VECTOR NOTATION FOR THE PREDICTION ERROR FILTER

The equation for the prediction error

$$\varepsilon[n] = x[n] + a_1^* x[n-1] + a_2^* x[n-2] + \cdots + a_P^* x[n-P]$$

can be written as

$$\varepsilon[n] = \boldsymbol{a}^{*T} \tilde{\boldsymbol{x}}[n] \quad \text{where} \quad \boldsymbol{a} \stackrel{\text{def}}{=} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} \qquad \boldsymbol{x}[n] \stackrel{\text{def}}{=} \begin{bmatrix} x[n-P] \\ x[n-P+1] \\ \vdots \\ x[n] \end{bmatrix}$$

NORMAL EQUATION DERIVATION

The orthogonality principle requires that

$$\mathcal{E}\left\{\begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-P] \end{bmatrix} \varepsilon^*[n]\right\} = \begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad \mathcal{E}\left\{\tilde{\boldsymbol{x}}[n]\varepsilon^*[n]\right\} = \sigma_\varepsilon^2 \boldsymbol{\iota}$$

$$\text{where } \boldsymbol{\iota} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$$

Then substitute $\varepsilon = \boldsymbol{a}^{*T}\tilde{\boldsymbol{x}}$

$$\mathcal{E}\left\{\tilde{\boldsymbol{x}}(\boldsymbol{a}^{*T}\tilde{\boldsymbol{x}})^*\right\} = \mathcal{E}\left\{\tilde{\boldsymbol{x}}(\tilde{\boldsymbol{x}}^{*T}\boldsymbol{a})\right\} = \underbrace{\mathcal{E}\left\{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^{*T}\right\}}_{\tilde{\mathbf{R}}_{\boldsymbol{x}}}\boldsymbol{a} = \sigma_\varepsilon^2\boldsymbol{\iota}$$

NORMAL EQUATIONS

$$\boxed{\tilde{\mathbf{R}}_{\boldsymbol{x}}\boldsymbol{a} = \sigma_\varepsilon^2 \boldsymbol{\iota}}$$

For order P this has the form

$$\begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[P] \\ R_x[-1] & R_x[0] & \cdots & R_x[P-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-P] & R_x[-P+1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Insert Example 7.1 here.

AUTOREGRESSIVE MODEL FOR A PROCESS

AR MODEL ($w[n]$ is white noise)

$$x[n] = -a_1 x[n-1] - a_2 x[n-2] - \cdots - a_P x[n-P] + w[n]$$

CORRELATION FUNCTION DIFFERENCE EQUATION

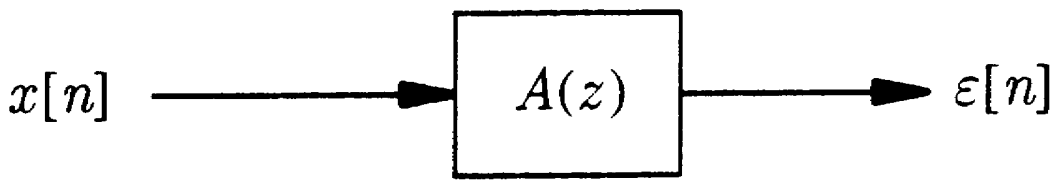
$$R_x[l] + a_1 R_x[l-1] + \cdots + a_P R_x[l-P] = R_{wx}[l] = \sigma_w^2 h^*[-l]$$

This leads to the *Yule-Walker* equations:

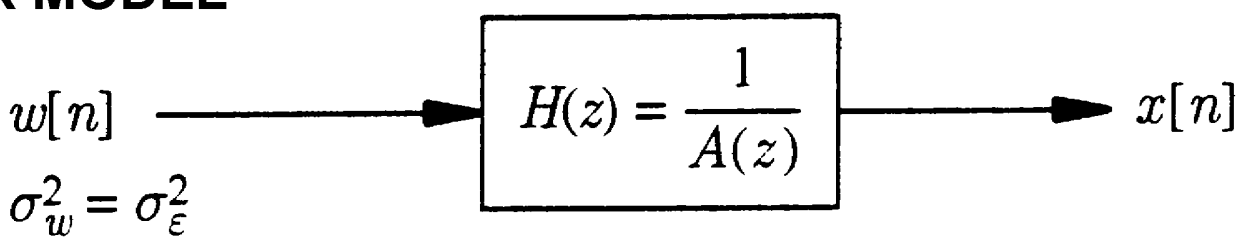
$$\begin{bmatrix} R_x[0] & R_x[-1] & \cdots & R_x[-P] \\ R_x[1] & R_x[0] & \cdots & R_x[-P+1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[P] & R_x[P-1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

COMPARISON

LINEAR PREDICTION



AR MODEL

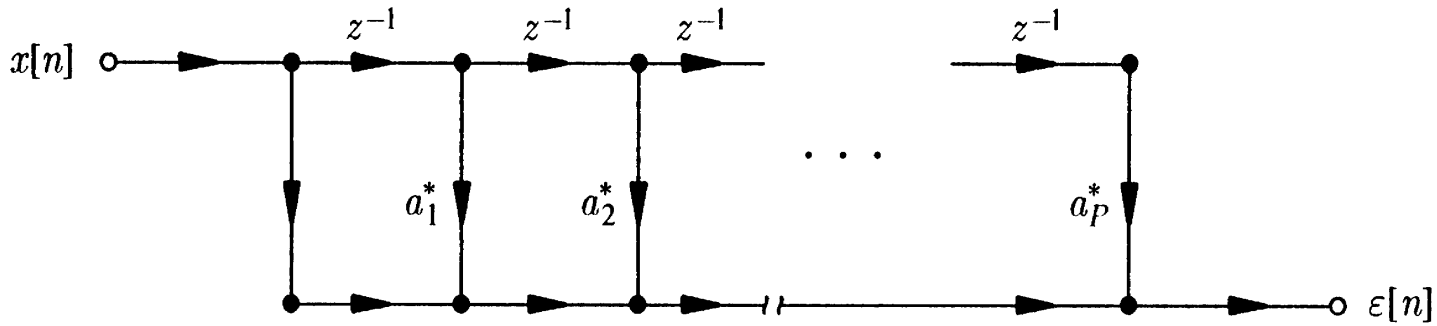


$$A(z) = 1 + a_1^* z^{-1} + a_2^* z^{-2} + \cdots + a_P^* z^{-P}$$

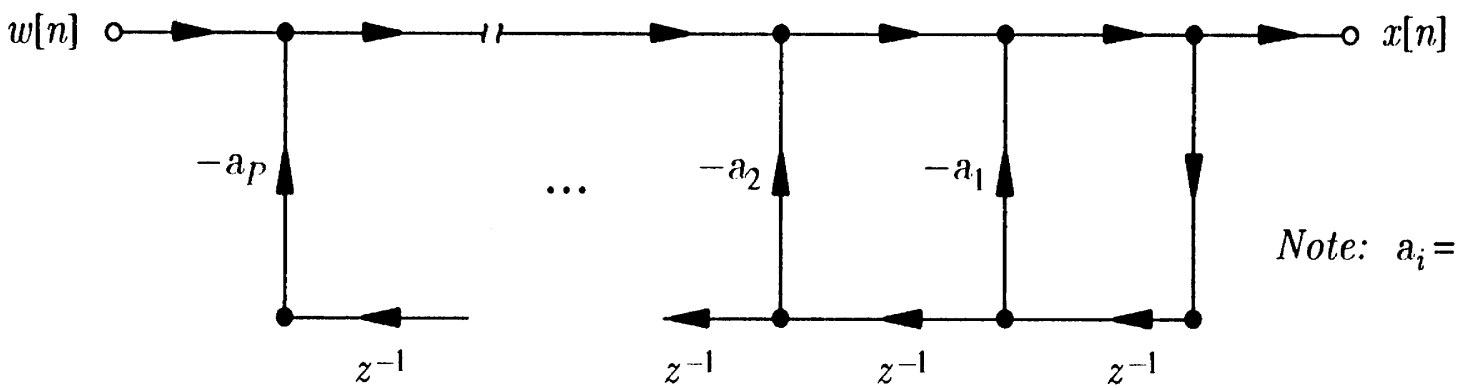
or $\quad A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_P z^{-P}$

DIRECT FORM REALIZATIONS

PREDICTION ERROR FILTER

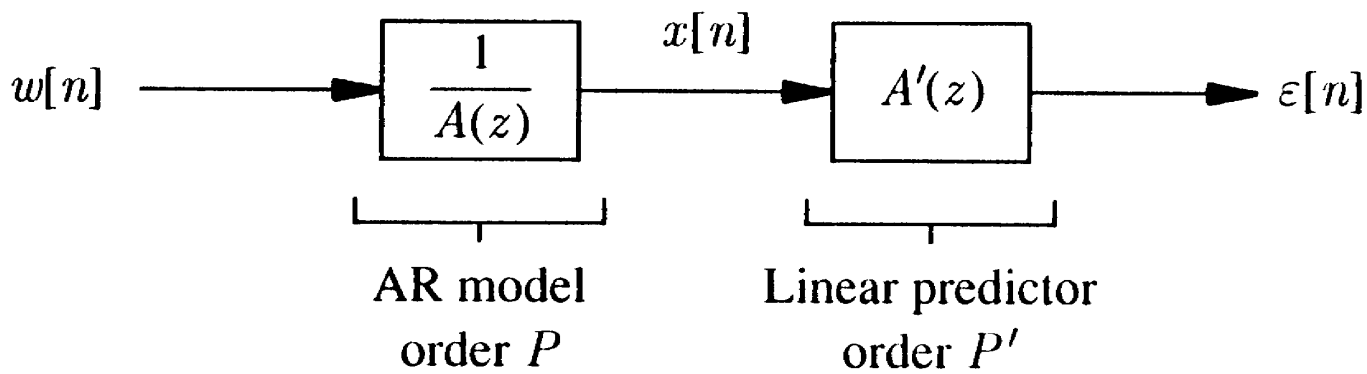


AR MODEL (ALL-POLE FILTER)



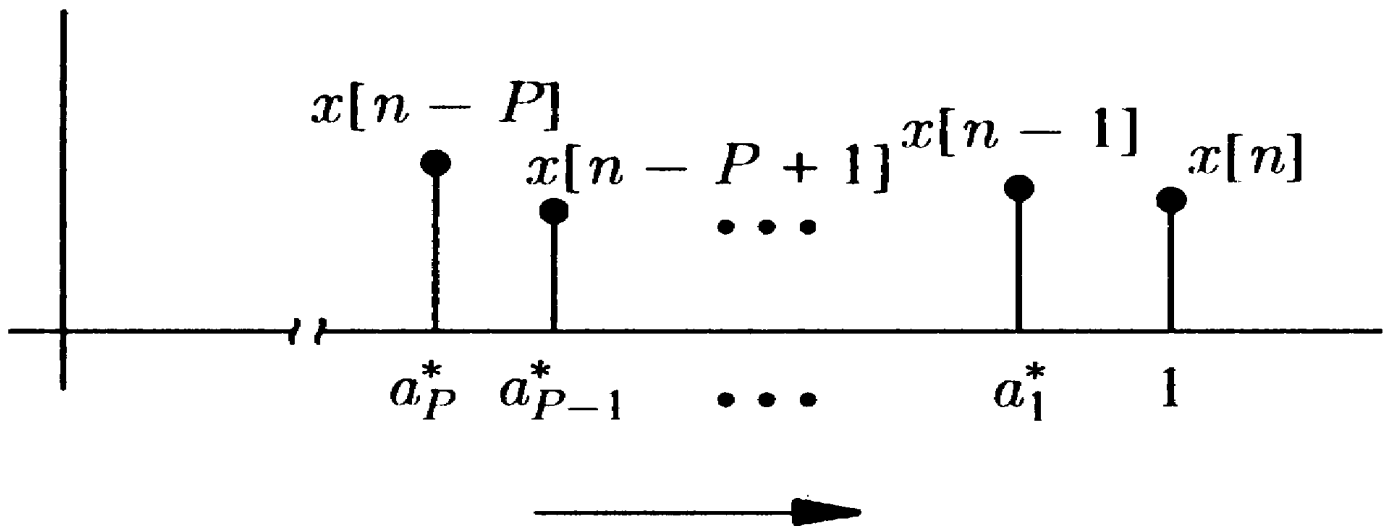
Note: $a_i = a_i^*$.

