\right] = r(j-i)$$
$$(0 \leq i,j \leq M-1)$$

$$\text{Thus } \mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & \cdots & r(M-1) \\ r(-1) & r(0) & r(1) & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ r(-M+2) & \cdots & \cdots & r(0) & r(1) \\ r(-M+1) & \cdots & \cdots & \cdots & r(0) \end{bmatrix}$$

Properties of $\mathbf{R}$

1. $\mathbf{R}$ is Hermitian, i.e., $\mathbf{R}^H = \mathbf{R}$

   Proof (Details)

2. $\mathbf{R}$ is Toeplitz.

   A matrix is said to be Toeplitz if all elements in the main diagonal are identical, and the elements in any other diagonal parallel to the main diagonal are identical.

   $\mathbf{R}$ Toeplitz $\Leftrightarrow$ the w.s.s. property.

Properties of **R**

3. **R** is non-negative definite , i.e., $\underline{x}^H \mathbf{R} \underline{x} \geq 0, \forall \underline{x}$

Proof (Details)

- eigenvalues of a Hermitian matrix are real.

(similar relation in FT: real in one domain $\sim$ conjugate symmetric in the other)

- eigenvalues of a non-negative definite matrix are non-negative.

Proof (Details)

Properties of **R**

4. $\underline{u}^B[n] \triangleq \begin{bmatrix} u[n-M+1] \\ \vdots \\ u[n-1] \\ u[n] \end{bmatrix}$, i.e., reversely ordering $\underline{u}[n]$,

then the corresponding correlation matrix becomes

$$\mathbb{E}\left[\underline{u}^B[n](\underline{u}^B[n])^H\right] = \begin{bmatrix} r(0) & r(-1) & \cdots & r(-M+1) \\ r(1) & r(0) & & \vdots \\ \vdots & & \ddots & \vdots \\ r(M-1) & \cdots & \cdots & r(0) \end{bmatrix} = \mathbf{R}^T$$

Properties of **R**

5. Recursive relations: correlation matrix for $(M+1) \times 1$ $\underline{u}[n]$:

(Details)

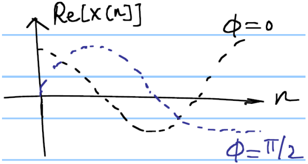
$$R_{M+1} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r^*(1) & r(0) & \cdots & r(M-1) \\ r^*(2) & r^*(1) & \ddots & \vdots \\ \vdots & \vdots & & \\ r^*(M) & r^*(M-1) & & r(0) \end{bmatrix} \quad \text{and} \quad \underline{u}_{M+1}[n] = \begin{bmatrix} \underline{u}_M[n] \\ \hline u(n-M) \end{bmatrix} = \begin{bmatrix} u[n] \\ \hline \underline{u}_M[n-1] \end{bmatrix}$$

$$= \begin{bmatrix} r(0) & \underline{r}^H \\ \underline{r} & R_M \end{bmatrix} = \begin{bmatrix} R_M & (\underline{r}^B)^* \\ (\underline{r}^B)^T & r(0) \end{bmatrix}$$

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

(4) Example-1: Complex Sinusoidal Signal

$x[n] = A\exp\left[j(2\pi f_0 n + \phi)\right]$ where A and f_0 are real constant, $\phi \sim$ uniform distribution over $[0, 2\pi)$ (i.e., random phase)



$\mathbb{E}\left[x[n]\right] = ?$

$\mathbb{E}\left[x[n]x^*[n-k]\right] = ?$

Is $x[n]$ is w.s.s.?

Example-2: Complex Sinusoidal Signal with Noise

Let $y[n] = x[n] + w[n]$ where $w[n]$ is white Gaussian noise uncorrelated to $x[n]$, $w[n] \sim N(0, \sigma^2)$

Note: for white noise, $\mathbb{E}\left[w[n]w^*[n-k]\right] = \begin{cases} \sigma^2 & k = 0 \\ 0 & o.w. \end{cases}$

$r_y(k) = \mathbb{E}\left[y[n]y^*[n-k]\right] =?$

$\mathbf{R}_y =?$

Rank of Correlation Matrices $\mathbf{R}_x$, $\mathbf{R}_w$, $\mathbf{R}_y =?$

(5) Power Spectral Density (a.k.a. Power Spectrum)

Power spectral density (p.s.d.) of a w.s.s. process $\{x[n]\}$

$$P_X(\omega) \triangleq \text{DTFT}[r_x(k)] = \sum_{k=-\infty}^{\infty} r_x(k) e^{-j\omega k}$$

$$r_x(k) \triangleq \text{DTFT}^{-1}[PX(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_X(\omega) e^{j\omega k} d\omega$$

The p.s.d. provides frequency domain description of the 2nd-order moment of the process (may also be defined as a function of f: $\omega = 2\pi f$)

The power spectrum in terms of ZT:

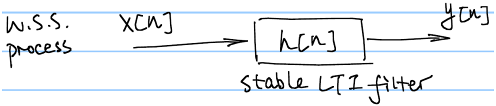
$$P_X(z) = \text{ZT}[r_x(k)] = \sum_{k=-\infty}^{\infty} r_x(k) z^{-k}$$

Physical meaning of p.s.d.: describes how the signal power of a random process is distributed as a function of frequency.

Properties of Power Spectral Density

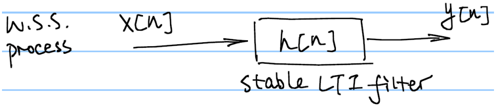
- $r_x(k)$ is conjugate symmetric: $r_x(k) = r_x^*(-k)$
  $\Leftrightarrow P_X(\omega)$ is real valued: $P_X(\omega) = P_X^*(\omega)$; $P_X(z) = P_X^*(1/z^*)$
- For real-valued random process: $r_x(k)$ is real-valued and even symmetric
  $\Rightarrow P_X(\omega)$ is real and even symmetric, i.e.,
  $P_X(\omega) = P_X(-\omega)$; $P_X(z) = P_X^*(z^*)$
- For w.s.s. process, $P_X(\omega) \geq 0$ (nonnegative)
- The power of a zero-mean w.s.s. random process is proportional to the area under the p.s.d. curve over one period 2π,
  i.e., $\mathbb{E}\left[|x[n]|^2\right] = r_x(0) = \frac{1}{2\pi}\int_0^{2\pi} P_X(\omega)d\omega$
  Proof: note $r_x(0) =$ IDTFT of $P_X(\omega)$ at $k = 0$

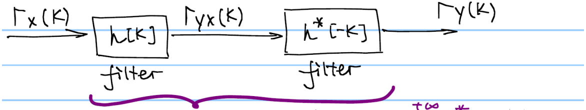
(6) Filtering a Random Process



(Details)

Filtering a Random Process

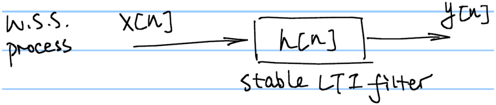




$$\Gamma_h[k] = h[k] * h^*[-k] = \sum_{l=-\infty}^{+\infty} h^*[l]\, h[k+l]$$

deterministic autocorrelation of filter's impulse response

Filtering a Random Process



In terms of ZT:

$$P_Y(z) = P_X(z)H(z)H^*(1/z^*)$$

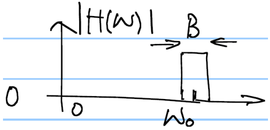
$$\Rightarrow P_Y(\omega) = P_X(\omega)H(\omega)H^*(\omega) = P_X(\omega)|H(\omega)|^2$$

When $h[n]$ is real, $H^*(z^*) = H(z)$

$$\Rightarrow P_Y(z) = P_X(z)H(z)H(1/z)$$

Interpretation of p.s.d.

If we choose $H(z)$ to be an ideal bandpass filter with very narrow bandwidth around any ω_0, and measure the output power:



$$\mathbb{E}\left[|y[n]|^2\right] = r_y(0) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} P_Y(\omega)d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{+\pi} P_X(\omega)|H(\omega)|^2 d\omega = \frac{1}{2\pi}\int_{\omega_0-B/2}^{\omega_0+B/2} P_X(\omega)\cdot 1 \cdot d\omega$$

$$\doteq \frac{1}{2\pi}P_X(\omega_0)\cdot B \geq 0$$

$$\therefore P_X(\omega_0) \doteq \mathbb{E}\left[|y[n]|^2\right]\cdot\frac{2\pi}{B}, \text{ and } P_X(\omega)\geq 0 \quad \forall\omega$$

i.e., p.s.d. is non-negative, and can be measured via power of $\{y[n]\}$.

⋇ $P_X(\omega)$ can be viewed as a density function describing how the power in $x[n]$ varies with frequency. The above BPF operation also provides a way to measure it by BPF.

Summary of §1.1

Summary: Review of Discrete-Time Random Process

1. An "ensemble" of sequences, where each outcome of the sample space corresponds to a discrete-time sequence
2. A general and complete way to characterize a random process: through joint p.d.f.
3. w.s.s process: can be characterized by 1st and 2nd moments (mean, autocorrelation)
   - These moments are ensemble averages; $\mathbb{E}[x[n]]$, $r(k) = \mathbb{E}[x[n]x^*[n-k]]$
   - Time average is easier to estimate (from just 1 observed sequence)
   - **Mean ergodicity** and **autocorrelation ergodicity**: correlation function should be asymptotically decay, i.e., uncorrelated between samples that are far apart. $\Rightarrow$ the time average over large number of samples converges to the ensemble average in mean-square sense.

Characterization of w.s.s. Process through Correlation Matrix and p.s.d.

1. Define a vector on signal samples (note the indexing order):
   $\underline{u}[n] = [u(n), u(n-1), ..., u(n-M+1)]^T$
2. Take expectation on the outer product:
   $$\mathbf{R} \triangleq \mathbb{E}\left[\underline{u}[n]\underline{u}^H[n]\right] = \begin{bmatrix} r(0) & r(1) & \cdots & \cdots & r(M-1) \\ r(-1) & r(0) & r(1) & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ r(-M+1) & \cdots & \cdots & \cdots & r(0) \end{bmatrix}$$
3. Correlation function of w.s.s. process is a one-variable deterministic sequence $\Rightarrow$ take DTFT($r[k]$) to get p.s.d.
   We can take DTFT on one sequence from the sample space of random process; different outcomes of the process will give different DTFT results; p.s.d. describes the statistical power distribution of the random process in spectrum domain.

Properties of Correlation Matrix and p.s.d.

4. Properties of correlation matrix:
   - Toeplitz (by w.s.s.)
   - Hermitian (by conjugate symmetry of $r[k]$);
   - non-negative definite

   Note: if we reversely order the sample vector, the corresponding correlation matrix will be transposed. This is the convention used in Hayes book (i.e. the sample is ordered from $n - M + 1$ to n), while Haykin's book uses ordering of n, $n - 1$, ... to $n - M + 1$.

5. Properties of p.s.d.:
   - real-valued (by conjugate symmetry of correlation function);
   - non-negative (by non-negative definiteness of $\mathbf{R}$ matrix)

Filtering a Random Process

1. Each specific realization of the random process is just a discrete-time signal that can be filtered in the way we've studied in undergrad DSP.
2. The ensemble of the filtering output is a random process. What can we say about the properties of this random process given the input process and the filter?
3. The results will help us further study such an important class of random processes that are generated by filtering a noise process by discrete-time linear filter with rational transfer function. Many discrete-time random processes encountered in practice can be well approximated by such a rational transfer function model: ARMA, AR, MA (see §II.1.2)

Detailed Derivations

Mean Ergodicity

A w.s.s. process $\{u[n]\}$ is <u>mean ergodic</u> in the mean square error sense if $\lim_{N\to\infty} \mathbb{E}\left[|m - \hat{m}(N)|^2\right] = 0$

Question: under what condition will this be satisfied?

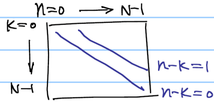
$$E\left[|\hat{m}(N) - m|^2\right] = E\left[\left|\frac{1}{N}\sum_{n=0}^{N-1} u[n] - m\right|^2\right]$$

$$= \frac{1}{N^2} E\left[\left|\sum_{n=0}^{N-1}(u[n] - m)\right|^2\right]$$

$$= \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{k=0}^{N-1} E\left[(u[n]-E(u[n]))(u[k]-E(u[k]))^*\right]$$

$$= \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{k=0}^{N-1} c(n-k)$$

$$\overset{\ell \triangleq n-k}{=} \frac{1}{N}\sum_{\ell=-N+1}^{N-1}\left(1 - \frac{|\ell|}{N}\right) c(\ell)$$



A total of $N-|\ell|$ terms for $c(\ell)$.