LINEAR PREDICTION FOR AR PROCESSES



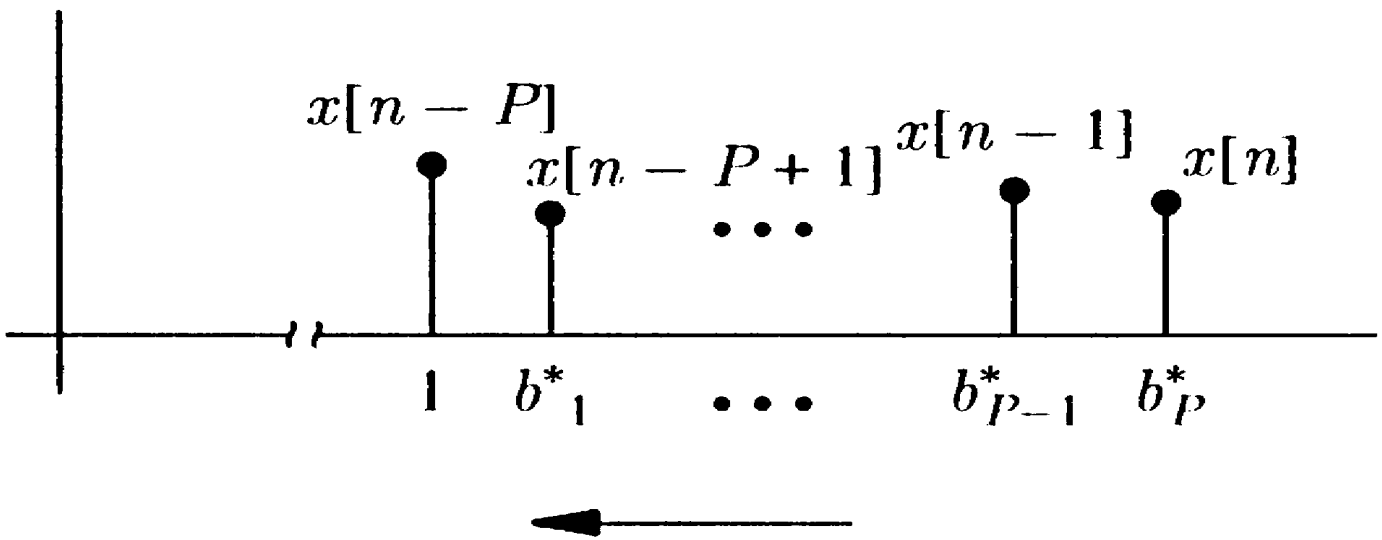
For $P' \geq P$: $\quad a_i^{(P')} = \begin{cases} a_i^* & 0 \leq i \leq P \\ 0 & P < i \leq P' \end{cases}$ and $\sigma^2_{\varepsilon_{P'}} = \sigma^2_w$

FORWARD LINEAR PREDICTION



$$\varepsilon[n] = x[n] - \hat{x}[n]$$

BACKWARD LINEAR PREDICTION



$$\varepsilon'[n-P] = x[n-P] - \hat{x}[n-P]$$

<u>Note:</u> *Backward prediction error* defined as $\varepsilon^b[n] = \varepsilon'[n-P]$.

COMPARISON OF NORMAL EQUATIONS

FORWARD

$$\begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[P] \\ R_x[-1] & R_x[0] & \cdots & R_x[P-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-P] & R_x[-P+1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

BACKWARD

$$\begin{bmatrix} R_x[0] & R_x[-1] & \cdots & R_x[-P] \\ R_x[1] & R_x[0] & \cdots & R_x[-P+1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[P] & R_x[P-1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ \vdots \\ b_P \end{bmatrix} = \begin{bmatrix} \sigma_{\varepsilon'}^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

ANTICAUSAL AR MODEL

MODEL FOR THE PROCESS ($w'[n]$ is white noise)

$$x[n] = -b_1 x[n+1] - b_2 x[n+2] - \cdots - b_P x[n+P] + w'[n]$$

YULE-WALKER EQUATIONS (Anticausal)

$$\begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[P] \\ R_x[-1] & R_x[0] & \cdots & R_x[P-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-P] & R_x[-P+1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ \vdots \\ b_P \end{bmatrix} = \begin{bmatrix} \sigma^2_{w'} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

SUMMARY OF EQUATIONS

	Forward	Backward	Relations
Normal equations (linear prediction)	$\tilde{\mathbf{R}}_{\boldsymbol{x}}\boldsymbol{a} = \sigma_{\varepsilon}^2\boldsymbol{\iota}$	$\mathbf{R}_{\boldsymbol{x}}\boldsymbol{b} = \sigma_{\varepsilon'}^2\boldsymbol{\iota}$	$\boldsymbol{a} = \boldsymbol{b}^*$ $\sigma_{\varepsilon}^2 = \sigma_{\varepsilon'}^2$
Yule–Walker equations (AR model)	$\mathbf{R}_{\boldsymbol{x}}\mathbf{a} = \sigma_{w}^2\boldsymbol{\iota}$	$\tilde{\mathbf{R}}_{\boldsymbol{x}}\mathbf{b} = \sigma_{w'}^2\boldsymbol{\iota}$	$\mathbf{a} = \mathbf{b}^*$ $\sigma_{w}^2 = \sigma_{w'}^2$

LEVINSON RECURSION

- Provides fast solution of Normal equations: computation $\propto P^2$ rather than P^3.

- Gives insights regarding linear prediction and AR modeling
  - lattice filter structure
  - additional fast algorithms

LEVINSON RECURSION (derivation)

Begin with forward Normal equations of order p:

$$\underbrace{\begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[p] \\ R_x[-1] & R_x[0] & \cdots & R_x[p-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-p] & R_x[-p+1] & \cdots & R_x[0] \end{bmatrix}}_{\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)}} \underbrace{\begin{bmatrix} 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \end{bmatrix}}_{\boldsymbol{a}_p} = \begin{bmatrix} \sigma_p^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and backward Normal equations of order p:

$$\underbrace{\begin{bmatrix} R_x[0] & R_x[-1] & \cdots & R_x[-p] \\ R_x[1] & R_x[0] & \cdots & R_x[1-p] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[p] & R_x[p-1] & \cdots & R_x[0] \end{bmatrix}}_{\mathbf{R}_{\boldsymbol{x}}^{(p)}} \underbrace{\begin{bmatrix} 1 \\ b_1^{(p)} \\ \vdots \\ b_p^{(p)} \end{bmatrix}}_{\boldsymbol{b}_p} = \begin{bmatrix} \sigma_p'^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

LEVINSON RECURSION (cont'd.)

Write forward augmented equations for order $p-1$:

$$\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} \begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p-1)}} & \tilde{\mathbf{r}}_{p-1} \\ & & \\ --- & -- & -- \\ \tilde{\mathbf{r}}_{p-1}^{*T} & & R_x[0] \end{array}\right] \begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_{p-1}^2 \\ 0 \\ \vdots \\ \Delta_p \end{bmatrix}$$

where

$$\mathbf{r}_p = \begin{bmatrix} R_x[1] \\ R_x[2] \\ \vdots \\ R_x[p+1] \end{bmatrix} \quad \text{and} \quad \Delta_p = \tilde{\mathbf{r}}_{p-1}^{*T} \boldsymbol{a}_{p-1} = \mathbf{r}_{p-1}^{*T} \tilde{\boldsymbol{a}}_{p-1}$$

LEVINSON RECURSION (cont'd.)

Backward augmented equations for order $p-1$ are:

$$\mathbf{R}_{\boldsymbol{x}}^{(p)} \begin{bmatrix} \boldsymbol{b}_{p-1} \\ -- \\ 0 \end{bmatrix} = \left[\begin{array}{cc|c} \mathbf{R}_{\boldsymbol{x}}^{(p-1)} & & \tilde{\mathbf{r}}_{p-1}^{*} \\ & & \\ --- & -- & -- \\ \tilde{\mathbf{r}}_{p-1}^{T} & & R_x[0] \end{array}\right] \begin{bmatrix} \boldsymbol{b}_{p-1} \\ -- \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_{p-1}'^2 \\ 0 \\ \vdots \\ \Delta_p' \end{bmatrix}$$

with $\Delta_p' = \tilde{\mathbf{r}}_{p-1}^T \boldsymbol{b}_{p-1} = \mathbf{r}_{p-1}^T \tilde{\boldsymbol{b}}_{p-1}$ or (in reversed form):

$$\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} \begin{bmatrix} 0 \\ -- \\ \tilde{\boldsymbol{b}}_{p-1} \end{bmatrix} = \begin{bmatrix} \Delta_p' \\ 0 \\ \vdots \\ \sigma_{p-1}'^2 \end{bmatrix}$$

LEVINSON RECURSION (cont'd.)

Form linear combination of forward and backward equations:

$$\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} \left[\begin{bmatrix} \boldsymbol{a}_{p-1} \\ \\ -- \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ -- \\ \\ \tilde{\boldsymbol{b}}_{p-1} \end{bmatrix} \right] = \begin{bmatrix} \sigma_{p-1}^2 \\ 0 \\ \vdots \\ \Delta_p \end{bmatrix} + c_1 \begin{bmatrix} \Delta_p' \\ 0 \\ \vdots \\ \sigma_{p-1}'^2 \end{bmatrix}$$

and compare to

$$\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} \boldsymbol{a}_p = \begin{bmatrix} \sigma_p^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

LEVINSON RECURSION (cont'd.)

If c_1 is chosen so that

$$\Delta_p + c_1 \sigma_{p-1}'^2 = 0 \qquad \text{i.e.,} \quad c_1 = -\Delta_p / \sigma_{p-1}'^2$$

then it will follow that

$$\boldsymbol{a}_p = \begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ -- \\ \tilde{\boldsymbol{b}}_{p-1} \end{bmatrix}$$

and

$$\sigma_p^2 = \sigma_{p-1}^2 + c_1 \Delta_p'$$

SUMMARY OF STEPS

I	$\Delta_p = \mathbf{r}_{p-1}^{*T} \tilde{\boldsymbol{a}}_{p-1}$	$\Delta'_p = \mathbf{r}_{p-1}^{T} \tilde{\boldsymbol{b}}_{p-1}$
II	$\gamma_p = \Delta_p / \sigma_{p-1}^{\prime 2} \qquad (\gamma_p = -c_1)$	$\gamma'_p = \Delta'_p / \sigma_{p-1}^{2}$
III	$\boldsymbol{a}_p = \begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ -- \\ \tilde{\boldsymbol{b}}_{p-1} \end{bmatrix}$	$\boldsymbol{b}_p = \begin{bmatrix} \boldsymbol{b}_{p-1} \\ -- \\ 0 \end{bmatrix} - \gamma'_p \begin{bmatrix} 0 \\ -- \\ \tilde{\boldsymbol{a}}_{p-1} \end{bmatrix}$
IV	$\sigma_p^2 = \sigma_{p-1}^2 - \gamma_p \Delta'_p$	$\sigma_p^{\prime 2} = \sigma_{p-1}^{\prime 2} - \gamma'_p \Delta_p$

SIMPLIFICATION

Use: $\boldsymbol{b}_p = \boldsymbol{a}_p^*$; $\sigma_p'^2 = \sigma_p^2$

Find that: $\gamma_p' = \gamma_p^*$; $\Delta_p' = \Delta_p^*$

Then

$$\begin{aligned}
\sigma_p^2 &= \sigma_{p-1}^2 - \gamma_p \Delta_p' \\
&= \sigma_{p-1}^2 - \gamma_p (\sigma_{p-1}'^2 \gamma_p') \\
&= (1 - |\gamma_p|^2) \sigma_{p-1}^2
\end{aligned}$$

LEVINSON RECURSION ALGORITHM

INITIALIZATION

$$\boldsymbol{a}_0 = [1]\ ; \quad \mathbf{r}_0 = R_x[1]\ ; \quad \sigma_0^2 = R_x[0]$$

RECURSION $(p = 1, 2, \ldots, P)$

$$\gamma_p = \frac{\mathbf{r}_{p-1}^{*T} \tilde{\boldsymbol{a}}_{p-1}}{\sigma_{p-1}^2} \qquad \text{where} \quad \mathbf{r}_{p-1}^{*T} = [R_x^*[1]\ R_x^*[2] \cdots R_x^*[p]]$$

$$\boldsymbol{a}_p = \begin{bmatrix} \boldsymbol{a}_{p-1} \\ \\ -- \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ -- \\ \\ \tilde{\boldsymbol{a}}_{p-1}^* \end{bmatrix}$$

$$\sigma_p^2 = (1 - |\gamma_p|^2)\sigma_{p-1}^2$$

IMPLICATIONS OF LEVINSON RECURSION

1. $\boldsymbol{a}_p = \begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ -- \\ \tilde{\boldsymbol{a}}^*_{p-1} \end{bmatrix}$ *implies* $a_p^{(p)} = -\gamma_p$

2. $\sigma_p^2 = (1 - |\gamma_p|^2)\sigma_{p-1}^2$ *implies* $|\gamma_p| \leq 1$ *and* $\sigma_p^2 \leq \sigma_{p-1}^2$

<u>Note:</u> For a *regular* process $|\gamma_p| < 1$ (strict inequality).

Insert Example 8.1 here.

RECURSION FOR PREDICTION ERRORS

Expressions for the prediction errors are:

$$\varepsilon_p[n] = \sum_{i=0}^{p} a_i^{(p)*} x[n-i] = \boldsymbol{a}_p^{*T} \tilde{\boldsymbol{x}}_p[n] = \tilde{\boldsymbol{a}}_p^{*T} \boldsymbol{x}_p[n]$$

$$\varepsilon_p^b[n] = \sum_{i=0}^{p} b_i^{(p)*} x[n-p+i] = \boldsymbol{b}_p^{*T} \boldsymbol{x}_p[n] = \tilde{\boldsymbol{b}}_p^{*T} \tilde{\boldsymbol{x}}_p[n]$$

where

$$\boldsymbol{x}_p[n] = \begin{bmatrix} x[n-p] & x[n-p+1] & \cdots & x[n-1] & x[n] \end{bmatrix}^T$$

RECURSION FOR PREDICTION ERRORS (cont'd.)

From the Levinson recursion:

$$\boldsymbol{a}_p^{*T}\tilde{\boldsymbol{x}}_p[n] = \begin{bmatrix} \boldsymbol{a}_{p-1}^{*T} & \mid & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_{p-1}[n] \\ \\ ---- \\ x[n-p] \end{bmatrix}$$

$$-\gamma_p^* \begin{bmatrix} 0 & \mid & \tilde{\boldsymbol{b}}_{p-1}^{*T} \end{bmatrix} \begin{bmatrix} x[n] \\ ---- \\ \\ \tilde{\boldsymbol{x}}_{p-1}[n-1] \end{bmatrix}$$

or

$$\boxed{\varepsilon_p[n] = \varepsilon_{p-1}[n] - \gamma_p^* \varepsilon_{p-1}^b[n-1]}$$

RECURSION FOR PREDICTION ERRORS (cont'd.)

Similarly, for the backward prediction error:

$$\boldsymbol{b}_p^{*T}\boldsymbol{x}_p[n] = \begin{bmatrix} \boldsymbol{b}_{p-1}^{*T} & \mid & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{p-1}[n-1] \\ \hline x[n] \end{bmatrix}$$

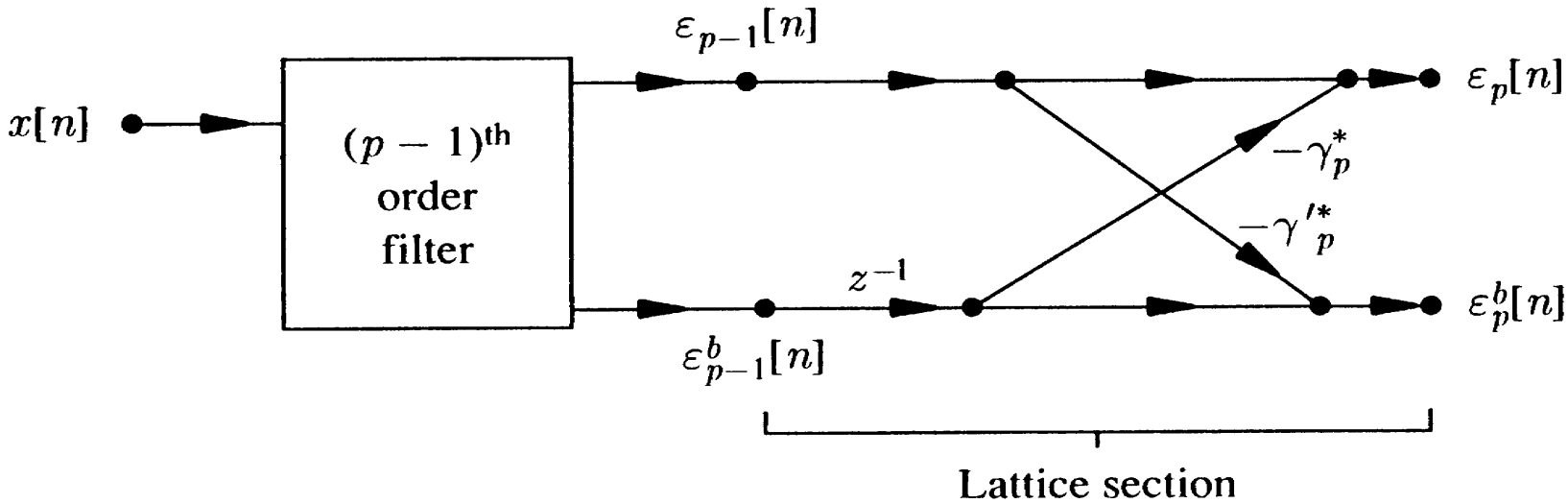
$$-\gamma_p'^{*} \begin{bmatrix} 0 & \mid & \tilde{\boldsymbol{a}}_{p-1}^{*T} \end{bmatrix} \begin{bmatrix} x[n-p] \\ \hline \boldsymbol{x}_{p-1}[n] \end{bmatrix}$$

or

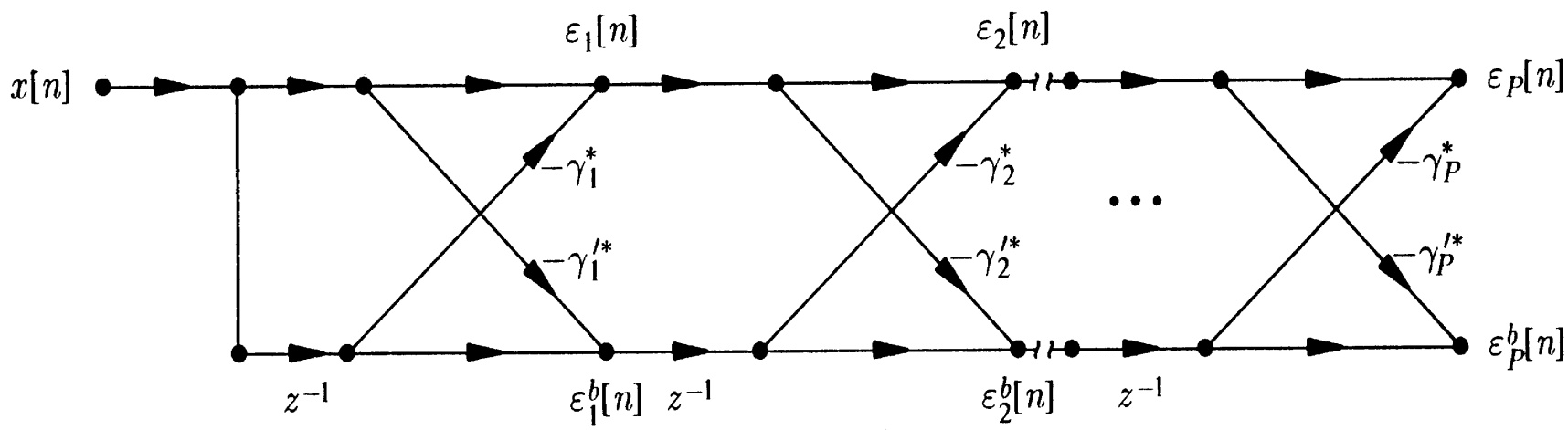
$$\boxed{\varepsilon_p^b[n] = \varepsilon_{p-1}^b[n-1] - \gamma_p'^{*}\varepsilon_{p-1}[n]}$$

PREDICTION ERROR RECURSION AND LATTICE REPRESENTATION

$$\begin{bmatrix} \varepsilon_p[n] \\ \varepsilon_p^b[n] \end{bmatrix} = \begin{bmatrix} 1 & -\gamma_p^* \\ -\gamma_p'^* & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{p-1}[n] \\ \varepsilon_{p-1}^b[n-1] \end{bmatrix}$$

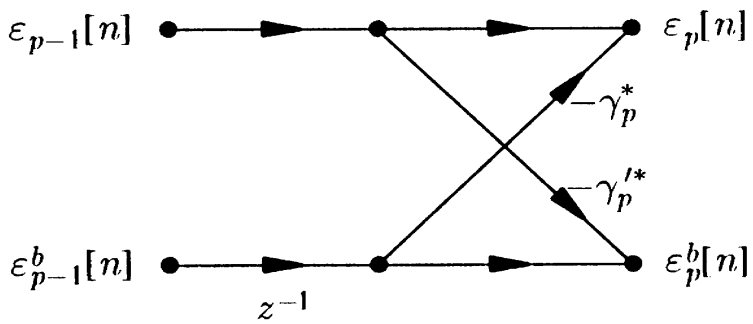


COMPLETE LATTICE REALIZATION FOR THE PREDICTION ERROR FILTER



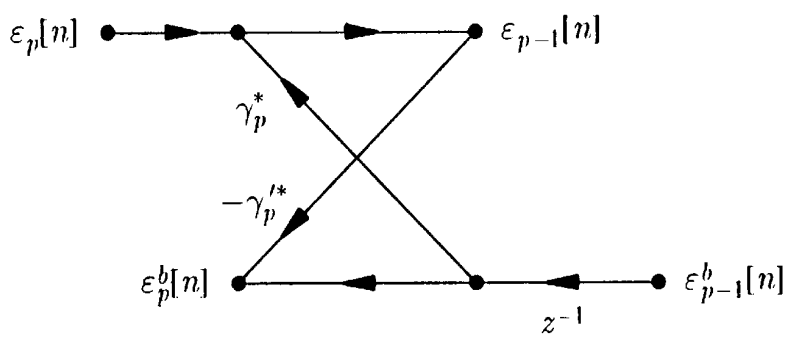
$Note: \quad \gamma_p'^* = \gamma_p$

INVERSION OF A LATTICE SECTION



$$\varepsilon_p[n] = \varepsilon_{p-1}[n] - \gamma_p^* \varepsilon_{p-1}^b[n-1]$$

$$\varepsilon_p^b[n] = \varepsilon_{p-1}^b[n-1] - \gamma_p'^* \varepsilon_{p-1}[n]$$