Therefore, the necessary and sufficient condition for $\{u[n]\}$ to be mean ergodic in MSE sense is

$$\lim_{N\to\infty} \frac{1}{N}\sum_{\ell=-N+1}^{N-1}\left(1 - \frac{|\ell|}{N}\right) c(\ell) = 0 \qquad [**]$$

Properties of **R**

R is Hermitian, i.e., $\mathbf{R}^H = \mathbf{R}$

Proof $r(k) \triangleq \mathbb{E}\left[u[n]u^*[n-k]\right] = \left(E[u[n-k]u^*[n]]\right)^* = [r(-k)]^*$

Bring into the above **R**, we have $\mathbf{R}^H = \mathbf{R}$

R is Toeplitz.

A matrix is said to be Toeplitz if all elements in the main diagonal are identical, and the elements in any other diagonal parallel to the main diagonal are identical.

R Toeplitz $\Leftrightarrow$ the w.s.s. property.

Properties of **R**

$\mathbf{R}$ is <u>non-negative definite</u>, i.e., $\underline{x}^H \mathbf{R} \underline{x} \geq 0,\ \forall \underline{x}$

<u>Proof</u>

Recall $\mathbf{R} \triangleq \mathbb{E}\left[\underline{u}[n]\underline{u}^H[n]\right]$. Now given $\forall \underline{x}$ (deterministic):

$$\underline{x}^H \mathbf{R} \underline{x} = \mathbb{E}\left[\underline{x}^H \underline{u}[n]\underline{u}^H[n]\underline{x}\right] = \mathbb{E}\left[\underbrace{(\underline{x}^H \underline{u}[n])}_{|x|\ \text{scalar}}(\underline{x}^H \underline{u}[n])^*\right] =$$

$$\mathbb{E}\left[|\underline{x}^H \underline{u}[n]|^2\right] \geq 0$$

- eigenvalues of a Hermitian matrix are real.
  (similar relation in FT analysis: real in one domain becomes conjugate symmetric in another)
- eigenvalues of a non-negative definite matrix are non-negative.
  <u>Proof</u> choose $\underline{x} = \mathbf{R}$'s eigenvector $\underline{v}$ s.t. $\mathbf{R}\underline{v} = \lambda \underline{v}$,
  $\underline{v}^H \mathbf{R} \underline{v} = \underline{v}^H \lambda \underline{v} = \lambda \underline{v}^H \underline{v} = \lambda |v|^2 \geq 0 \Rightarrow \lambda \geq 0$

Properties of **R**

Recursive relations: correlation matrix for $(M+1) \times 1$ $\underline{u}[n]$:

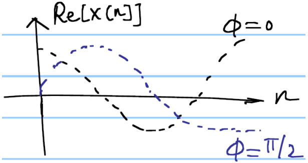
$$R_{M+1} = \begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r^*(1) & r(0) & \cdots & r(M-1) \\ r^*(2) & r^*(1) & \ddots & \vdots \\ \vdots & \vdots & & \\ r^*(M) & r^*(M-1) & & r(0) \end{bmatrix} \quad \text{and} \quad \underline{u}_{M+1}[n] = \begin{bmatrix} \underline{u}_M[n] \\ \hline u(n-M) \end{bmatrix} = \begin{bmatrix} u[n] \\ \hline \underline{u}_M[n-1] \end{bmatrix}$$

$$= \left[\begin{array}{c|c} r(0) & \underline{r}^H \\ \hline \underline{r} & R_M \end{array}\right] = \left[\begin{array}{c|c} R_M & (\underline{r}^B)^* \\ \hline (\underline{r}^B)^T & r(0) \end{array}\right]$$

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

(4) Example: Complex Sinusoidal Signal

$x[n] = A \exp[j(2\pi f_0 n + \phi)]$ where A and f_0 are real constant, $\phi \sim$ uniform distribution over $[0, 2\pi)$ (i.e., random phase)



We have:
$\mathbb{E}[x[n]] = 0 \;\forall n$

$\mathbb{E}[x[n]x^*[n-k]]$
$= \mathbb{E}[A\exp[j(2\pi f_0 n + \phi)] \cdot A\exp[-j(2\pi f_0 n - 2\pi f_0 k + \phi)]]$
$= A^2 \cdot \exp[j(2\pi f_0 k)]$

$\therefore\; x[n]$ is zero-mean w.s.s. with $r_x(k) = A^2 \exp(j2\pi f_0 k)$.

Example: Complex Sinusoidal Signal with Noise

Let $y[n] = x[n] + w[n]$ where $w[n]$ is white Gaussian noise uncorrelated to $x[n]$, $w[n] \sim N(0, \sigma^2)$

Note: for <u>white noise</u>, $\mathbb{E}[w[n]w^*[n-k]] = \begin{cases} \sigma^2 & k = 0 \\ 0 & o.w. \end{cases}$

$$
\begin{aligned}
r_y(k) &= \mathbb{E}[y[n]y^*[n-k]] \\
&= \mathbb{E}[(x[n] + w[n])(x^*[n-k] + w^*[n-k])] \\
&= r_x[k] + r_w[k] \quad (\because \mathbb{E}[x[\cdot]w[\cdot]] = 0 \text{ uncorrelated and } w[\cdot] \text{ zero mean}) \\
&= A^2 \exp[j2\pi f_0 k] + \sigma^2 \delta[k]
\end{aligned}
$$

$$
\therefore \mathbf{R}_y = \mathbf{R}_x + \mathbf{R}_w = A^2 \underline{e}\underline{e}^H + \sigma^2 \mathbb{I}, \text{ where } \underline{e} = \begin{bmatrix} 1 \\ e^{-j2\pi f_0} \\ e^{-j4\pi f_0} \\ \vdots \\ e^{-j2\pi f_0 (M-1)} \end{bmatrix}
$$

Rank of Correlation Matrix

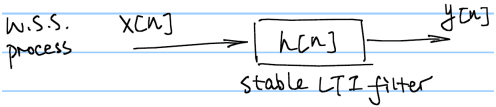
Questions:

The rank of $\mathbf{R}_x = 1$ ($\because$ only one independent row/column, corresponding to only one frequency component f_0 in the signal)

The rank of $\mathbf{R}_w = M$

The rank of $\mathbf{R}_y = M$

Filtering a Random Process



① $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[n-k]\,h[k]$

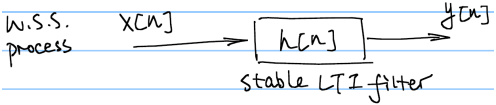
$E[y[n]] = m_x \sum_{k=-\infty}^{+\infty} h[k] = m_x H(\omega)\big|_{\omega=0}$

② $r_{yx}(n+k, n) \triangleq E[\,y[n+k]\,x^*[n]\,] = E\left[\sum_{l=-\infty}^{+\infty} x[n+k-l]\,h[l]\,x^*[n]\right]$

$= \sum_{l=-\infty}^{+\infty} r_x(k-l)\,h[l]$ i.e. $r_{yx}(n+k, n)$ depends only on k and not on n.

$\Rightarrow r_{yx}(k) = r_x(k) * h[k]$

Filtering a Random Process



③ $r_y(n+k, n) = E[y[n+k]\, y^*[n]] = E\left[y[n+k] \sum_{l=-\infty}^{+\infty} x^*[n-l]\, h^*[l]\right]$

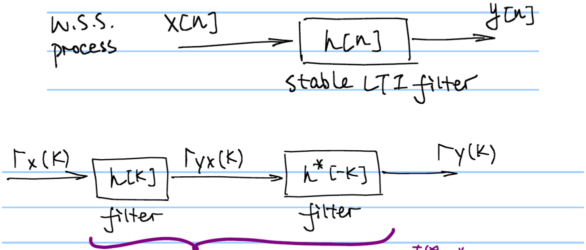
$= \sum_{l=-\infty}^{+\infty} r_{yx}(k+l)\, h^*[l] \underset{l' \triangleq -l}{=} \sum_{l'=-\infty}^{+\infty} r_{yx}(k-l')\, h^*[-l']$

i.e. $E(y[n])$ & $r_y(\cdot)$ is not a func. of n $\Rightarrow$ $\{y[n]\}$ is w.s.s.

$\Rightarrow \; r_y(k) = r_{yx}(k) * h^*[-k] = r_x(k) * h[k] * h^*[-k]$

$= \sum_{l=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} h[l]\, h^*[-m]\, r_x(k-l-m)$

Filtering a Random Process



$$\Gamma_h[k] = h[k] * h^*[-k] = \sum_{l=-\infty}^{+\infty} h^*[l]\, h[k+l]$$

deterministic autocorrelation of filter's impulse response

Parametric Signal Modeling and Linear Prediction Theory 1. Discrete-time Stochastic Processes (2)

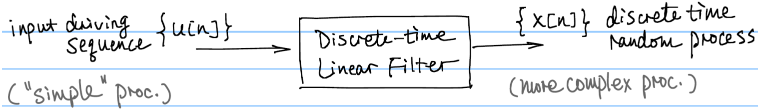
Electrical & Computer Engineering
University of Maryland, College Park

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: October 25, 2011.

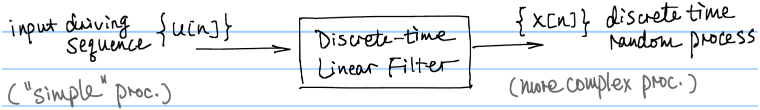
(1) The Rational Transfer Function Model

Many discrete-time random processes encountered in practice can be well approximated by a rational function model (Yule 1927).



Readings: Haykin 4th Ed. 1.5

The Rational Transfer Function Model



Typically $u[n]$ is a noise process, gives rise to randomness of $x[n]$.

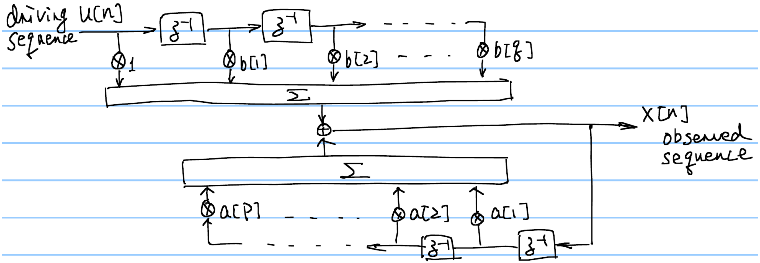
The input driving sequence $u[n]$ and the output sequence $x[n]$ are related by a linear constant-coefficient difference equation

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + \sum_{k=0}^{q} b[k]u[n-k]$$

This is called the autoregressive-moving average (ARMA) model:

- autoregressive on previous outputs
- moving average on current & previous inputs

The Rational Transfer Function Model



The system transfer function

$$H(z) \triangleq \frac{X(z)}{U(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{\sum_{k=0}^{p} a[k]z^{-k}} \triangleq \frac{B(z)}{A(z)}$$

To ensure the system's stationarity, $a[k]$ must be chosen s.t. all poles are inside the unit circle.

(2) Power Spectral Density of ARMA Processes

Recall the relation in autocorrelation function and p.s.d. after filtering:

$$
\begin{aligned}
r_x[k] &= h[k] * h^*[-k] * r_u[k] \\
P_x(z) &= H(z)H^*(1/z^*)P_U(z) \\
\Rightarrow P_x(\omega) &= |H(\omega)|^2 P_U(\omega)
\end{aligned}
$$

$\{u[n]\}$ is often chosen as a white noise process with zero mean and variance σ^2, then $P_{\text{ARMA}}(\omega) \triangleq P_X(\omega) = \sigma^2 |\frac{B(\omega)}{A(\omega)}|^2$,
i.e., the p.s.d. of $x[n]$ is determined by $|H(\omega)|^2$.

We often pick a filter with $a[0] = b[0] = 1$ (normalized gain)

The random process produced as such is called an ARMA(p, q) process, also often referred to as a pole-zero model.

(3) MA and AR Processes

MA Process

If in the ARMA model $a[k] = 0 \ \forall k > 0$, then

$$x[n] = \sum_{k=0}^{q} b[k]u[n-k]$$

This is called an MA(q) process with $P_{\text{MA}}(\omega) = \sigma^2 |B(\omega)|^2$. It is also called an all-zero model.