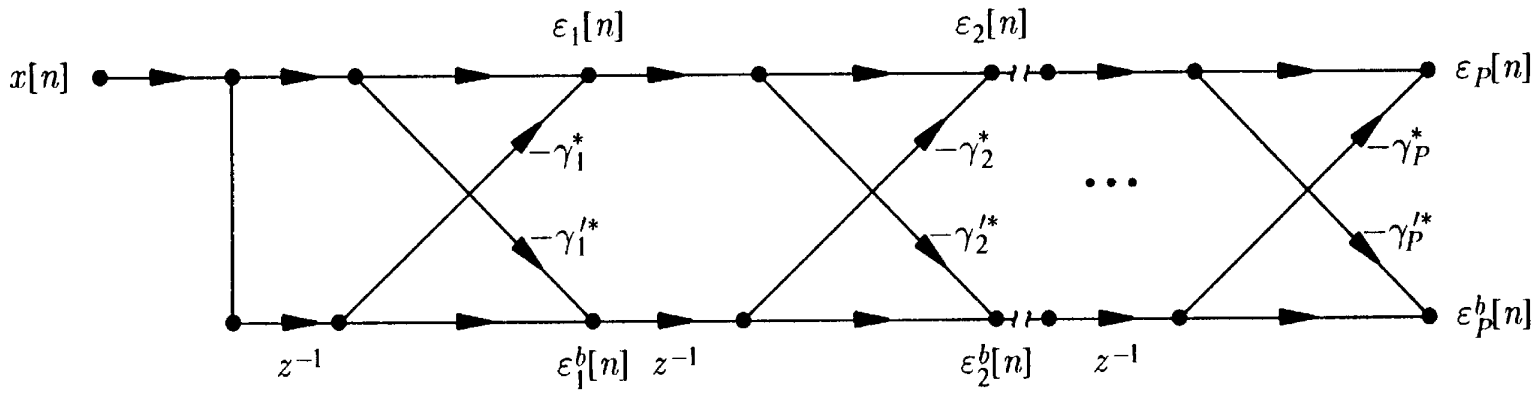


$$\varepsilon_{p-1}[n] = \varepsilon_p[n] + \gamma_p^* \varepsilon_{p-1}^b[n-1]$$

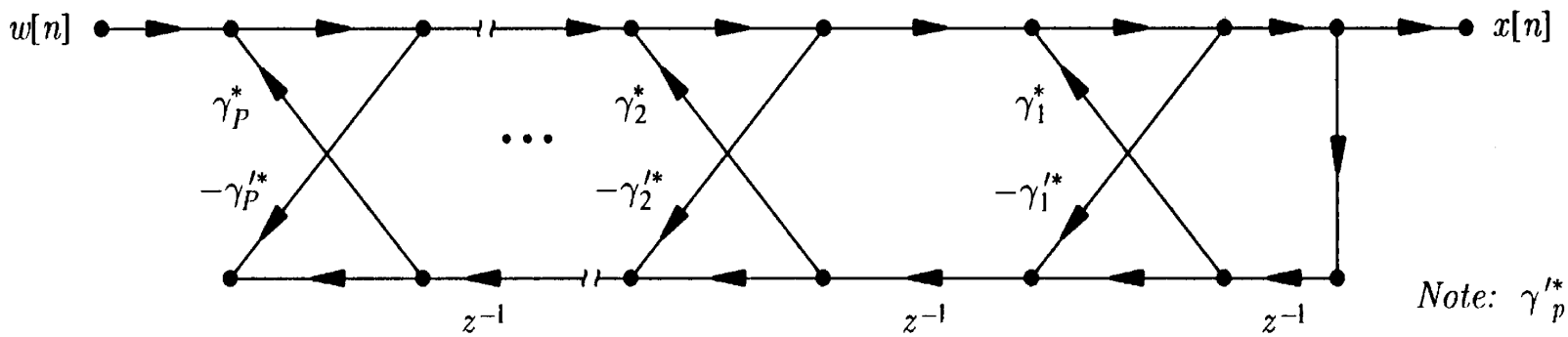
$$\varepsilon_p^b[n] = \varepsilon_{p-1}^b[n-1] - \gamma_p'^* \varepsilon_{p-1}[n]$$

LATTICE FORM OF PEF AND AR MODEL

PREDICTION ERROR FILTER



AR MODEL (ALL-POLE FILTER)



Note: $\gamma'^*_p = \gamma_p$

REVERSE-ORDER LEVINSON RECURSION

Given the vector $\boldsymbol{a}_p$ of coefficients of the Pth order filter, the reflection coefficients for order $p = P,\ P-1,\ P-2, \ldots, 1$ can be computed by:

$$\gamma_p = -a_p^{(p)}$$

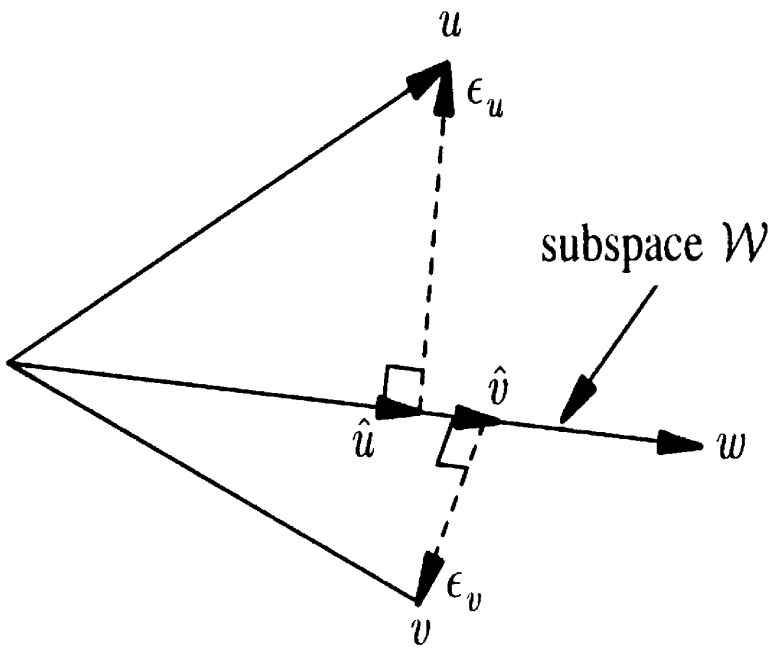
$$\begin{bmatrix} \boldsymbol{a}_{p-1} \\ -- \\ 0 \end{bmatrix} = \frac{1}{1-|\gamma_p|^2}\left[\boldsymbol{a}_p + \gamma_p \tilde{\boldsymbol{a}}_p^*\right]$$

$$\sigma_{p-1}^2 = \frac{\sigma_p^2}{1-|\gamma_p|^2} \qquad \text{(optional)}$$

Insert Example 8.2 here.

PARTIAL CORRELATION

For random variables $\{u, w_1, w_2, \ldots, w_L, v\}$ the partial correlation measures the *direct* influence of u upon v after removing the effect of the other variables.

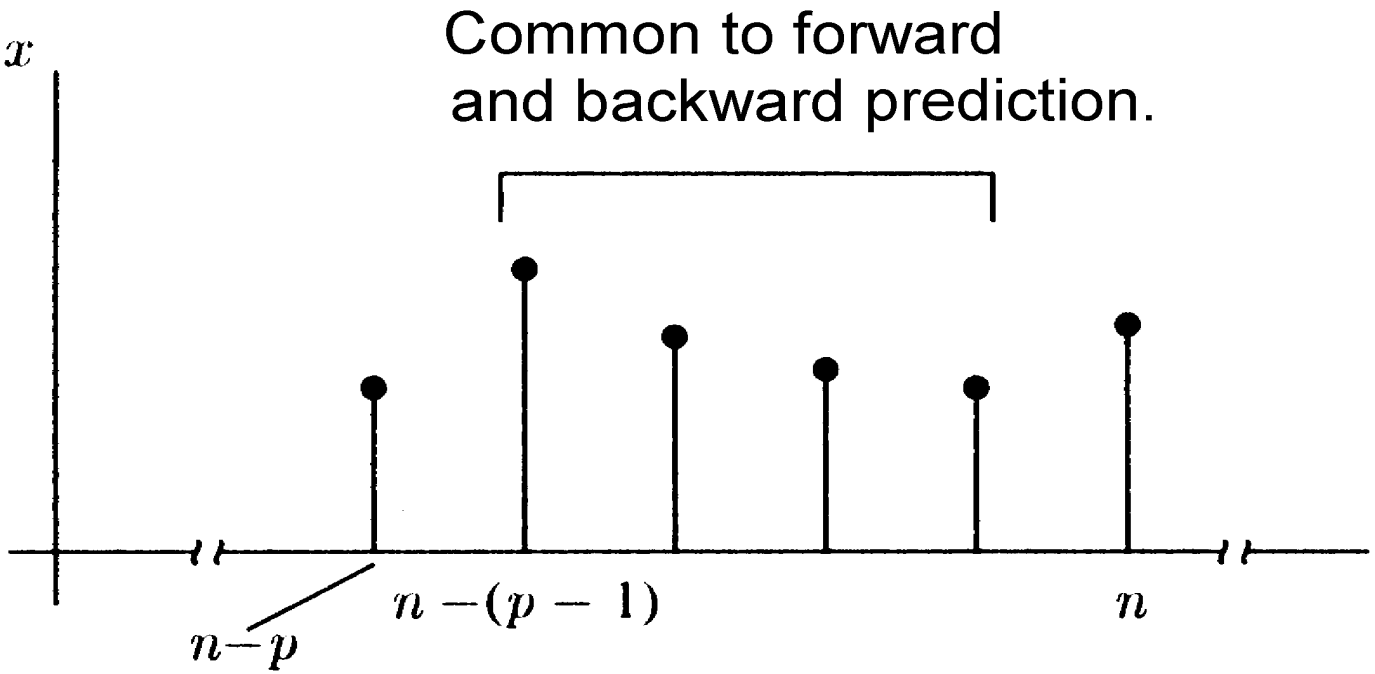


$$u = \hat{u} + \epsilon_u\,; \quad v = \hat{v} + \epsilon_v$$

$$\mathcal{E}\{uv^*\} = \mathcal{E}\{\hat{u}\hat{v}^*\} + \mathcal{E}\{\epsilon_u \epsilon_v^*\}$$

$$\text{PARCOR}\,[u; v] \stackrel{\text{def}}{=} \frac{\mathcal{E}\{\epsilon_u \epsilon_v^*\}}{\mathcal{E}\left\{|\epsilon_u|^2\right\}}$$

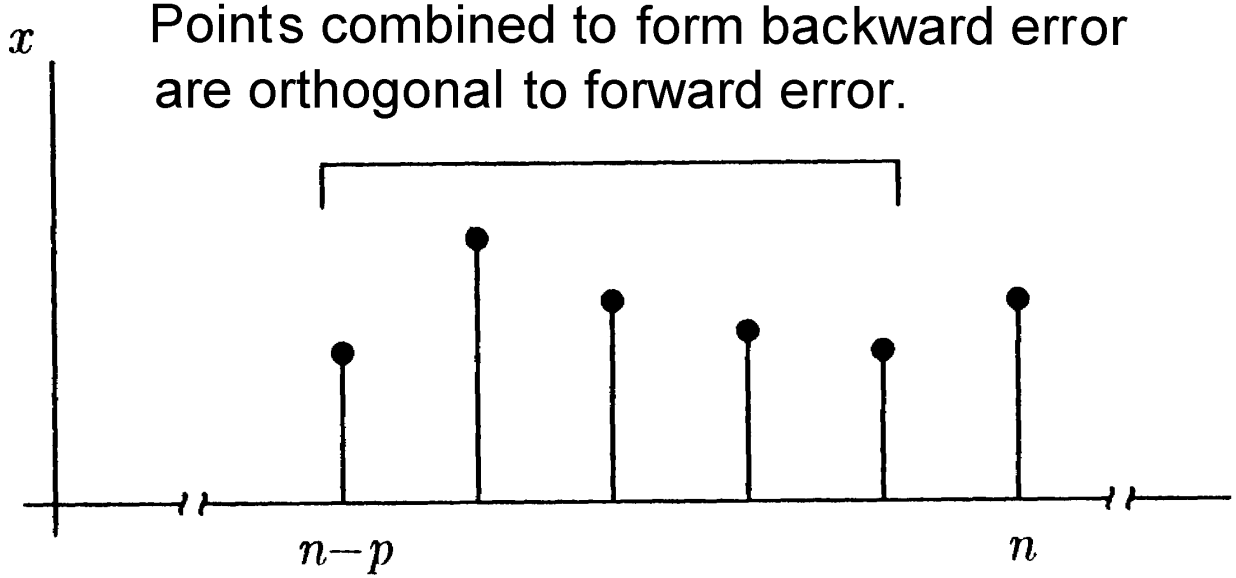
PARTIAL CORRELATION IN TIME SERIES



$$\text{PARCOR}\,[x[n-p]; x[n]] = \frac{\mathcal{E}\left\{\varepsilon^b_{p-1}[n-1]\varepsilon^*_{p-1}[n]\right\}}{\mathcal{E}\left\{|\varepsilon^b_{p-1}[n-1]|^2\right\}}$$

γ_p IS THE PARTIAL CORRELATION!

Points combined to form backward error are orthogonal to forward error.



$$\mathcal{E}\left\{\varepsilon^b_{p-1}[n-1]\varepsilon^*_p[n]\right\}=\mathcal{E}\left\{\varepsilon^b_{p-1}[n-1]\left(\varepsilon_{p-1}[n]-\gamma^*_p\varepsilon^b_{p-1}[n-1]\right)^*\right\}=0$$

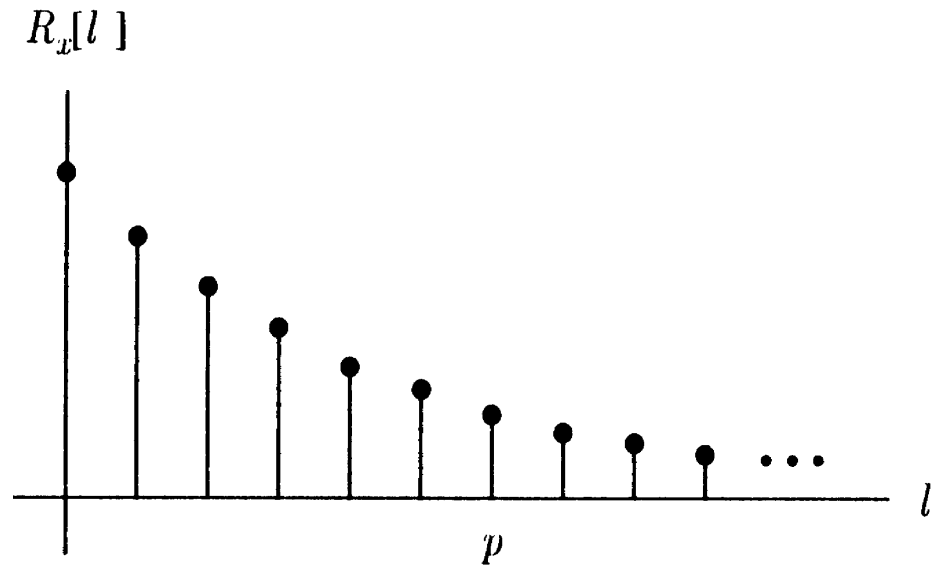
$$\gamma_p = \frac{\mathcal{E}\left\{\varepsilon^b_{p-1}[n-1]\varepsilon^*_{p-1}[n]\right\}}{\mathcal{E}\left\{|\varepsilon^b_{p-1}[n-1]|^2\right\}} = \text{PARCOR}\left[x[n-p]; x[n]\right]$$

PARTIAL CORRELATION FOR A FIRST ORDER AR PROCESS

Correlation function for the process

$$x[n] = \rho x[n-1] + w[n]$$

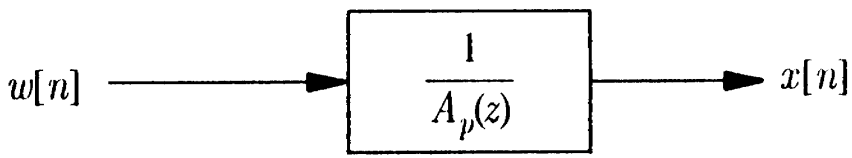
Note that $R_x[p] \neq 0 \quad p > 1$.



However: $\gamma_p = -a_p^{(p)} = \begin{cases} \rho & p = 1 \\ 0 & p > 1 \end{cases}$

MINIMUM-PHASE PROPERTY OF THE PEF (PRELIMINARIES)

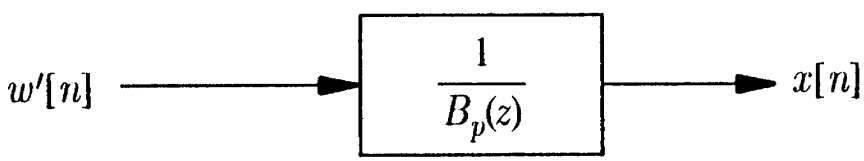
Causal AR Model



$$A_p(z) = \sum_{k=0}^{p} a_k^{(p)} z^{-k}$$

Anticausal AR Model

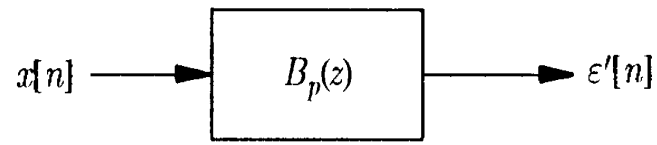
$$B_p(z) = \sum_{k=0}^{p} b_k^{(p)} z^{k}$$



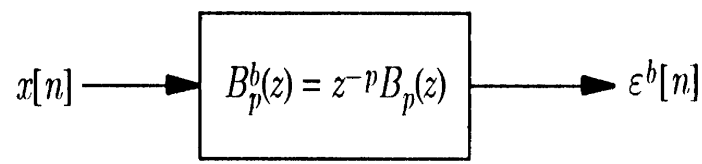
$$\boxed{B_p(z) = A_p^*(1/z^*)}$$

MINIMUM-PHASE PROPERTY OF THE PEF (PRELIMINARIES)

Recursion for the PEF



$$\varepsilon_p[n] = \varepsilon_{p-1}[n] - \gamma_p^* \varepsilon_{p-1}^b[n-1]$$

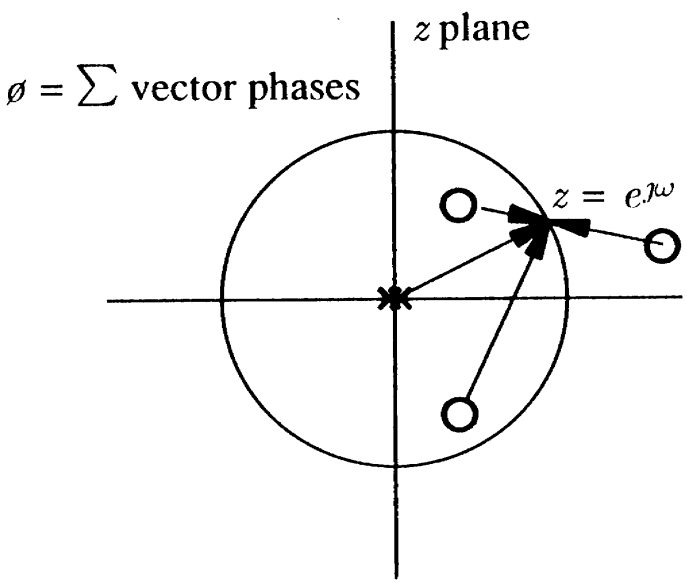


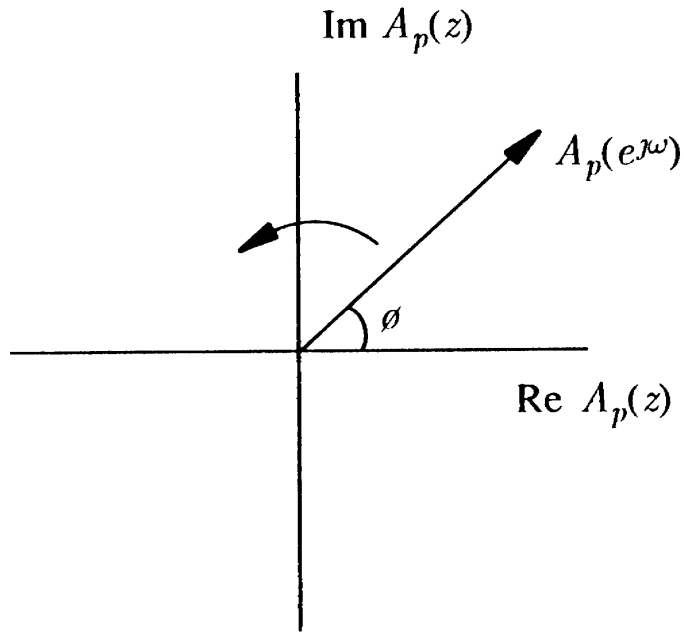
$$A_p(z) = A_{p-1}(z) - \gamma_p^* z^{-1} B_{p-1}^b(z)$$

$$\boxed{A_p(z) = A_{p-1}(z) - \gamma_p^* z^{-p} A_{p-1}^*(1/z^*)}$$

MINIMUM-PHASE PROPERTY OF THE PEF

$$A_p(z) = \frac{z^p + a_1^{(p)*} z^{p-1} + \cdots + a_p^{(p)*}}{z^p}$$





Net change of phase $= 2\pi\,[\#$ zeros $A_p(z) - \#$ poles $A_p(z)]$

MINIMUM-PHASE PROPERTY OF THE PEF (cont'd.)

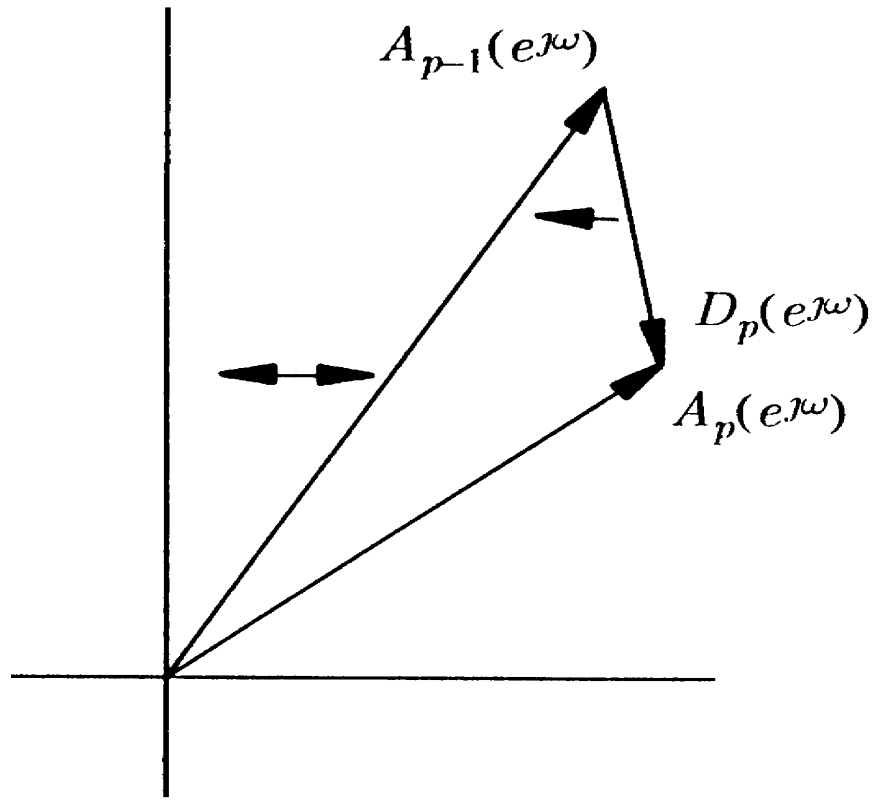
$$A_p(z) = A_{p-1}(z) \underbrace{-\gamma_p^* z^{-p} A_{p-1}^*(1/z^*)}_{D_p(z)}$$

$$|D_p(z)|_{z=e^{\jmath\omega}} = |\gamma_p| \cdot \left|A_{p-1}(e^{\jmath\omega})\right|$$