AR Process

If $b[k] = 0 \ \forall k > 0$, then

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + u[k]$$

This is called an AR(p) process with $P_{\text{AR}}(\omega) = \frac{\sigma^2}{|A(\omega)|^2}$. It is also called an all-pole model.

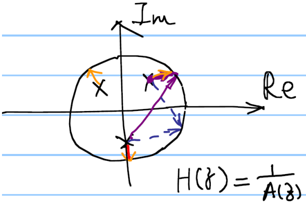
$$H(z) = \frac{1}{(1-c_1 z^{-1})(1-c_2 z^{-1}) \cdots (1-c_p z^{-1})}$$

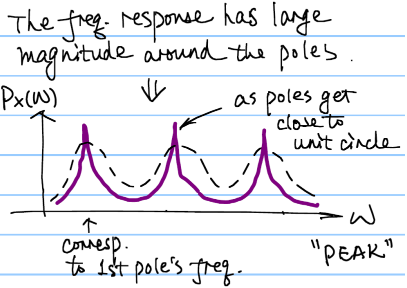
(4) Power Spectral Density: AR Model

$$\text{ZT:} \quad P_X(z) = \sigma^2 H(z) H^*(1/z^*) = \sigma^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}$$

$$\text{p.s.d.:} \quad P_X(\omega) = P_X(z)|_{z=e^{j\omega}} = \sigma^2 |H(\omega)|^2 = \sigma^2 \left|\frac{B(\omega)}{A(\omega)}\right|^2$$

- AR model: all poles $H(z) = \frac{1}{(1-c_1 z^{-1})(1-c_2 z^{-1})\cdots(1-c_p z^{-1})}$



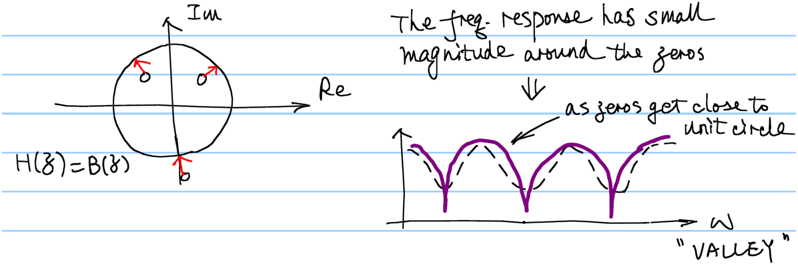


Power Spectral Density: MA Model

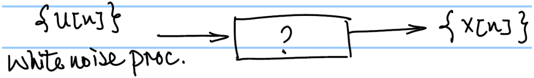
$$\text{ZT:} \quad P_X(z) = \sigma^2 H(z) H^*(1/z^*) = \sigma^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}$$

$$\text{p.s.d.:} \quad P_X(\omega) = P_X(z)|_{z=e^{j\omega}} = \sigma^2 |H(\omega)|^2 = \sigma^2 |\frac{B(\omega)}{A(\omega)}|^2$$

- MA model: all zeros



(5) Parameter Equations

Motivation: 

Want to determine the filter parameters that gives $\{x[n]\}$ with desired autocorrelation function?

Or observing $\{x[n]\}$ and thus the estimated $r(k)$, we want to figure out what filters generate such a process? (i.e., ARMA modeling)

Readings: Hayes §3.6

Parameter Equations: ARMA Model

Recall that the power spectrum for ARMA model

$$P_X(z) = H(z)H^*(1/z^*)\sigma^2$$

and $H(z)$ has the form of $H(z) = \frac{B(z)}{A(z)}$

$\Rightarrow P_X(z)A(z) = H^*(1/z^*)B(z)\sigma^2$

$\Rightarrow \sum_{\ell=0}^{p} a[\ell] r_x[k-\ell] = \sigma^2 \sum_{\ell=0}^{q} b[\ell] h^*[\ell-k], \forall k.$
(convolution sum)

Parameter Equations: ARMA Model

For the filter $H(z)$ (that generates the ARMA process) to be causal, $h[k] = 0$ for $k < 0$.
Thus the above equation array becomes

Yule-Walker Equations for ARMA process

$$\begin{cases} r_x[k] = -\sum_{\ell=1}^{p} a[\ell] r_x[k-\ell] + \sigma^2 \sum_{\ell=0}^{q-k} h^*[\ell] b[\ell+k], \; k = 0, \ldots, q \\ r_x[k] = -\sum_{\ell=1}^{p} a[\ell] r_x[k-\ell], k \geq q+1. \end{cases}$$

The above equations are a set of **nonlinear** equations
(relate $r_x[k]$ to the parameters of the filter)

Parameter Equations: AR Model

For AR model, $b[\ell] = \delta[\ell]$. The parameter equations become

$$r_x[k] = -\sum_{\ell=1}^{p} a[\ell] r_x[k-\ell] + \sigma^2 h^*[-k]$$

Note:

1. $r_x[-k]$ can be determined by $r_x[-k] = r_x^*[k]$ ($\because$ w.s.s.)
2. $h^*[-k] = 0$ for $k > 0$ by causality, and
   $h^*[0] = [\lim_{z\to\infty} H(z)]^* = \left(\frac{b[0]}{a[0]}\right)^* = 1$

Yule-Walker Equations for AR Process

$$\Rightarrow r_x[k] = \begin{cases} -\sum_{\ell=1}^{p} a[\ell] r_x[-\ell] + \sigma^2 & \text{for } k = 0 \\ -\sum_{\ell=1}^{p} a[\ell] r_x[k-\ell] & \text{for } k \geq 1 \end{cases}$$

The parameter equations for AR are **linear** equations in $\{a[\ell]\}$

Parameter Equations: AR Model

Yule-Walker Equations in matrix-vector form

$$\Rightarrow \begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p+1) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+2) \\ \vdots & & \ddots & \vdots \\ r(p-1) & \cdots & & r(0) \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \\ \vdots \\ r_x(p) \end{bmatrix}$$

put together $k = 1, \ldots, p$

i.e., $\mathbf{R}^T \underline{a} = -\underline{r}$

- $\mathbf{R}$: correlation matrix
- $\underline{r}$: autocorrelation vector

If $\mathbf{R}$ is non-singular, we have $\underline{a} = -(\mathbf{R}^T)^{-1}\underline{r}$.

We'll see better algorithm computing $\underline{a}$ in §2.3.

Parameter Equations: MA Model

For MA model, $a[\ell] = \delta[\ell]$, and $h[\ell] = b[\ell]$. The parameter equations become

$$r_x[k] = \delta^2 \sum_{\ell=0}^{q} b[\ell] b^*[\underbrace{\ell - k}_{\triangleq \ell'}] = \sigma^2 \sum_{\ell'=-k}^{q-k} b[\ell' + k] b^*[\ell']$$

And by causality of $h[n]$ (and $b[n]$), we have

$$r_x[k] = \begin{cases} \sigma^2 \sum_{\ell=0}^{q-k} b^*[\ell] b[\ell + k] & \text{for } k = 0, 1, \ldots, q \\ 0 & \text{for } k \geq q + 1 \end{cases}$$

This is again a set of **non-linear** equations in $\{b[\ell]\}$.

(6) Wold Decomposition Theorem

Recall the earlier example: $y[n] = A\exp[j2\pi f_0 n + \phi)] + w[n]$

- ϕ: (initial) random phase
- $w[n]$ white noise

Theorem

Any stationary w.s.s. discrete time stochastic process $\{x[n]\}$ may be expressed in the form of $x[n] = u[n] + s[n]$, where

1. $\{u[n]\}$ and $\{s[n]\}$ are mutually uncorrelated processes, i.e., $\mathbb{E}\left[u[m]s^*[n]\right] = 0\ \forall m, n$
2. $\{u[n]\}$ is a general random process represented by MA model: $u[n] = \sum_{k=0}^{\infty} b[k]v[n-k]$, $\sum_{k=0}^{\infty} |b_k|^2 < \infty$, $b_0 = 1$
3. $\{s[n]\}$ is a predictable process (i.e., can be predicted from its own pass with zero prediction variance): $s[n] = -\sum_{k=1}^{\infty} a[k]s[n-k]$

Corollary of Wold Decomposition Theorem

ARMA(p,q) can be a good general model for stochastic processes: has a predictable part and a new random part ("innovation process").

Corollary (Kolmogorov 1941)

Any ARMA or MA process can be represented by an AR process (of infinite order).

Similarly, any ARMA or AR process can be represented by an MA process (of infinite order).

Example: Represent ARMA(1,1) by AR(∞) or MA(∞)

E.g., for an ARMA(1, 1), $H_{\text{ARMA}}(z) = \frac{1+b[1]z^{-1}}{1+a[1]z^{-1}}$

1. Use an AR(∞) to represent it:

2. Use an MA(∞) to represent it:

(7) Asymptotic Stationarity of AR Process

Example: we initialize the generation of an AR process with specific status of $x[0], x[-1], \ldots, x[-p+1]$ (e.g., set to zero) and then start the regression $x[1], x[2], \ldots,$

$$x[n] = -\sum_{\ell=1}^{p} a[\ell]x[n-\ell] + u[n]$$

The initial zero states are deterministic and the overall random process has changing statical behavior, i.e., non-stationary.

Asymptotic Stationarity of AR Process

If all poles of the filter in the AR model are inside the unit circle, the temporary nonstationarity of the output process (e.g., due to the initialization at a particular state) can be gradually forgotten and the output process becomes asymptotically stationary.

This is because $H(z) = \frac{1}{\sum_{k=0}^{p} a_k z^{-k}} = \sum_{k=1}^{p} \frac{A_k}{1-\rho_k z^{-1}}$

$$\Rightarrow h[n] = \sum_{k=1}^{p'} A_k \rho_k^n + \sum_{k=1}^{p''} c_k r_k^n \cos(\omega_k n + \phi_k)$$

p': # of real poles
p'': # of complex poles, $\rho_i = r_i e^{\pm j\omega_i}$
$\Rightarrow p = p' + 2p''$ for real-valued $\{a_k\}$.

If all $|\rho_k| < 1$, $h[n] \to 0$ as $n \to \infty$.

Asymptotic Stationarity of AR Process

The above analysis suggests the effect of the input and past outputs on future output is only **short-term**.

So even if the system's output is initially zero to initialize the process's feedback loop, the system can gradually forget these initial states and become **asymptotically stationary** as $n \to \infty$. (i.e., be more influenced by the "recent" w.s.s. samples of the driving sequence)

Detailed Derivations

Example: Represent ARMA(1,1) by AR(∞) or MA(∞)

E.g., for an ARMA(1, 1), $H_{\text{ARMA}}(z) = \frac{1+b[1]z^{-1}}{1+a[1]z^{-1}}$

1. Use an AR(∞) to represent it, i.e.,
   $H_{\text{AR}}(z) = \frac{1}{1+c[1]z^{-1}+c[2]z^{-2}+\ldots}$
   $\Rightarrow$ Let $\frac{1+a[1]z^{-1}}{1+b[1]z^{-1}} = \frac{1}{H_{\text{AR}}(z)} = 1 + c[1]z^{-1} + c[2]z^{-2} + \ldots$
   inverse ZT $\therefore c[k] = \mathbb{Z}^{-1}\left[H_{\text{ARMA}}^{-1}(z)\right]$
   $$\Rightarrow \begin{cases} c[0] = 1 \\ c[k] = (a[1] - b[1])(-b[1])^{k-1} \text{ for } k \geq 1. \end{cases}$$

2. Use an MA(∞) to represent it, i.e.,
   $H_{\text{MA}}(z) = 1 + d[1]z^{-1} + d[2]z^{-2} + \ldots$
   $\therefore d[k] = \mathbb{Z}^{-1}\left[H_{\text{ARMA}}(z)\right]$
   $$\Rightarrow \begin{cases} d[0] = 1 \\ d[k] = (b[1] - a[1])(-a[1])^{k-1} \text{ for } k \geq 1. \end{cases}$$

Part-II Parametric Signal Modeling and Linear Prediction Theory
2. Discrete Wiener Filtering

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 1, 2011.

Preliminaries

[Readings: Haykin's 4th Ed. Chapter 2, Hayes Chapter 7]

- Why prefer FIR filters over IIR?

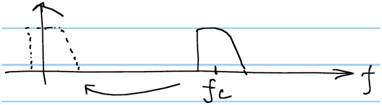
  $\Rightarrow$ FIR is inherently stable.

- Why consider complex signals?
  - Baseband representation is complex valued for narrow-band messages modulated at a carrier frequency.
  - Corresponding filters are also in complex form.