Thus for a *regular* process: $|D_p(e^{\jmath\omega})| < \left|A_{p-1}(e^{\jmath\omega})\right|$

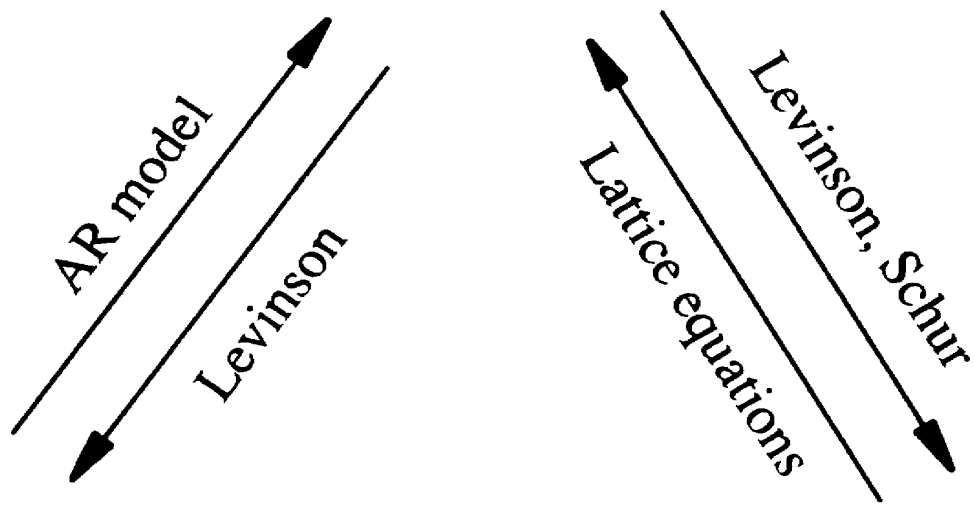
A_{p-1} minimum-phase

$\Longrightarrow A_p$ minimum-phase



EQUIVALENCES FOR AR PROCESSES

$$R_x[0]\ R_x[1] \cdots R_x[P]$$



$$\sigma_P^2\ a_1^{(P)} \cdots a_P^{(P)} \underset{\text{Levinson}}{\overset{\text{Reverse-order Levinson}}{\rightleftarrows}} \sigma_P^2 \gamma_1 \gamma_2 \cdots \gamma_P$$

SCHUR ALGORITHM

- Computes reflection coefficents and prediction error variance

- Computational requirements like Levinson ($\propto P^2$)

- Avoids computation of inner product ($\mathbf{r}_{p-1}^{*T}\tilde{\boldsymbol{a}}_{p-1}$)

  $\Rightarrow$ suitable for parallel computation

SCHUR ALGORITHM (derivation)

From the *orthogonality principle* of linear prediction:

$$\mathcal{E}\left\{\varepsilon_p[n]x^*[n-l]\right\} = \mathcal{E}\left\{\left(\varepsilon_{p-1}[n] - \gamma_p^*\varepsilon_{p-1}^b[n-1]\right)x^*[n-l]\right\} = 0$$

$$l = 1, 2, \ldots, p$$

This can be written as

$$\left[\quad g_{p-1}[l] - \gamma_p g_{p-1}^b[l-1] = 0\,; \qquad l = 1, 2, \ldots, p \quad\right]$$

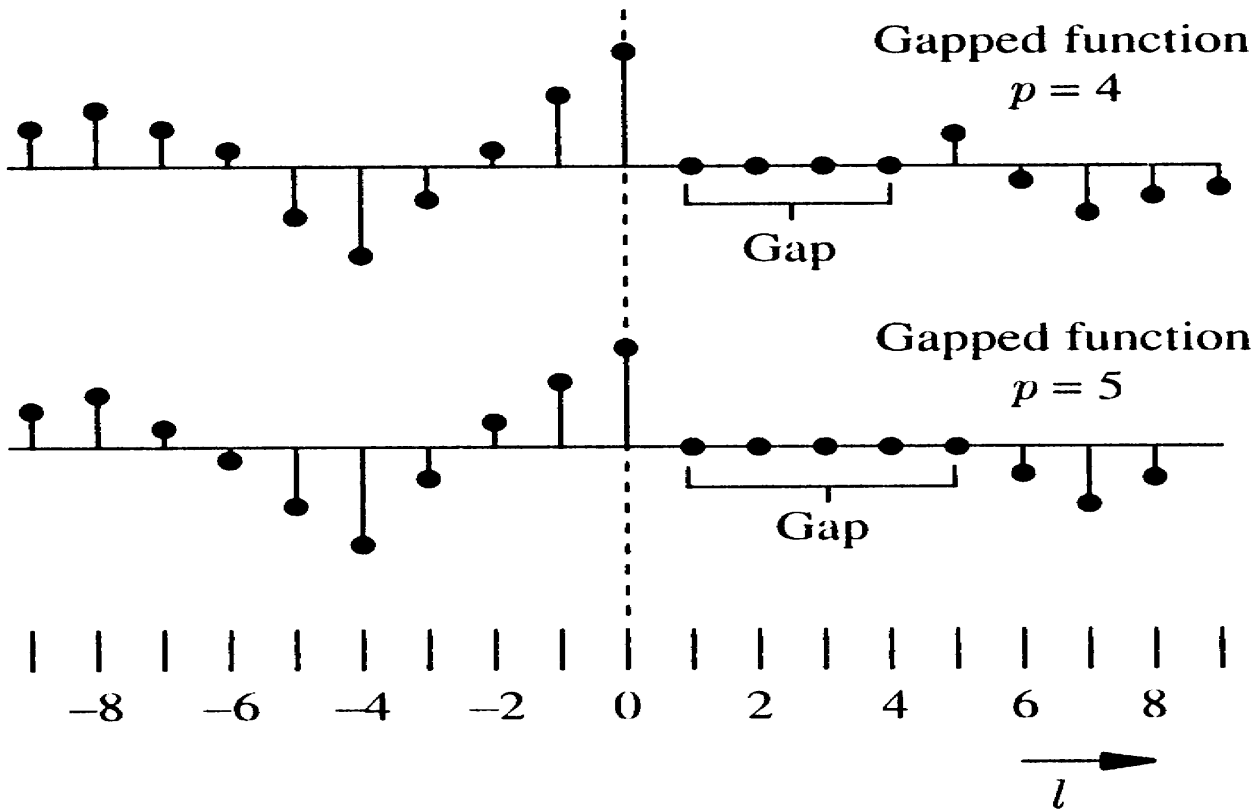
where g and g^b are "gapped functions" defined as

$$g_p[l] \stackrel{\text{def}}{=} R^*_{\varepsilon_p x}[l] = \mathcal{E}\left\{\varepsilon_p^*[n]x[n-l]\right\}$$

$$g_p^b[l] \stackrel{\text{def}}{=} R^*_{\varepsilon_p^b x}[l] = \mathcal{E}\left\{\varepsilon_p^{b*}[n]x[n-l]\right\}$$

GAPPED FUNCTION PROPERTIES

$g_p[l] = 0\,, \quad 1 \le l \le p\,; \qquad g_p^b[l] = g_p^*[p-l] = 0\,, \quad 0 \le l \le p-1$



ELEMENTS OF THE SCHUR ALGORITHM

1. $g_{p-1}[p] - \gamma_p g^b_{p-1}[p-1] = 0 \quad \Longrightarrow \quad \gamma_p = \dfrac{g_{p-1}[p]}{g^b_{p-1}[p-1]}$

2. $\sigma_p^2 = R_{\varepsilon_p x}[0] \quad \Longrightarrow \quad \sigma_p^2 = g_p[0] = g^b_p[p]$

3. (Recursion of gapped functions)

$$\begin{bmatrix} R_{\varepsilon_p x}[l] \\ R_{\varepsilon^b_p x}[l] \end{bmatrix} = \begin{bmatrix} 1 & -\gamma_p^* \\ -\gamma_p'^* & 1 \end{bmatrix} \begin{bmatrix} R_{\varepsilon_{p-1} x}[l] \\ R_{\varepsilon^b_{p-1} x}[l-1] \end{bmatrix}$$

$$\Longrightarrow \quad \begin{bmatrix} g_p[l] \\ g^b_p[l] \end{bmatrix} = \begin{bmatrix} 1 & -\gamma_p \\ -\gamma_p' & 1 \end{bmatrix} \begin{bmatrix} g_{p-1}[l] \\ g^b_{p-1}[l-1] \end{bmatrix}$$

SCHUR ALGORITHM - GENERATOR MATRIX FORM

Initialization: Begin with the "generator matrix"

$$\mathbf{G}_0 \stackrel{\text{def}}{=} \begin{bmatrix} g_0[0] & g_0[1] & g_0[2] & \cdots & g_0[P] \\ g_0^b[0] & g_0^b[1] & g_0^b[2] & \cdots & g_0^b[P] \end{bmatrix}$$

$$= \begin{bmatrix} \times & R_x^*[1] & R_x^*[2] & \cdots & R_x^*[P] \\ R_x[0] & R_x^*[1] & R_x^*[2] & \cdots & R_x^*[P] \end{bmatrix}$$

SCHUR ALGORITHM (cont'd.)

Step 1: $(p = 1)$

(a) Define

$$\gamma_1 = \frac{g_0[1]}{g_0^b[0]} = \frac{R_x^*[1]}{R_x^*[0]} \quad \text{and} \quad \gamma_1' = \gamma_1^*$$

(b) Shift lower row of $\mathbf{G}_0$ and form

$$\begin{aligned}
\mathbf{G}_1 &= \begin{bmatrix} 1 & -\gamma_1 \\ -\gamma_1' & 1 \end{bmatrix} \begin{bmatrix} \times & R_x^*[1] & R_x^*[2] & \cdots & R_x^*[P] \\ \times & R_x[0] & R_x^*[1] & \cdots & R_x^*[P-1] \end{bmatrix} \\
&= \begin{bmatrix} \times & 0 & g_1[2] & \cdots & g_1[P] \\ \times & g_1^b[1] & g_1^b[2] & \cdots & g_1^b[P] \end{bmatrix}
\end{aligned}$$

(c) Set $\sigma_1^2 = g_1^b[1]$

SCHUR ALGORITHM (cont'd.)

Step 2: $(p = 2)$

(a) Define

$$\gamma_2 = \frac{g_1[2]}{g_1^b[1]} \quad \text{and} \quad \gamma_2' = \gamma_2^*$$

(b) Shift lower row of $\mathbf{G}_1$ and form

$$\begin{aligned}
\mathbf{G}_2 &= \begin{bmatrix} 1 & -\gamma_2 \\ -\gamma_2' & 1 \end{bmatrix} \begin{bmatrix} \times & 0 & g_1[2] & \cdots & g_1[P] \\ \times & \times & g_1^b[1] & \cdots & g_1^b[P-1] \end{bmatrix} \\
&= \begin{bmatrix} \times & \times & 0 & g_2[3] & \cdots & g_2[P] \\ \times & \times & g_2^b[2] & g_2^b[3] & \cdots & g_2^b[P] \end{bmatrix}
\end{aligned}$$

(c) Set $\quad \sigma_2^2 = g_2^b[2]$

SCHUR ALGORITHM (cont'd.)

General step:

(a) Define

$$\gamma_p = \frac{g_{p-1}[p]}{g^b_{p-1}[p-1]} \quad \text{and} \quad \gamma'_p = \gamma^*_p$$

(b) Shift lower row of $\mathbf{G}_{p-1}$ and form

$$\begin{aligned}
\mathbf{G}_p &= \begin{bmatrix} 1 & -\gamma_p \\ -\gamma'_p & 1 \end{bmatrix} \begin{bmatrix} \times & \cdots & 0 & g_{p-1}[p] & \cdots & g_{p-1}[P] \\ \times & \cdots & \times & g^b_{p-1}[p-1] & \cdots & g^b_{p-1}[P-1] \end{bmatrix} \\
&= \begin{bmatrix} \times & \cdots & \times & 0 & g_p[p+1] & \cdots & g_p[P] \\ \times & \cdots & \times & g^b_p[p] & g^b_p[p+1] & \cdots & g^b_p[P] \end{bmatrix}
\end{aligned}$$

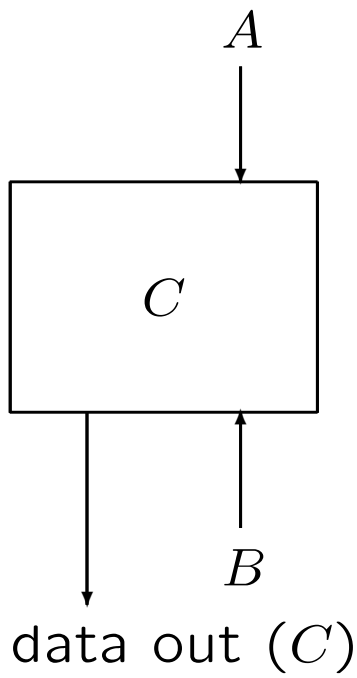
(c) Set $\sigma^2_p = g^b_p[p]$

Insert Example 8.3 here.

SCHUR SYSTOLIC IMPLEMENTATION

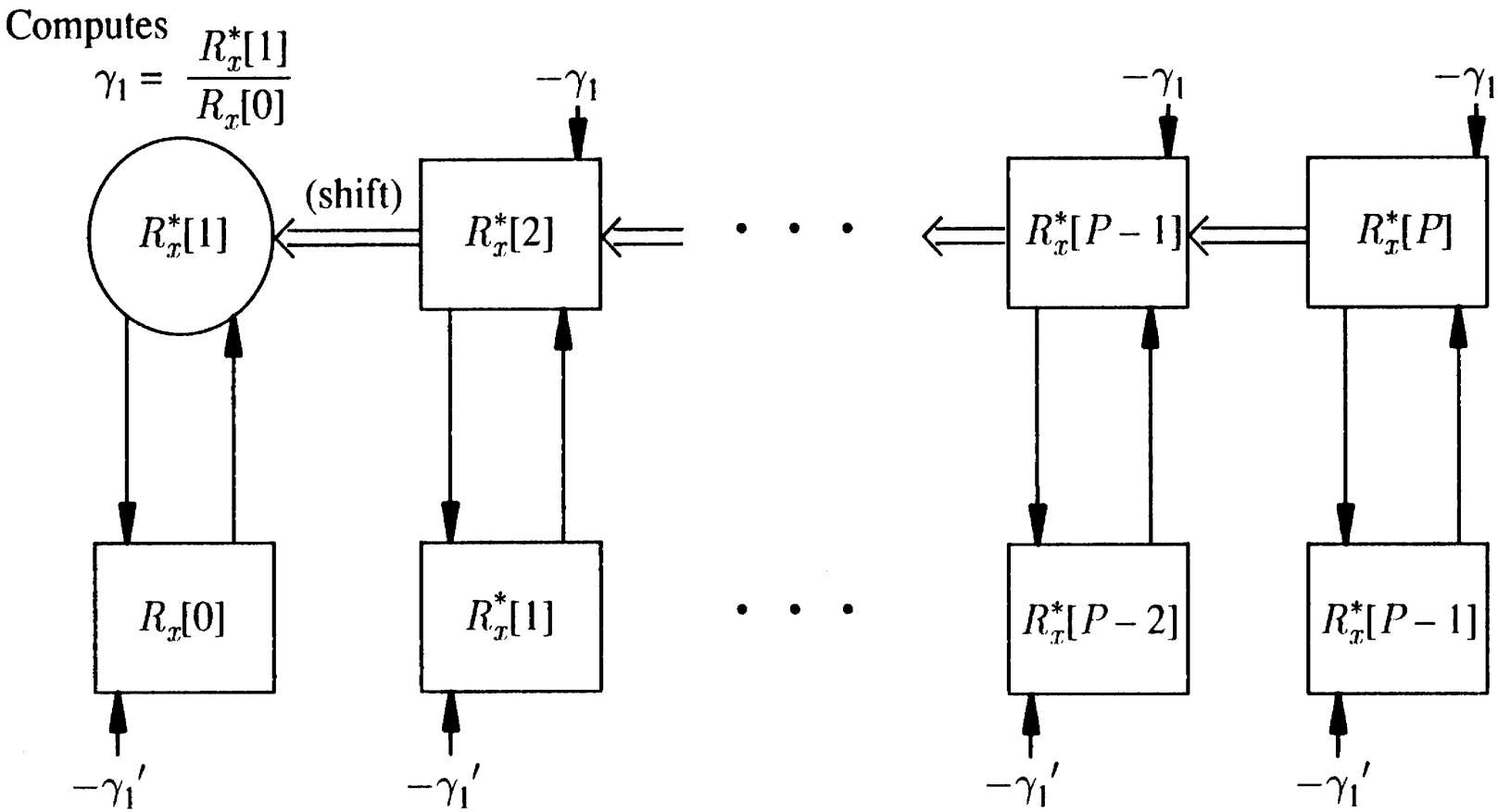
MULTIPLY/ACCUMULATE (MAC) CELL USED IN IMPLEMENTATION

$$C \leftarrow C + AB$$



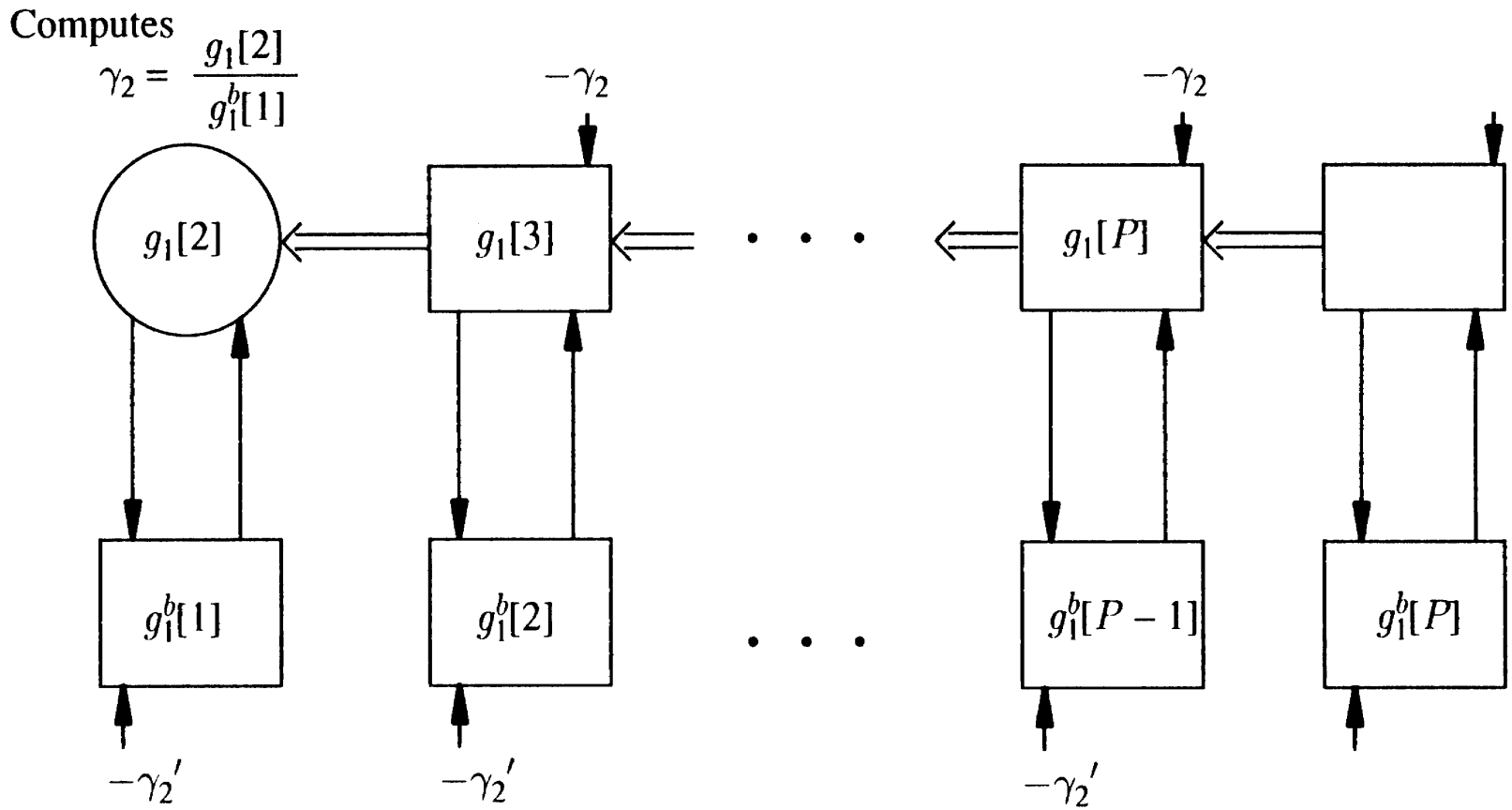
SCHUR SYSTOLIC IMPLEMENTATION (cont'd.)

FIRST CLOCK CYCLE



SCHUR SYSTOLIC IMPLEMENTATION (cont'd.)

SECOND CLOCK CYCLE



SCHUR ALGORITHM - VECTOR FORM

VECTORS USED IN THE ALGORITHM

$$\boldsymbol{g}_p^{\dagger} \stackrel{\text{def}}{=} \begin{bmatrix} g_p[p+1] \\ \vdots \\ g_p[P-1] \\ g_p[P] \end{bmatrix} \qquad \boldsymbol{g}_p^{b} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{g}_p^{b(s)} \\ \\ \\ ---- \\ g_p^b[P] \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} g_p^b[p] \\ g_p^b[p+1] \\ \vdots \\ g_p^b[P-1] \\ g_p^b[P] \end{bmatrix}$$

INITIALIZATION

$$\sigma_0^2 = R_x[0] \qquad \boldsymbol{g}_0^{\dagger} = \begin{bmatrix} R_x^*[1] \\ R_x^*[2] \\ \vdots \\ R_x^*[P] \end{bmatrix} \qquad \boldsymbol{g}_0^{b(s)} = \begin{bmatrix} R_x[0] \\ R_x^*[1] \\ \vdots \\ R_x^*[P-1] \end{bmatrix}$$

SCHUR ALGORITHM - VECTOR FORM (cont'd.)

RECURSION ($p = 1, 2, \ldots, P$)

$$\gamma_p = \frac{\text{top}(\boldsymbol{g}_{p-1}^{\dagger})}{\sigma_{p-1}^2}$$

$$\begin{bmatrix} 0 \\ \boldsymbol{g}_p^{\dagger} \end{bmatrix} = \boldsymbol{g}_{p-1}^{\dagger} - \gamma_p \boldsymbol{g}_{p-1}^{b(s)}$$

$$\boldsymbol{g}_p^b \left\{ \begin{bmatrix} \boldsymbol{g}_p^{b(s)} \\ \times \end{bmatrix} \right. = \boldsymbol{g}_{p-1}^{b(s)} - \gamma_p^* \boldsymbol{g}_{p-1}^{\dagger}$$

$$\sigma_p^2 = \text{top}(\boldsymbol{g}_p^b)$$

<u>Note:</u> Function "top" returns first element of vector.