$u[n] = u_I[n] + j u_Q[n]$

- $u_I[n]$: in-phase component •

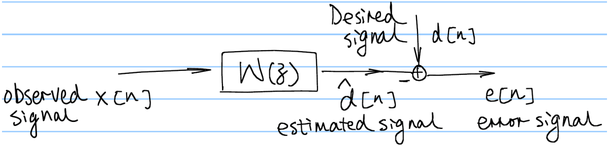
$u_Q[n]$: quadrature component



the two parts can be amplitude modulated by $\cos 2\pi f_c t$ and $\sin 2\pi f_c t$.

(1) General Problem

(Ref: Hayes §7.1)



Want to process $x[n]$ to minimize the difference between the estimate and the desired signal in some sense:

A major class of estimation (for simplicity & analytic tractability) is to use linear combinations of $x[n]$ (i.e. via linear filter).

When $x[n]$ and $d[n]$ are from two w.s.s. random processes, we often choose to minimize the mean-square error as the performance index.

$$\min_{\underline{w}} J \triangleq \mathbb{E}\left[|e[n]|^2\right] = \mathbb{E}\left[\left|d[n] - \hat{d}[n]\right|^2\right]$$

(2) Categories of Problems under the General Setup

1. Filtering
2. Smoothing
3. Prediction
4. Deconvolution

Wiener Problems: Filtering & Smoothing

- Filtering
  - The classic problem considered by Wiener
  - $x[n]$ is a noisy version of $d[n]$: $[n] = d[n] + v[n]$
  - The goal is to estimate the true $d[n]$ using a causal filter (i.e., from the current and post values of $x[n]$)
  - The causal requirement allows for filtering on the fly
- Smoothing
  - Similar to the filtering problem, except the filter is allowed to be non-causal (i.e., all the $x[n]$ data is available)

Wiener Problems: Prediction & Deconvolution

- Prediction
  - The causal filtering problem with $d[n] = x[n+1]$, i.e., the Wiener filter becomes a linear predictor to predict $x[n+1]$ in terms of the linear combination of the previous value $x[n], x[n-1], , \ldots$
- Deconvolution
  - To estimate $d[n]$ from its filtered (and noisy) version $x[n] = d[n] * g[n] + v[n]$
  - If $g[n]$ is also unknown $\Rightarrow$ blind deconvolution. We may iteratively solve for both unknowns

FIR Wiener Filter for w.s.s. processes

Design an FIR Wiener filter for jointly w.s.s. processes $\{x[n]\}$ and $\{d[n]\}$:

$W(z) = \sum_{k=0}^{M-1} a_k z^{-k}$ (where a_k can be complex valued)

$\hat{d}[n] = \sum_{k=0}^{M-1} a_k x[n-k] = \underline{a}^T \underline{x}[n]$ (in vector form)

$$\Rightarrow e[n] = d[n] - \hat{d}[n] = d[n] - \underbrace{\sum_{k=0}^{M-1} a_k x[n-k]}_{\hat{d}[n]=\underline{a}^T\underline{x}[n]}$$

$$J = E\left[|e[n]|^2\right] = E\left[e[n]\, e^*[n]\right]$$

$$= E\left[|d[n]|^2\right] - E\left[d[n] \sum_{k=0}^{M-1} a_k^* x^*[n-k]\right] - E\left[d^*[n] \sum_{k=0}^{M-1} a_k x[n-k]\right] + E\left[\sum_{k=0}^{M-1}\sum_{l=0}^{M-1} a_k a_l^* x[n-k] x^*[n-l]\right]$$

$$= E\left[|d[n]|^2\right] - \sum_{k=0}^{M-1} a_k^* E\left[d[n] x^*[n-k]\right] - \sum_{k=0}^{M-1} a_k E\left[d^*[n] x[n-k]\right] + \sum_{k=0}^{M-1}\sum_{l=0}^{M-1} a_k a_l^* \underbrace{E\left[x[n-k] x^*[n-l]\right]}_{r_x(l-k)}$$

FIR Wiener Filter for w.s.s. processes

In matrix-vector form:

$$J = \mathbb{E}\left[|d[n]|^2\right] - \underline{a}^H \underline{p}^* - \underline{p}^T \underline{a} + \underline{a}^H \mathbf{R} \underline{a}$$

where $\underline{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-M+1 \end{bmatrix}$, $\underline{p} = \begin{bmatrix} \mathbb{E}\left[x[n]d^*[n]\right] \\ \vdots \\ \mathbb{E}\left[x[n-M+1]d^*[n]\right] \end{bmatrix}$,

$\underline{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_{M-1} \end{bmatrix}$.

- $\mathbb{E}\left[|d[n]|^2\right]$: σ^2 for zero-mean random process
- $\underline{a}^H \mathbf{R} \underline{a}$: represent $\mathbb{E}\left[\underline{a}^T \underline{x}[n] \underline{x}^H[n] \underline{a}^*\right] = \underline{a}^T \mathbf{R} \underline{a}^*$

Perfect Square

1. If $\mathbf{R}$ is positive definite, $\mathbf{R}^{-1}$ exists and is positive definite.
2. $(\mathbf{R}\underline{a}^* - \underline{p})^H \mathbf{R}^{-1} (\mathbf{R}\underline{a}^* - \underline{p}) = (\underline{a}^T \mathbf{R}^H - \underline{p}^H)(\underline{a}^* - \mathbf{R}^{-1}\underline{p}) =$ $\underline{a}^T \mathbf{R}^H \underline{a}^* - \underline{p}^H \underline{a}^* - \underline{a}^T \underbrace{\mathbf{R}^H \mathbf{R}^{-1}}_{=\mathbb{I}} \underline{p} + \underline{p}^H \mathbf{R}^{-1} \underline{p}$

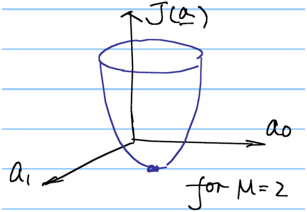
Thus we can write $J(\underline{a})$ in the form of perfect square:

$$J(\underline{a}) = \underbrace{\mathbb{E}\left[|d[n]|^2\right] - \underline{p}^H \mathbf{R}^{-1} \underline{p}}_{\text{Not a function of } \underline{a}\text{; Represent } J_{\min}.} + \underbrace{(\mathbf{R}\underline{a}^* - \underline{p})^H \mathbf{R}^{-1} (\mathbf{R}\underline{a}^* - \underline{p})}_{>0 \text{ except being zero if } \mathbf{R}\underline{a}^* - \underline{p} = 0}$$

Perfect Square

$J(\underline{a})$ represents the error performance surface:

convex and has unique minimum at $\mathbf{R}\underline{a}^* = \underline{p}$



Thus the necessary and sufficient condition for determining the optimal linear estimator (linear filter) that minimizes MSE is

$$\mathbf{R}\underline{a}^* - \underline{p} = 0 \Rightarrow \mathbf{R}\underline{a}^* = \underline{p}$$

This equation is known as the **Normal Equation**.
A FIR filter with such coefficients is called a **FIR Wiener filter**.

Perfect Square

$\mathbf{R}\underline{a}^* = \underline{p} \quad \therefore \underline{a}^*_{\text{opt}} = \mathbf{R}^{-1}\underline{p}$ if $\mathbf{R}$ is not singular
(which often holds due to noise)

When $\{x[n]\}$ and $\{d[n]\}$ are jointly w.s.s.
(i.e., crosscorrelation depends only on time difference)

$$\underbrace{\begin{bmatrix} r_x(0) & r_x^*(1) & & \\ r_x(1) & r_x(0) & \ddots & \\ \vdots & & \ddots & \\ r_x(M-1) & & & \ddots \end{bmatrix}}_{R^T} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{M-1} \end{bmatrix}}_{\underline{a}} = \underbrace{\begin{bmatrix} r_{dx}(0) \\ \vdots \\ r_{dx}(M-1) \end{bmatrix}}_{\underline{p}^*}$$

This is also known as the Wiener-Hopf equation (the discrete-time counterpart of the continuous Wiener-Hopf integral equations)

Principle of Orthogonality

Note: to minimize a real-valued func. $f(z, z^*)$ that's analytic (differentiable everywhere) in z and z^*, set the derivative of f w.r.t. either z or z^* to zero.

- Necessary condition for minimum $J(\underline{a})$: (nece.&suff. for convex J)

$$\frac{\partial}{\partial a_k^*} J = 0 \text{ for } k = 0, 1, \ldots, M-1.$$

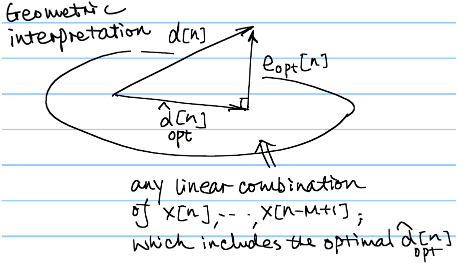
$$\Rightarrow \frac{\partial}{\partial a_k^*} \mathbb{E}\left[e[n]e^*[n]\right] = \mathbb{E}\left[e[n] \frac{\partial}{\partial a_k^*}\left(d^*[n] - \sum_{j=0}^{M-1} a_j^* x^*[n-j]\right)\right]$$
$$= \mathbb{E}\left[e[n] \cdot (-x^*[n-k])\right] = 0$$

Principal of Orthogonality

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

The optimal error signal $e[n]$ and each of the M samples of $x[n]$ that participated in the filtering are statistically uncorrelated (i.e., orthogonal in a statistical sense)

Principle of Orthogonality: Geometric View



Analogy:
r.v. $\Rightarrow$ vector;
E(XY) $\Rightarrow$ inner product of vectors

$\Rightarrow$ The optimal $\hat{d}[n]$ is the projection of $d[n]$ onto the hyperplane spanned by $\{x[n], \ldots, x[n-M+1]\}$ in a statistical sense.

The vector form: $\mathbb{E}\left[\underline{x}[n]e^*_{\text{opt}}[n]\right] = \underline{0}$.

This is true for any linear combination of $\underline{x}[n]$, and for FIR & IIR:

$$\mathbb{E}\left[\hat{d}_{\text{opt}}[n]e_{\text{opt}}[n]\right] = 0$$

Minimum Mean Square Error

Recall the perfect square form of J:

$$J(\underline{a}) = \underbrace{\mathbb{E}\left[|d[n]|^2\right] - \underline{p}^H \mathbf{R}^{-1} \underline{p}} + \underbrace{(\mathbf{R}\underline{a}^* - \underline{p})^H \mathbf{R}^{-1} (\mathbf{R}\underline{a}^* - \underline{p})}$$

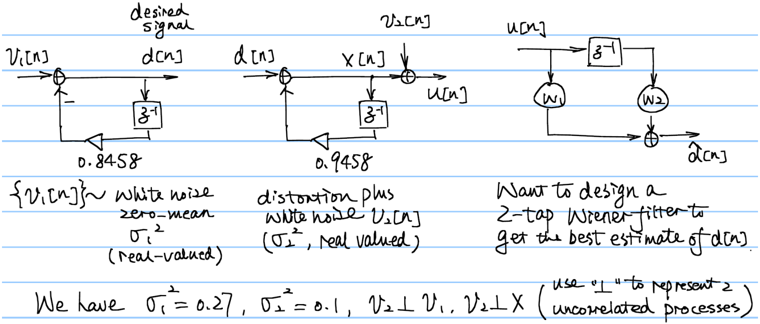
$$\therefore J_{\min} = \sigma_d^2 - \underline{a}_o^H \underline{p}^* = \sigma_d^2 - \underline{p}^H \mathbf{R}^{-1} \underline{p}$$

Also recall $d[n] = \hat{d}_{\text{opt}}[n] + e_{\text{opt}}[n]$. Since $\hat{d}_{\text{opt}}[n]$ and $e_{\text{opt}}[n]$ are uncorrelated by the principle of orthogonality, the variance is $\sigma_d^2 = \text{Var}(\hat{d}_{\text{opt}}[n]) + J_{\min}$

$$\therefore \text{Var}(\hat{d}_{\text{opt}}[n]) = \underline{p}^H \mathbf{R}^{-1} \underline{p}$$

$$= \underline{a}_0^H \underline{p}^* = \underline{p}^H \underline{a}_o^* = \underline{p}^T \underline{a}_o \quad \text{real and scalar}$$

Example and Exercise



- What kind of process is $\{x[n]\}$?
- What is the correlation matrix of the channel output?
- What is the cross-correlation vector?
- $w_1 = ?$ $\quad w_2 = ?$ $\quad J_{\min} = ?$

Detailed Derivations

Another Perspective (in terms of the gradient)

Theorem: If $f(\underline{z}, \underline{z}^*)$ is a **real-valued** function of complex vectors $\underline{z}$ and $\underline{z}^*$, then the vector pointing in the direction of the maximum rate of the change of f is $\bigtriangledown_{\underline{z}^*} f(\underline{z}, \underline{z}^*)$, which is a vector of the derivative of $f()$ w.r.t. each entry in the vector $\underline{z}^*$.

Corollary: Stationary points of $f(\underline{z}, \underline{z}^*)$ are the solutions to $\bigtriangledown_{\underline{z}^*} f(\underline{z}, \underline{z}^*) = 0$.

Complex gradient of a complex function:

	$\underline{a}^H \underline{z}$	$\underline{z}^H \underline{a}$	$\underline{z}^H A \underline{z}$
$\bigtriangledown_{\underline{z}}$	$\underline{a}^*$	0	$A^T \underline{z}^* = (A\underline{z})^*$
$\bigtriangledown_{\underline{z}^*}$	0	$\underline{a}$	$A\underline{z}$

Using the above table, we have $\bigtriangledown_{\underline{a}^*} J = -\underline{p}^* + \mathbf{R}^T \underline{a}$.

For optimal solution: $\bigtriangledown_{\underline{a}^*} J = \frac{\partial}{\partial \underline{a}^*} J = 0$

$\Rightarrow \mathbf{R}^T \underline{a} = \underline{p}^*$, or $\mathbf{R}\underline{a}^* = \underline{p}$, the Normal Equation. $\therefore \underline{a}^*_{\text{opt}} = \mathbf{R}^{-1} \underline{p}$

(Review on matrix & optimization: Hayes 2.3; Haykins(4th) Appendix A,B,C)

Review: differentiating complex functions and vectors

<1> Differentiable at z_0

$\lim_{\Delta z \to 0} \frac{f(z_0+\Delta z)-f(z_0)}{\Delta z}$ exist $\Rightarrow$ Need to converge in all directions for $\Delta z \to 0$

Recall: $f(z)$ is analytic (i.e. differentiable everywhere) on region D if $f(z)=u(x,y)+iv(x,y)$ is continuous and satisfy Cauchy-Riemann condition $\frac{\partial u}{\partial x}=\frac{\partial v}{\partial y}$ & $\frac{\partial u}{\partial y}=-\frac{\partial v}{\partial x}$.

<2> e.g. $f_1(z)=zz^*=|z|^2=(x^2+y^2)+i\times 0$

$f_2(z)=z^*=x-iy$

$\Rightarrow$ DOES NOT satisfy Cauchy-Riemann.

Review: differentiating complex functions and vectors

unlike the real value optimiz. case, $\frac{df(x)}{dx}=0$. ← Note: $f(z) = |z|^2$ has unique minimum at $z=0$, but not differentiable from complex analysis (any func. that depends on z^* is not differentiable)

We can either minimize $f(x,y)$ wrt x & y where $z = x+iy$, or treat z and z^* as indep. variables and minimize $f(z, z^*)$ w.r.t. both z and z^*: i.e. $\frac{\partial f}{\partial z} = 0$ and $\frac{\partial f}{\partial z^*} = 0$

Minimizing a real-valued func. of z and z^* (and the func. is analytic w.r.t. both z and z^*) is somewhat easier: the optimal points may be found by setting the derivative of $f(z, z^*)$ w.r.t. either z or z^* equal to zero and solve for z. ☆

e.g. $f(z, z^*) = |z|^2 = z \cdot z^*$. sufficient to have $\frac{\partial f}{\partial z^*} = z = 0$.

Differentiating complex functions: More details

$z = x + iy$ $\qquad f(z) = u(x,y) + i\,v(x,y)$

Note: $x = \frac{1}{2}(z + z^*)$, $\quad y = \frac{1}{2i}(z - z^*)$ $\quad \Rightarrow \begin{cases} \frac{\partial f}{\partial x} = \frac{\partial u}{\partial x} + i\frac{\partial v}{\partial x} \\ \frac{\partial f}{\partial y} = \frac{\partial u}{\partial y} + i\frac{\partial v}{\partial y} \end{cases}$

$$\begin{cases} \frac{\partial f}{\partial z} \overset{\text{def}}{=} \frac{1}{2}\left[\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}(-i)\right] \\ \frac{\partial f}{\partial z^*} \overset{\text{def}}{=} \frac{1}{2}\left[\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\cdot i\right] \end{cases} \qquad \text{i.e. } \frac{\partial f}{\partial z} = \frac{\partial f}{\partial x}\cdot\frac{\partial x}{\partial z} + \frac{\partial f}{\partial y}\cdot\frac{\partial y}{\partial z}$$

For real-valued $f(z)$: i.e. $f(z) = u(x,y)$,

we have: ① $\frac{\partial f}{\partial z} = \left(\frac{\partial f}{\partial z^*}\right)^*$; ② Gradient $\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} \Rightarrow$ written as complex# $\frac{\partial u}{\partial x} + i\frac{\partial u}{\partial y} = \frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}$

E.g. ① if $f(z) = z = x + iy$

$$\frac{\partial f}{\partial z^*} \overset{\text{def}}{=} \frac{1}{2}\left[\frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}\right] = \frac{1}{2}[1 + i\cdot i] = 0 \ ; \quad \frac{\partial f}{\partial z} \overset{\text{def}}{=} \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right) = \frac{1}{2}(1 - i\cdot i) = 1$$

E.g. ② $f(z) = |z|^2$.

Let $A \overset{\text{def}}{=} \frac{f(z+\Delta z) - f(z)}{\Delta z} = \frac{(z+\Delta z)(z^* + (\Delta z)^*) - z\cdot z^*}{\Delta z} = z^* + (\Delta z)^* + z\frac{(\Delta z)^*}{\Delta z}$

for $\Delta z = \Delta x + 0\cdot i$: $\lim_{\Delta x \to 0} \frac{(\Delta z)^*}{\Delta z} = 1 \Rightarrow A \to z^* + z$

for $\Delta z = 0 + \Delta y\cdot i$: $\lim_{\Delta y \to 0} \frac{(\Delta z)^*}{\Delta z} = -1 \Rightarrow A \to z^* - z$

$\Rightarrow$ A converges to different results for different directions as $\Delta z \to 0$ except for $z = 0$

$\therefore$ the limit doesn't exist, except for $z = 0$ (and thus not differentiable)

Example: solution

① What is $\{x[n]\}$?

$$d[n] = -0.8458\, d[n-1] + v_1[n] \quad \Rightarrow \quad H_1(z) = \frac{1}{1+0.8458z^{-1}}$$

$$x[n] = 0.9458\, x[n-1] + d[n] \quad \Rightarrow \quad H_2(z) = \frac{1}{1-0.9458z^{-1}}$$

$$H(z) = H_1(z)\, H_2(z) = \frac{1}{1+0.8458z^{-1}} \cdot \frac{1}{1-0.9458z^{-1}}$$

$$= \frac{1}{1-0.1z^{-1}-0.8z^{-2}} = \frac{1}{1+a_1z^{-1}+a_2z^{-2}}$$

i.e. $\{x[n]\}$ is an AR(2) process of

$$x[n] - 0.1\,x[n-1] - 0.8\,x[n-2] = v_1[n]$$

Example: solution

② The channel output is $u[n] = x[n] + v_2[n]$

$$\underset{2\times 2}{R_u} = E\left[\underline{u}[n]\underline{u}^H[n]\right] = \begin{bmatrix} r_u(0) & r_u(1) \\ r_u(1)^* & r_u(0) \end{bmatrix} = R_x + R_{v_2}$$

$$= \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

[where $r_x(\cdot)$ can be obtained from the AR parameter equation as seen from the example at the end of §2.1:

$$r_x(0) = \left(\frac{1+a_2}{1-a_2}\right)\frac{\sigma_1^2}{(1+a_2)^2 - a_1^2} = 1, \qquad r_x(1) = \frac{-a_1}{1+a_2} r_x(0) = 0.5$$

Example: solution

③ Obtain the cross correlation vector $\underline{P} = E\left[d^*[n] \begin{pmatrix} u[n] \\ u[n-1] \end{pmatrix} \right]$

$$E[d[n]u[n]] = E[(x[n] - 0.9458x[n-1])(x[n] + v_2[n])]$$
$$= r_x(0) - 0.9458\, r_x(-1) = 1 - 0.9458 \times 0.5$$
$$= 1 - 0.4729 = 0.5271$$

Similarly, $E[d[n]u[n-1]] = r_x(1) - 0.9458\, r_x(0) = -0.4458$

$$\therefore \underline{P} = \begin{bmatrix} 0.5271 \\ -0.4458 \end{bmatrix}$$

④ optimal weights are

$$\underline{W}_o = R^{-1}\underline{P} = \begin{bmatrix} 0.8360 \\ -0.7853 \end{bmatrix}$$

$$J(w_1, w_2) = 0.9486 - 1.0544w_1 + 0.8916w_2 + w_1w_2 + 1.1(w_1^2 + w_2^2)$$

$$\Rightarrow J_{min} = 0.1579$$

Preliminaries

- In many communication and signal processing applications, messages are modulated onto a carrier wave. The bandwidth of message is usually much smaller than the carrier frequency $\Rightarrow$ i.e., the signal modulated is "narrow-band".
- It is convenient to analyze in the baseband form to remove the effect of the carrier wave by translating signal down in frequency yet fully preserve the information in the message.
- The baseband signal so obtained is complex in general. $u[n] = u_I[n] + ju_Q[n]$
- Accordingly, the filters developed for the applications are also in complex form to preserve the mathematical formulations and elegant structures of the complex signal in the applications.

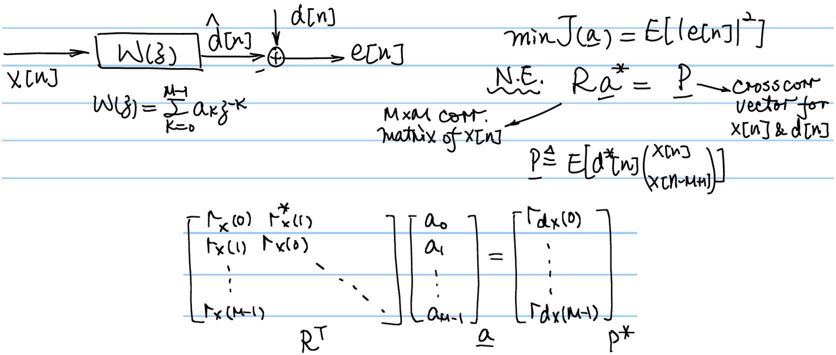
Part-II Parametric Signal Modeling and Linear Prediction Theory 3. Linear Prediction

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 3, 2011.

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 $k = 0, ...M-1$.

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

$\Rightarrow r_{dx}(k) = \sum_{\ell=0}^{M-1} a_\ell r_x(k-\ell) \Rightarrow$ 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 $\underline{a} \Rightarrow \mathrm{Var}(\hat{d}[n]) = \underline{a}^T \underline{p} = \underline{p}^H \mathbf{R}^{-1} \underline{p}$

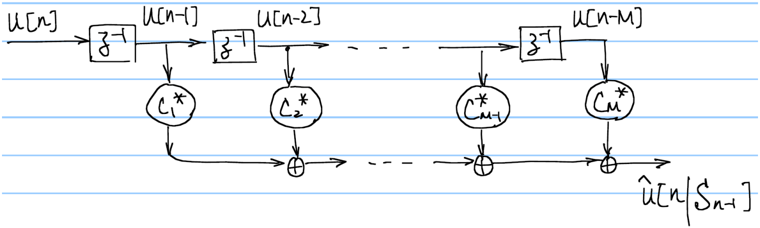
May also use the vector form to derive N.E.: set gradient $\bigtriangledown_{\underline{a}^*} J = 0$

Forward Linear Prediction

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

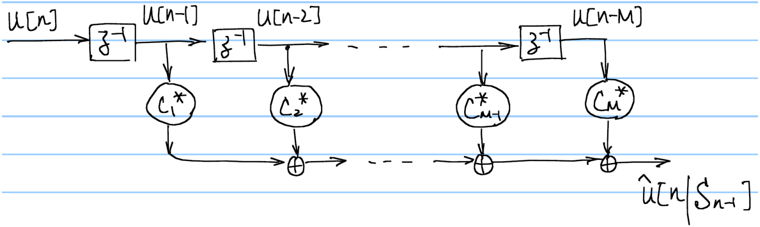
Let $c_k \triangleq a_k^*$ (i.e., c_k^* represents the filter coefficients and helps us to avoid many conjugates in the normal equation)

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



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

$\mathbb{S}_{n-1}$ 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

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

  $e[n]$ $\quad$ $d[n]$ $\quad$ $\leftarrow$ 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

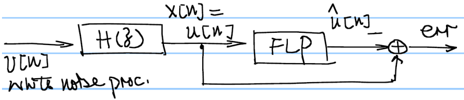
3. Thus the Normal Equation for FLP is

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

The prediction error is

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

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



$$\begin{cases} u[n] = -\sum_{k=1}^{p} a_k u[n-k] + v[n] \\ u[n] = e[n] + \hat{u}[n] \\ \quad\;\; = e[n] + \sum_{k=1}^{p} c_k^* u[n-k] \end{cases}$$

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

Yule-Walker Eq.

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

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

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

$\Rightarrow$ 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.

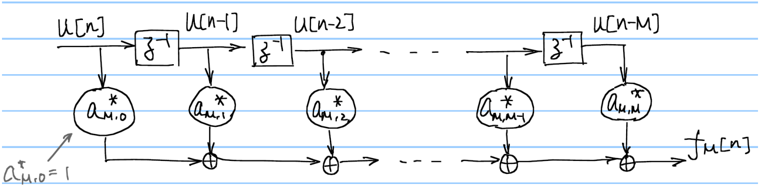
Question: Optimal predictor for $\{u[n]\}$=AR(p) when $p < M$?

Forward-Prediction-Error Filter

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

Let $a_{M,k} = \begin{cases} 1 & k = 0 \\ -c_k & k = 1, 2, \ldots, M \end{cases}$, 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)

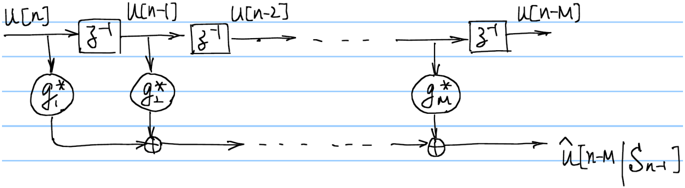
Backward Linear Prediction

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

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

- $\mathbb{S}_n$: 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: $P_{M,\text{BLP}} = r(0) - (\underline{r}^B)^T \underline{g}$

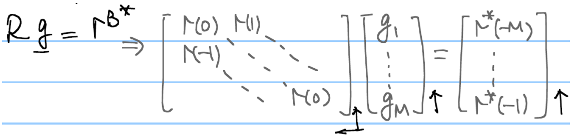
Relations between FLP and BLP

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

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

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

reversely order:



$\therefore$ 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

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

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

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's 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) $\underline{r}^{B*H} \mathbf{R}^{-1} \underline{r}^{B*} = (\underline{r}^H \mathbf{R}^{T*^{-1}} \underline{r})^{B*} = \underline{r}^H \mathbf{R}^{-1} \underline{r}$

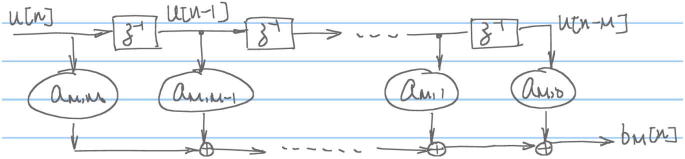
Backward-Prediction-Error Filter

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

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

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 have high correlation among 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.

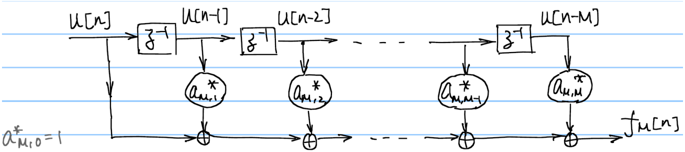
Tool #3: Linear Prediction. we can first figure out the best predictor for a chunk of approximately stationary samples, encode the first sample, then do prediction and encode the prediction residues (as well as the prediction parameters).

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

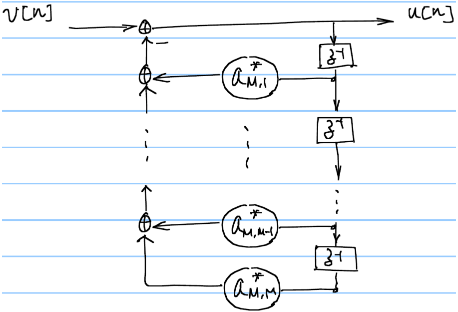
$$u[n] + a^*_{M,1}u[n-1] + \ldots + a^*_{M,M}u[n-M] = f_M[n]$$



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 will use the synthesis structure to reconstruct to speech signal, with a suitable driving sequence (periodic impulse train for voiced sound; white noise for fricative sound)

Detailed Derivations

Review: Recursive Relation of Correlation Matrix

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

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

Summary: General Wiener vs. FLP

	General Wiener	Forward LP	Backward LP
Tap input	$\underline{x}[n]$	$\underline{u}[n-1]$	
Desired response	$d[n]$	$u[n]$	
(conj) Weight vector	$\underline{c} = \underline{a}^*$	$\underline{c}$	
Estimated sig	$\hat{d}[n]$	$\hat{d}[n] = \underline{c}^H \underline{u}[n-1]$	
Estimation error	$e[n]$	$f_M[n]$	
Correlation matrix	$\mathbf{R}_M$	$\mathbf{R}_M$	
Cross-corr vector	$\underline{p}$	$\underline{r}$	
MMSE	$J_{\min}$	P_M	
Normal Equation	$\mathbf{R}\underline{c} = \underline{p}$	$\mathbf{R}\underline{c} = \underline{r}$	
Augmented N.E.		$\mathbf{R}_{M+1}\underline{a}_M = \begin{bmatrix} P_M \\ \underline{0} \end{bmatrix}$	

(return)

Summary: General Wiener vs. FLP vs. BLP

	General Wiener	Forward LP	Backward LP
Tap input	$\underline{x}[n]$	$\underline{u}[n-1]$	$\underline{u}[n]$
Desired response	$d[n]$	$u[n]$	$u[n-M]$
(conj) Weight vector	$\underline{c} = \underline{a}^*$	$\underline{c}$	$\underline{g}$
Estimated sig	$\hat{d}[n]$	$\hat{d}[n] = \underline{c}^H \underline{u}[n-1]$	$\hat{d}[n] = \underline{g}^H \underline{u}[n]$
Estimation error	$e[n]$	$f_M[n]$	$b_M[n]$
Correlation matrix	$\mathbf{R}_M$	$\mathbf{R}_M$	$\mathbf{R}_M$
Cross-corr vector	$\underline{p}$	$\underline{r}$	$\underline{r}^{B^*}$
MMSE	$J_{\min}$	P_M	P_M
Normal Equation	$\mathbf{R}\underline{c} = \underline{p}$	$\mathbf{R}\underline{c} = \underline{r}$	$\mathbf{R}\underline{g} = \underline{r}^{B^*}$
Augmented N.E.		$\mathbf{R}_{M+1}\underline{a}_M = \begin{bmatrix} P_M \\ \underline{0} \end{bmatrix}$	$\mathbf{R}_{M+1}\underline{a}_M^{B^*} = \begin{bmatrix} \underline{0} \\ P_M \end{bmatrix}$

Matrix Inversion Lemma for Homework

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

special case:

$$(A + \underset{B}{\underline{u}}\,\underset{D}{\underline{v}}^H)^{-1} = A^{-1} - \frac{A^{-1}\underline{u}\,\underline{v}^H A^{-1}}{1 + \underline{v}^H A^{-1}\underline{u}}$$

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

Parametric Signal Modeling and Linear Prediction Theory 4. The Levinson-Durbin Recursion

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, 2011.

Complexity in Solving Linear Prediction

(Refs: Hayes §5.2; Haykin 4th Ed. §3.3)

Recall Augmented Normal Equation for linear prediction:

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

As $\mathbf{R}_{M+1}$ is usually non-singular, $\underline{a}_M$ may be obtained by inverting $\mathbf{R}_{M+1}$, or Gaussian elimination for solving equation array:

$\Rightarrow$ Computational complexity $O(M^3)$.

Motivation for More Efficient Structure

Complexity in solving a general linear equation array:

- Method-1: invert the matrix, e.g. compute determinant of $\mathbf{R}_{M+1}$ matrix and the adjacency matrices

  $\Rightarrow$ matrix inversion has $O(M^3)$ complexity

- Method-2: use Gaussian elimination

  $\Rightarrow$ approximately $M^3/3$ multiplication and division

By exploring the structure in the matrix and vectors in LP, Levison-Durbin recursion can reduce complexity to $O(M^2)$

- M steps of order recursion, each step has a linear complexity w.r.t. intermediate order
- Memory use: Gaussian elimination $O(M^2)$ for the matrix, vs. Levinson-Durbin $O(M)$ for the autocorrelation vector and model parameter vector.

Levinson-Durbin recursion

The **Levinson-Durbin recursion** is an order-recursion to efficiently solve the Augmented N.E.

> M steps of order recursion, each step has a linear complexity w.r.t. intermediate order

The recursion can be stated in two ways:

1. Forward prediction point of view
2. Backward prediction point of view

Two Points of View of LD Recursion

Denote $\underline{a}_m \in \mathbb{C}^{(m+1)\times 1}$ as the tap weight vector of a forward-prediction-error filter of order $m = 0, ..., M$.

$a_{m-1,0} = 1$, $a_{m-1,m} \triangleq 0$, $a_{m,m} = \Gamma_m$ (a constant "**reflection coefficient**")

Forward prediction point of view

$$a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k},\ k = 0, 1, \ldots, m$$

In vector form: $\underline{a}_m = \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ \underline{a}^{B*}_{m-1} \end{bmatrix} \quad (**)$

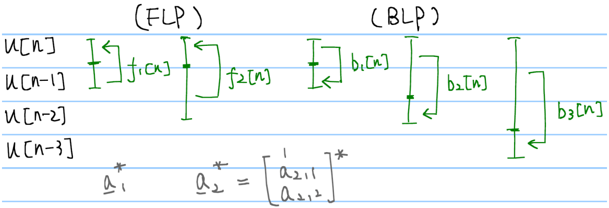
Backward prediction point of view

$$a^*_{m,m-k} = a^*_{m-1,m-k} + \Gamma^*_m a^*_{m-1,k},\ k = 0, 1, \ldots, m$$

In vector form: $\underline{a}^{B*}_m = \begin{bmatrix} 0 \\ \underline{a}^{B*}_{m-1} \end{bmatrix} + \Gamma^*_m \begin{bmatrix} \underline{a}_{m-1} \\ 0 \end{bmatrix}$

(can be obtained by reordering and conjugating $(**)$)

Recall: Forward and Backward Prediction Errors



- $f_m[n] = u[n] - \hat{u}[n] = \underline{a}_m^H \underbrace{\underline{u}[n]}_{(m+1)\times 1}$
- $b_m[n] = u[n-m] - \hat{u}[n-m] = \underline{a}_m^{B,T} \underline{u}[n]$

(3) Rationale of the Recursion

Left multiply both sides of $(**)$ by $\mathbf{R}_{m+1}$:

LHS: $\mathbf{R}_{m+1}\underline{a}_m = \begin{bmatrix} P_m \\ \underline{0}_m \end{bmatrix}$ (by augmented N.E.)

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

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

RHS (2): $\mathbf{R}_{m+1}\begin{bmatrix} 0 \\ \underline{a}_{m-1}^{B*} \end{bmatrix} = \begin{bmatrix} r(0) & \underline{r}^H \\ \underline{r} & \mathbf{R}_m \end{bmatrix}\begin{bmatrix} 0 \\ \underline{a}_{m-1}^{B*} \end{bmatrix}$

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

Computing Γ_m

Put together LHS and RHS: for the order update recursion $(**)$ to hold, we should have

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

$$\Rightarrow \begin{cases} P_m = P_{m-1} + \Gamma_m \Delta^*_{m-1} \\ 0 = \Delta_{m-1} + \Gamma_m P_{m-1} \end{cases}$$

$\Rightarrow$

$$a_{m,m} = \Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}}$$

$$P_m = P_{m-1}\left(1 - |\Gamma_m|^2\right)$$

Caution: not to confuse P_m and Γ_m!

(4) Reflection Coefficients Γ_m

To ensure the prediction MSE $P_m \geq 0$ and P_m non-increasing when we increase the order of the predictor (i.e., $0 \leq P_m \leq P_{m-1}$), we require $|\Gamma_m|^2 \leq 1$ for $\forall m > 0$.

Let $P_0 = r(0)$ as the initial estimation error has power equal to the signal power (i.e., no regression is applied), we have

$$P_M = P_0 \cdot \prod_{m=1}^{M}(1 - |\Gamma_m|^2)$$

Question: Under what situation $\Gamma_m = 0$?
i.e., increasing order won't reduce error.

Consider a process with Markovian-like property in 2nd order statistic sense (e.g. AR process) s.t. info of further past is contained in k recent samples

(5) About Δ_m

Cross-correlation of <u>BLP error</u> and <u>FLP error</u> : can be shown as
$\Delta_{m-1} = \mathbb{E}\left[b_{m-1}[n-1]f^*_{m-1}[n]\right]$

(Derive from the definition $\Delta_{m-1} \triangleq \underline{r}_m^{BT}\underline{a}_{m-1}$, and use definitions of $b_{m-1}[n-1], f^*_{m-1}[n]$ and orthogonality principle.)

Thus the reflection coefficient can be written as

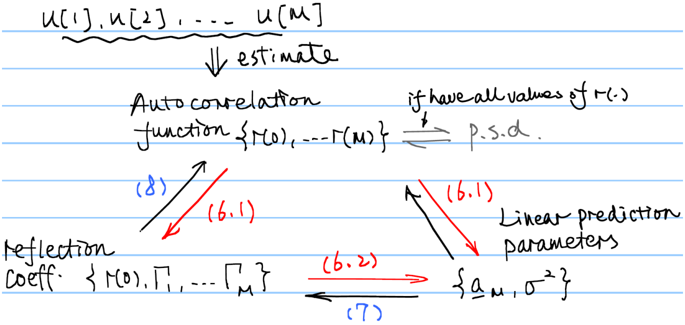
$$\Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}} = -\frac{\mathbb{E}\left[b_{m-1}[n-1]f^*_{m-1}[n]\right]}{\mathbb{E}\left[|f_{m-1}[n]|^2\right]}$$

Note: for the 0th order predictor, use mean value (zero) as estimate, s.t. $f_0[n] = u[n] = b_0[n]$,

$\therefore \Delta_0 = \mathbb{E}\left[b_0[n-1]f_0^*[n]\right] = \mathbb{E}\left[u[n-1]u^*[n]\right] = r(-1) = r^*(1)$

Preview: Relations of w.s.s and LP Parameters

For w.s.s. process $\{u[n]\}$:



(6) Computing $\underline{a}_M$ and P_M by Forward Recursion

Case-1 : If we know the autocorrelation function $r(\cdot)$:

① $\Delta_0 = r(-1)$, $P_0 = r(0)$

② for $m = 1, \ldots M$ (order recursion)

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

for $k = 1, \ldots m$ (diff predictor parameters for order-m)

$$a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k}$$

(where $a_{m-1,0} = 1$; $a_{m-1,m} = 0$)

$$\Delta_m = r^{BT}_{m+1} \underline{a}_m$$

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

- # of iterations $= \sum_{m=1}^{M} m = \frac{M(M+1)}{2}$, comp. complexity is $O(M^2)$
- $r(k)$ can be estimated from time average of one realization of $\{u[n]\}$:

  $\hat{r}(k) = \frac{1}{N-k} \sum_{n=k+1}^{N} u[n]u^*[n-k], \; k = 0, 1, \ldots, M$
  (recall correlation ergodicity)

(6) Computing $\underline{a}_M$ and P_M by Forward Recursion

Case-2 : If we know Γ_1, Γ_2, ..., Γ_M and $P_0 = r(0)$, we can carry out the recursion for $m = 1, 2, \ldots, M$:

$$\begin{cases} a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k}, \; k = 1, \ldots, m \\ P_m = P_{m-1} \left(1 - |\Gamma_m|^2\right) \end{cases}$$

(7) Inverse Form of Levinson-Durbin Recursion

Given the tap-weights $\underline{a}_M$, find the reflection coefficients $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$:

Recall: $$\begin{cases} \text{(FP) } a_{m,k} = a_{m-1,k} + \Gamma_m a^*_{m-1,m-k}, \; k = 0, \ldots, m \\ \text{(BP) } a^*_{m,m-k} = a^*_{m-1,m-k} + \Gamma^*_m a^*_{m-1,k}, \; a_{m,m} = \Gamma_m \end{cases}$$

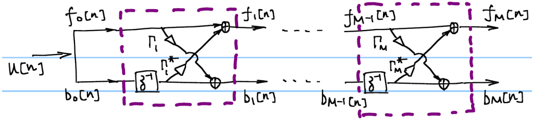
Multiply (BP) by Γ_m and subtract from (FP):

$$a_{m-1,k} = \frac{a_{m,k} - \Gamma_m a^*_{m,m-k}}{1 - |\Gamma_m|^2} = \frac{a_{m,k} - a_{m,m} a^*_{m,m-k}}{1 - |a_{m,m}|^2}, \; k = 0, \ldots, m$$

$\Rightarrow \Gamma_m = a_{m,m}$, $\Gamma_{m-1} = a_{m-1,m-1}, \ldots$, i.e., From $\underline{a}_M \Rightarrow \underline{a}_m \Rightarrow \Gamma_m$

iterate with $m = M-1, M-2, \ldots$ *to lower order*

see §5 Lattice structure:



(8) Autocorrelation Function & Reflection Coefficients

The 2nd-order statistics of a stationary time series can be represented in terms of autocorrelation function $r(k)$, or equivalently the power spectral density by taking DTFT.

Another way is to use $r(0), \Gamma_1, \Gamma_2, \ldots, \Gamma_M$.

To find the relation between them, recall:

$\Delta_{m-1} \triangleq \underline{r}_m^{BT} \underline{a}_{m-1} = \sum_{k=0}^{M-1} a_{m-1,k} r(-m+k)$ and $\Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}}$

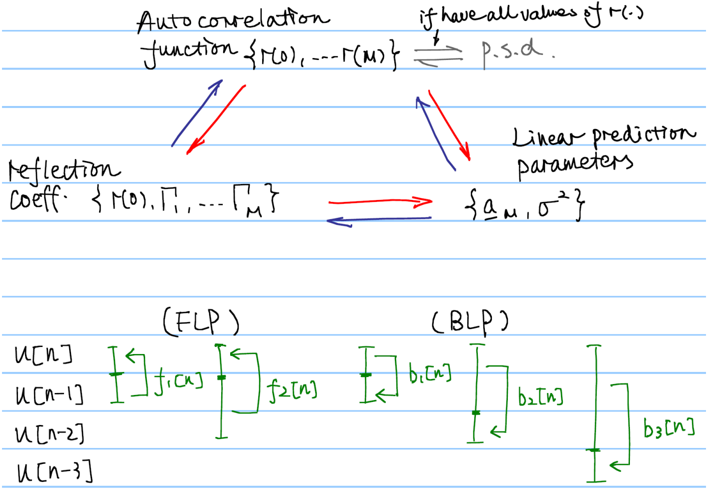
$\Rightarrow -\Gamma_m P_{m-1} = \sum_{k=0}^{m-1} a_{m-1,k} r(k-m)$, where $a_{m-1,0} = 1$.

(8) Autocorrelation Function & Reflection Coefficients

1. $r(m) = r^*(-m) = -\Gamma_m^* P_{m-1} - \sum_{k=1}^{m-1} a_{m-1,k}^* r(m-k)$
   Given $r(0), \Gamma_1, \Gamma_2, \ldots, \Gamma_M$, can get $\underline{a}_m$ using Levinson-Durbin recursion s.t. $r(1), \ldots, r(M)$ can be generated recursively.
2. Recall if $r(0), \ldots, r(M)$ are given, we can get $\underline{a}_m$.
   So $\Gamma_1, \ldots, \Gamma_M$ can be obtained recursively: $\Gamma_m = a_{m,m}$
3. These facts imply that the reflection coefficients $\{\Gamma_k\}$ can uniquely represent the 2nd-order statistics of a w.s.s. process.

Summary

Statistical representation of w.s.s. process



Detailed Derivations/Examples

Example of Forward Recursion Case-2

e.g. (case 2). Given $\Gamma_1, \Gamma_2, \Gamma_3$ and $P(0)$, find $\underline{a}_3$ and P_3 of a prediction-error filter of order 3.

⓪ $P_0 = r(0)$

① $m=1$: $a_{1,0} = 1$; $a_{1,1} = \Gamma_1$; $a_{1,2} = 0$; $P_1 = P_0(1-|\Gamma_1|^2)$

② $m=2$: $a_{2,0} = 1$; $a_{2,1} = a_{1,1} + \Gamma_2 a_{1,1}^* = \Gamma_1 + \Gamma_2 \cdot \Gamma_1^*$

used in §2.5.4. for inverse filtering

$a_{2,2} = \Gamma_2$

$P_2 = P_1(1-|\Gamma_2|^2)$

③ $m=3$: $a_{3,0} = 1$; $a_{3,1} = a_{2,1} + \Gamma_3 a_{2,2}^* = \Gamma_1 + \Gamma_2 \Gamma_1^* + \Gamma_3 \cdot \Gamma_2^*$

$a_{3,2} = a_{2,2} + \Gamma_3 a_{2,1}^* = \Gamma_2 + \Gamma_3 \Gamma_1^* + \Gamma_1 \Gamma_2^* \Gamma_3$

$a_{3,3} = \Gamma_3$

$P_3 = P_2(1-|\Gamma_3|^2)$

Proof for Δ_{m-1} Property

Proof: In HW#4

$$\Delta_{m-1} = \underline{r}_m^{BT} \underline{a}_{m-1} = [r(-m), \dots r(-1)]\, \underline{a}_{m-1}$$
$$= E[u^*[n]\, \underline{u}^B_m[n-1]]\, \underline{a}_{m-1}$$
$$= E[u^*[n]\, \underline{u}^B_m[n-1]\, \underline{a}_{m-1}]$$
$$= E[u^*[n]\, (\underline{u}_m[n-1]\, \underline{a}^B_{m-1})]$$
$$= E[u^*[n]\, b_{m-1}[n-1]]$$
$$= E[f^*_{m-1}[n]\, b_{m-1}[n-1]]$$

Recall

① $\underline{r}_M = \begin{bmatrix} r(-1) \\ \vdots \\ r(-M) \end{bmatrix}$

② $r(-k) = E[u[n-k]\, u^*[n]] = (E[u[n]\, u^*[n-k]])^*$

③ $\underline{u}_m[n-1] = \begin{bmatrix} u[n-1] \\ \vdots \\ u[n-m] \end{bmatrix}$

④ $b_{m-1}[n-1] = \sum_{k=0}^{m-1} a_{m-1,m-1-k}\, u[n-1-k] = \underline{a}^B_{m-1}\, \underline{u}_m[n-1]$

⑤ $a_{m-1,0} = 1$

$f_{m-1}[n] = \sum_{k=0}^{m-1} a^*_{m-1,k}\, u[n-k] = u[n] + \sum_{k=1}^{m-1} a^*_{m-1,k}\, u[n-k]$

⑥ $b_{m-1}[n-1] \perp \{u[n-1], \dots u[n-m+1]\}$

Haykin's 4th Ed. (P152)

* Partial correlation (PARCOR) coeff. between $f_{m-1}[n]$ and $b_{m-1}[n-1]$:

$$\rho_m \triangleq \frac{E[b_{m-1}[n-1]\, f^*_{m-1}[n]]}{(E[|b_{m-1}[n-1]|^2]\, E[|f_{m-1}[n]|^2])^{1/2}} \overset{\text{for w.s.s.}}{=\!=} \frac{\Delta_{m-1}}{P_{m-1}} = -\Gamma_m$$

Recall $E[|f_m[n]|^2] = E[|b_m[n]|^2] = P_m$

(see HW#7)

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 15, 2011.

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

2. Levinson-Durbin recursion:

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

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

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

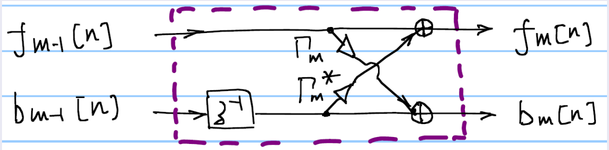
(Details)

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

Basic Lattice Structure

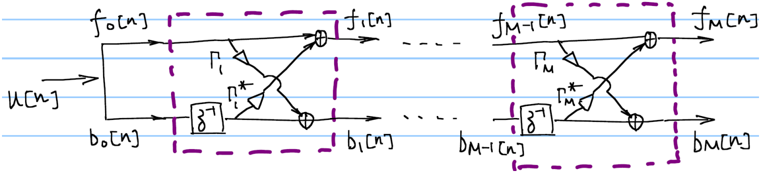
$$\begin{bmatrix} f_m[n] \\ b_m[n] \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m^* \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}[n] \\ b_{m-1}[n-1] \end{bmatrix}, \quad m = 1, 2, \ldots, M$$

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

Proof : (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)

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 the **same information**.

Correlation Properties

4. Correlations of error signals across orders and time:

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

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

Proof : (Details)

5. Correlations of error signals across orders and time:

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

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

Proof : (Details)

Correlation Properties

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

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

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

Joint Process Estimator: Motivation

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

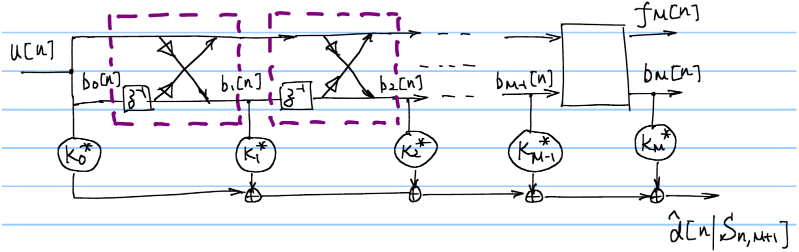
In (general) Wiener filtering theory, we use $\{x[n]\}$ process to estimate a desired response $\{d[n]\}$.

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 $\underline{b}[n]$

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

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

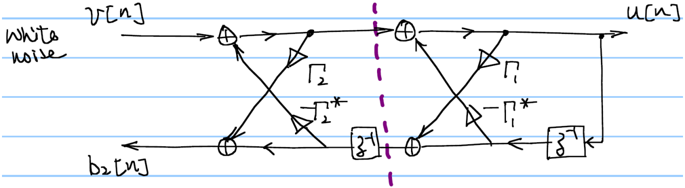
- One is a **lattice predictor** (characterized by $\Gamma_1, \ldots, \Gamma_M$) transforming a sequence of correlated samples $u[n]$, $u[n-1], \ldots, u[n-M]$ into a sequence of uncorrelated samples $b_0[n], b_1[n], \ldots, b_M[n]$.
- The other is called a **multiple regression filter** (characterized 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



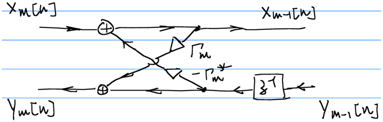
$$u[n] = v[n] - \Gamma_1^* u[n-1] - \Gamma_2^*(\Gamma_1 u[n-1] + u[n-2])$$

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

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

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

Basic Building Block for All-pole Filtering



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

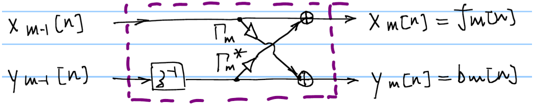
$$\Rightarrow \begin{cases} x_m[n] = x_{m-1}[n] + \Gamma_m^* y_{m-1}[n-1] \\ y_m[n] = \Gamma_m x_{m-1}[n] + y_{m-1}[n-1] \end{cases}$$

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

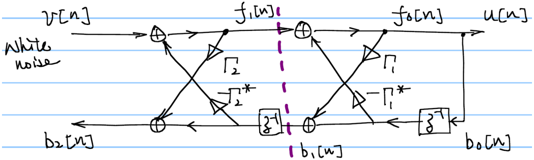
All-pole Filter via Inverse Filtering

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

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



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



Detailed Derivations

Basic Lattice Structure

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

1. FLP:

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

$$= \underline{a}_{m-1}^H \underline{u}_m[n] + \Gamma_m^* \underline{a}_{m-1}^{BT} \underline{u}_m[n-1]$$

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

2. BLP:

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

$$= \underline{a}_{m-1}^{BT} \underline{u}_m[n-1] + \Gamma_m \underline{a}_{m-1}^H \underline{u}_m[n]$$

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

Proof of Correlation Property 2

2. $E[f_m[n]\, u^*[n]] = E[b_m[n]\, u^*[n-m]] = P_m$

<u>Proof</u>:

$\because$ $u[n] = f_m[n] + \hat{u}[n \mid S_{n-1,m}]$ — recall, this denote the space of all possible linear combination of $u[n-1], \cdots u[n-m]$

and $f_m[n] \perp \hat{u}[n \mid S_{n-1,m}]$

$\therefore$ $E[f_m[n]\, u^*[n]] = E[f_m[n]\, f_m^*[n]] = P_m$

The case of $b_m[n]$ can be shown similarly: $u[n-m] = b_m[n] + \hat{u}[n-m \mid S_{n,m}]$

Proof of Correlation Property 3

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

(FLP) $E[f_m[n]f_i^*[n]] = P_m$ for $i \le m$

Proof: ① BLP case.

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

$b_i[n]$ depends only on $\{u[n], \cdots u[n-i]\}$

$$E[b_m[n]b_i^*[n]] = \sum_{k=0}^{i} a_{i,i-k}^*\, E[b_m[n]\, u^*[n-k]]$$

For $i<m$, by property-1, $E[b_m[n]\, u^*[n-k]] = 0$ for all $k=0,\cdots,i$.

Thus, $E[b_m[n]b_i^*[n]] = 0$.

For $i=m$, $E[b_m[n]\, b_m^*[n]] = P_m$.

The correlation for $i>m$ can be obtained by conjugation

if $i>m \Rightarrow E[b_m[n]\, b_i^*[n]] = \left[E[b_i[n]\, b_m^*[n]]\right]^* = 0$

Proof of Correlation Property 3 (cont'd)

② FLP case:

$$f_i[n] = \sum_{k=0}^{i} a_{i,k}^* u[n-k]$$

$$E[f_m[n] f_i^*[n]] = \underbrace{E[f_m[n] u^*[n]]}_{= P_m \text{ by Property-2}} + \sum_{k=1}^{i} a_{i,k} \underbrace{E[f_m[n] u^*[n-k]]}_{\text{all } = 0 \text{ for } i \leq m}$$

Proof of Correlation Property 4

Proof: $E[f_m[n] f_i^*[n-l]]$ $\qquad f_i[n] = \sum_{k=0}^{i} a_{i,k}^* u[n-k]$

$= \sum_{k=0}^{i} a_{i,k} \underbrace{E[f_m[n] u^*[n-(l+k)]]}$

the $E(\cdot)$ term is zero if $1 \le l+k \le m$

$\because k \le i \quad \therefore$ if $l \le m-i$, we have $l \le m-k$

$\because$ the minimum k is 0, $\therefore$ $1 \le l+k$ requires $1 \le l$

and $m-i \ge 1 \Rightarrow m > i$

Thus $E[f_m[n] f_i^*[n-l]] = 0$ for $1 \le l \le m-i$.

The case for $b_m[n]$ can be shown similarly:

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

$E[b_m[n] b_i^*[n-l]] = \sum_{k=0}^{i} a_{i,i-k}^* \underbrace{E[b_m[n] u^*[n-(l+k)]]}$

$= 0$ if $0 \le l+k \le m-1$

Proof of Correlation Property 5

Proof: $E[f_m(n+m) f_i^*(n+i)] = E[f_m(n+m) f_i^*(n+m-(m-i))]$

from Property-4, this $= 0$ if $i < m$

$E[b_m(n+m) b_i^*(n+i)] = E[b_m(n+m) b_i^*(n+m-(m-i))]$

We can't directly apply property-4 as it is for $l \leq m-i-1$.

Let $n' = n+m$, $l = m-i \geq 0$

$$E[b_m[n'] b_i^*[n'-l]] = \sum_{k=0}^{i} a_{i,i-k} E[b_m[n'] u^*[n'-(l+k)]]$$

$$= \sum_{k=0}^{i-1} a_{i,i-k} \underbrace{E[b_m[n'] u^*[n'-(l+k)]]}_{=0 \text{ as here } 0 \leq l+k \leq m-1} + \underbrace{E[b_m[n'] u^*[n'-m]]}_{=P_m \text{ from Property-2}}$$

$= P_m$

Proof of Correlation Property 6

Proof:

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

① for $i \leq m$:

$$E[f_m[n] b_i^*[n]] = E[f_m[n] \sum_{k=0}^{i} a_{i,i-k}^* u^*[n-k]]$$

$$= a_{i,i}^* E[f_m[n] u^*[n]] + \sum_{k=1}^{i} a_{i,i-k}^* \underbrace{E[f_m[n] u^*[n-k]]}_{=0 \text{ for } i \leq m}$$

$$= \Gamma_i^* P_m$$

from property #2.

② for $i > m$:

$f_i[n] = \sum_{k=0}^{i} a_{i,k}^* u[n-k]$

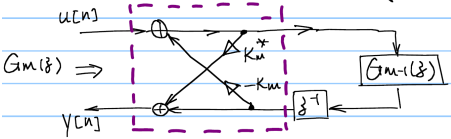
$$E[f_m[n] b_i^*[n]] = E[\sum_{k=0}^{m} a_{m,k}^* u[n-k] b_i^*[n]]$$

$$= \sum_{k=0}^{m} a_{m,k}^* \underbrace{E[u[n-k] b_i^*[n]]}_{=0 \text{ for } i > m} = 0$$

by principle of orthogonality

Lattice Filter Structure: All-pass

The lattice filter structure: (REF Vaidynathan Book §3.4.3.)



$G_0(z) = 1$

The cascaded lattice structure for all-pass filters

Let $G_m(z) = \dfrac{Y(z)}{U(z)} \Rightarrow z^{-1} G_{m-1}(z) = \dfrac{G_m(z) - K_m^*}{1 - K_m G_m(z)}$

Lattice Filter Structure: All-pass (cont'd)

Thus if $G_m(z)$ is an m^{th} order causal, stable, all pass function, it can be implemented in lattice where $|K_m| < 1$ and $G_{m-1}(z)$ is an $(m-1)^{th}$ order causal, stable, all pass function,

i.e. $G_m(z) = z^{-m} \frac{\tilde{B}_m(z)}{B_m(z)}$ — conjugate coeff. and evaluate at z^{-1}, i.e. $B_{m*}(z^{-1})$

$$\Rightarrow z^{-1} G_{m-1}(z) = \frac{z^{-m}\tilde{B}_m(z) - K_m^* B_m(z)}{B_m(z) - K_m z^{-m} \tilde{B}_m(z)}$$

1st order all pass $\frac{a^* - z^{-1}}{1 - a z^{-1}}$ pole a, zero $1/a^*$

$K_m = b_{m,m} / b_{m,0}^*$

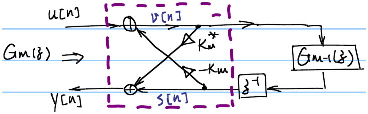
s.t. to keep the denominator's order at $(m-1)$.

$\Rightarrow z^{-m}$ term has coeff. zero $b_{m,m} - K_m b_{m,0}^* = 0$

Thus from input $v[n]$ to output $b_m[n]$ in last page is an all-pass filter.

☆ SEE Hayes Book §6.4.1 for explanation on the All-pole lattice.

Lattice Filter Structure: All-pass (cont'd)



$$Y(z) = K_m^* V(z) + S(z)$$

$$= K_m^* V(z) + z^{-1} G_{m-1}(z) V(z)$$

$$V(z) = U(z) - K_m S(z)$$

$$= U(z) - z^{-1} K_m G_{m-1}(z) V(z)$$

$$\Rightarrow Y(z) = \left[K_m^* + z^{-1} G_{m-1}(z) \right] V(z)$$

$$= \underbrace{\frac{K_m^* + z^{-1} G_{m-1}(z)}{1 + K_m z^{-1} G_{m-1}(z)}}_{G_m(z)} U(z)$$

$$\Rightarrow \quad G_m(z) = \frac{K_m^* + z^{-1} G_{m-1}(z)}{1 + K_m z^{-1} G_{m-1}(z)}$$

or equiv.

$$z^{-1} G_{m-1}(z) = \frac{G_m(z) - K_m^*}{1 - K_m G_m(z)}$$