SCHUR ALGORITHM IN MATLAB

MAIN LOOP

```
L=length(g);
for p=2:P+1;
    gamma(p) = top(g)/sigma(p-1);
    g1=g - gamma(p)*gbs;
    gb= gbs - conj(gamma(p))*g;
    g=g1(2:L);
    gbs=gb(1:L-1);
    sigma(p) = top(gb);
    L=L-1;
end;
```

Function `top(g)` defined as `g(1)`

"SPLIT" ALGORITHMS

- Reduce multiplications by an additional factor of 2
- Involve three-term recursion of symmetric variables
- Produce new lattice structures for the PEF
- Both Levinson and Schur-type algorithms exist
- Versions exist for real and complex sequences

ORDER OF PRESENTATION

- The "split" variables and three-term recursion
- Split Normal equations
- Recovery of conventional linear prediction parameters
- Split Levinson algorithms
- Split lattice representations
- Split Schur algorithms

VARIABLES AND THREE-TERM RECURSION

"SPLIT" VARIABLES

$$\boldsymbol{s}_p = K_p \boldsymbol{a}_p + K_p^* \tilde{\boldsymbol{b}}_p$$

$\boxed{\text{note symmetry:} \quad \tilde{\boldsymbol{s}}_p^* = \boldsymbol{s}_p}$

RECURSION

$$\boldsymbol{s}_{p+1} = \beta_p \begin{bmatrix} \boldsymbol{s}_p \\ \\ 0 \end{bmatrix} + \beta_p^* \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_p \end{bmatrix} - \beta_p \alpha_p \begin{bmatrix} 0 \\ \boldsymbol{s}_{p-1} \\ 0 \end{bmatrix}$$

where $\alpha_p \stackrel{\text{def}}{=} \left(K_p / K_{p-1}\right)^* \left(1 - |\gamma_p|^2\right)$ and $\beta_p \stackrel{\text{def}}{=} C_{p+1}/C_p$

with $C_p \stackrel{\text{def}}{=} K_p - \gamma_p^* K_p^*$ and constraint K_{p-1}/C_p real.

SPLIT NORMAL EQUATIONS

Starting with

$$\tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} \boldsymbol{s}_p = \tilde{\mathbf{R}}_{\boldsymbol{x}}^{(p)} (K_p \boldsymbol{a}_p + K_p^* \tilde{\boldsymbol{b}}_p)$$

it follows that

$$\begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[p] \\ R_x[-1] & R_x[0] & \cdots & R_x[p-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-p] & R_x[-p+1] & \cdots & R_x[0] \end{bmatrix} \begin{bmatrix} s_0^{(p)} \\ s_1^{(p)} \\ \vdots \\ s_p^{(p)} \end{bmatrix} = \begin{bmatrix} \tau_p \\ 0 \\ \vdots \\ 0 \\ \tau_p' \end{bmatrix}$$

where $\tau_p = \tau_p'^* = K_p \sigma_p^2$. The relation $\sigma_p^2 = \sigma_{p-1}^2 (1 - |\gamma_p|^2)$ then yields

$$\frac{\tau_p}{\tau_{p-1}} = \frac{K_p}{K_{p-1}} (1 - |\gamma_p|^2) = \alpha_p^* \qquad \Longrightarrow \qquad \boxed{\alpha_p = \frac{\tau_p^*}{\tau_{p-1}^*}}$$

RECOVERY OF CONVENTIONAL LINEAR PREDICTION PARAMETERS

RECOVERY RELATIONS

$$\begin{bmatrix} \boldsymbol{a}_P \\ \\ 0 \end{bmatrix} = \frac{1}{C_{P+1}} \boldsymbol{s}_{P+1} - \frac{1}{K_P} \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_P \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \boldsymbol{a}_P \end{bmatrix} ; \qquad \sigma_P^2 = \frac{1}{K_P} \tau_P$$

($a_p^{(P)}$ determines $a_{p+1}^{(P)}$, etc.)

The constants are found from:

$$C_{P+1} = \text{top}(\boldsymbol{s}_{P+1}) \qquad \text{and} \qquad K_P = \frac{\boldsymbol{\Sigma}(\boldsymbol{s}_P)}{\boldsymbol{\Sigma}(\boldsymbol{s}_{P+1})} C_{P+1}$$

SPLIT LEVINSON ALGORITHM ($\beta_p = 1$)

INITIALIZATION

$$\tau_0 = R_x[0]\,; \quad \boldsymbol{s}_0 = [2]\,; \quad \boldsymbol{s}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

RECURSION ($p = 1, 2, \ldots, P$)

$$\begin{aligned}
\tau_p &= \sum_{k=0}^{p} R_x[k] s_k^{(p)} \qquad \text{(compute half)} \\
\alpha_p &= \tau_p / \tau_{p-1} \\
\boldsymbol{s}_{p+1} &= \begin{bmatrix} \boldsymbol{s}_p \\ \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_p \end{bmatrix} - \alpha_p \begin{bmatrix} 0 \\ \boldsymbol{s}_{p-1} \\ 0 \end{bmatrix} \qquad \text{(compute half)}
\end{aligned}$$

SPLIT LEVINSON ALGORITHM (cont'd.)

RECOVERY RELATIONS

$$\begin{bmatrix} \boldsymbol{a}_P \\ \\ 0 \end{bmatrix} = \boldsymbol{s}_{P+1} - K'_P \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_P \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \boldsymbol{a}_P \end{bmatrix}$$

$$\sigma_P^2 = K'_P \tau_P$$

$$\text{where} \quad K'_P = \frac{1}{K_P} = \frac{\boldsymbol{\Sigma}(\boldsymbol{s}_{P+1})}{\boldsymbol{\Sigma}(\boldsymbol{s}_P)}$$

COMPLEX SPLIT LEVINSON ALGORITHM ($\beta_p = 1/\alpha_p$)

INITIALIZATION

$$\tau_0 = R_x[0]\,; \quad \boldsymbol{s}_0 = [2]\,; \quad \boldsymbol{s}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

RECURSION ($p = 1, 2, \ldots, P$)

$$\begin{aligned}
\tau_p &= \sum_{k=0}^{p} R_x[k] s_k^{(p)} \qquad \text{(compute half)} \\
\beta_p^* &= \tau_{p-1}/\tau_p \\
\boldsymbol{s}_{p+1} &= \beta_p \begin{bmatrix} \boldsymbol{s}_p \\ \\ 0 \end{bmatrix} + \beta_p^* \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_p \end{bmatrix} - \begin{bmatrix} 0 \\ \boldsymbol{s}_{p-1} \\ 0 \end{bmatrix} \qquad \text{(compute half)}
\end{aligned}$$

COMPLEX SPLIT LEVINSON ALGORITHM (cont'd.)

RECOVERY RELATIONS

$$\begin{bmatrix} \boldsymbol{a}_P \\ \\ 0 \end{bmatrix} = C'_{P+1}\boldsymbol{s}_{P+1} - K'_P \begin{bmatrix} 0 \\ \\ \boldsymbol{s}_P \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \boldsymbol{a}_P \end{bmatrix}$$

$$\sigma_P^2 = K'_P \tau_P$$

where $C'_{P+1} = \dfrac{1}{C_{P+1}} = \dfrac{1}{\text{top}(\boldsymbol{s}_{P+1})}; \qquad K'_P = \dfrac{\boldsymbol{\Sigma}(\boldsymbol{s}_{P+1})}{\boldsymbol{\Sigma}(\boldsymbol{s}_P)} C'_{P+1}$

Insert Example 8.4 here.

SPLIT ERROR RELATIONS

SPLIT ERROR

$$e_p[n] \stackrel{\text{def}}{=} \boldsymbol{s}_p^{*T}\tilde{\boldsymbol{x}}_p[n] = K_p^*\varepsilon[n] + K_p\varepsilon^b[n]$$

THREE-TERM RECURSION FOR ERROR

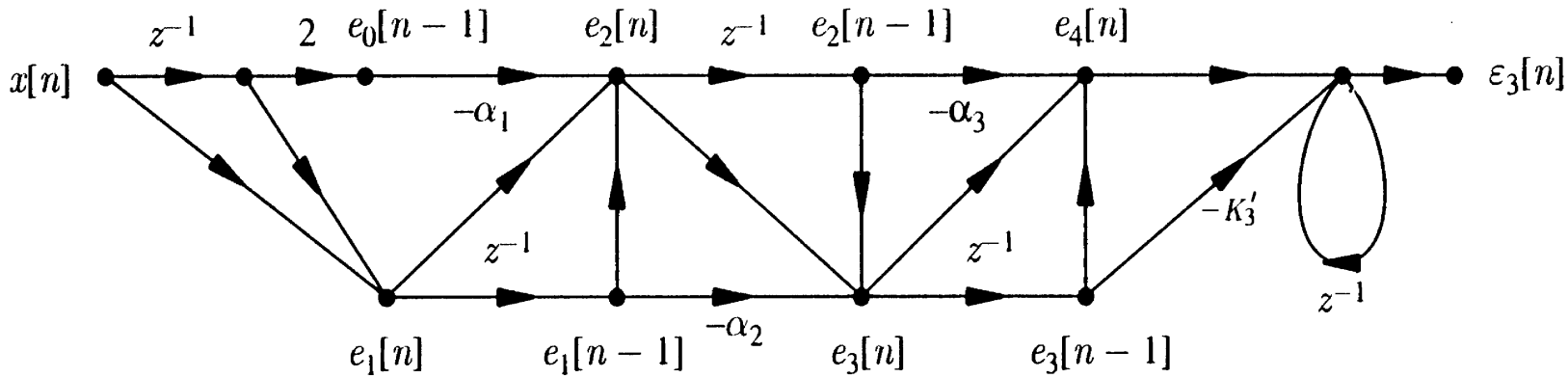
$$e_{p+1}[n] = \beta_p^* e_p[n] + \beta_p e_p[n-1] - \beta_p^*\alpha_p^* e_{p-1}[n-1]$$

RECOVERY RELATION

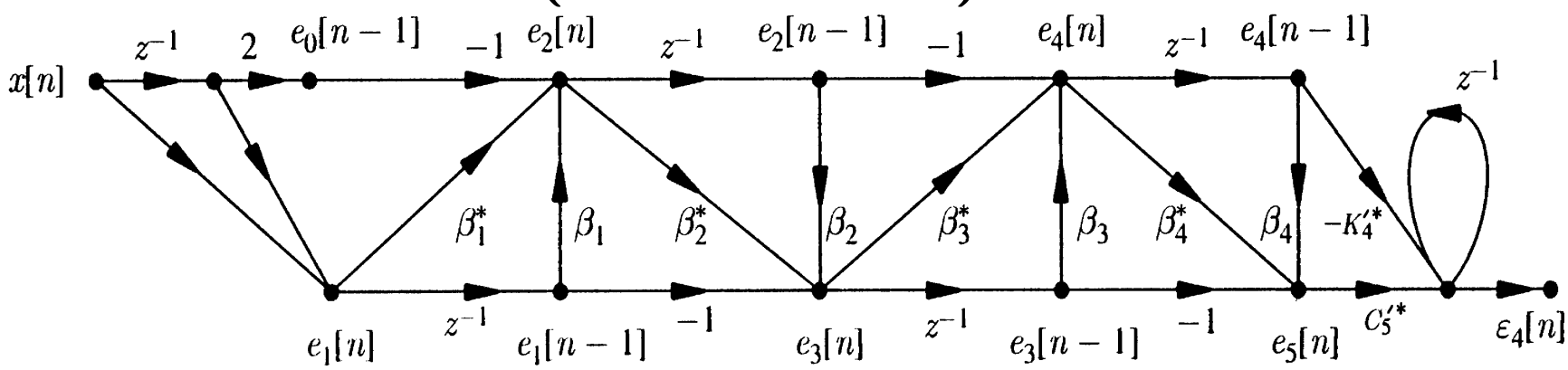
$$\varepsilon_P[n] = C_{P+1}'^* e_{P+1}[n] - K_P'^* e_P[n-1] + \varepsilon_P[n-1]$$

SPLIT LATTICE REALIZATIONS

SPLIT LATTICE PEF (ORDER $P = 3$)



COMPLEX FORM (ORDER $P = 4$)



SPLIT SCHUR RELATIONS

SPLIT SCHUR FUNCTIONS
(analogous to gapped functions)

$$v_p[l] \stackrel{\text{def}}{=} R^*_{e_p x}[l] = \left(\mathcal{E}\left\{e_p[n]x^*[n-l]\right\}\right)^*$$

THREE-TERM RECURSION
(follows from split error recursion)

$$v_{p+1}[l] = \beta_p v_p[l] + \beta_p^* v_p[l-1] - \beta_p \alpha_p v_{p-1}[l-1]$$

INITIALIZATION

$$\begin{aligned} v_0[l] &= 2R^*_x[l] && \text{for } 0 \le l \le P-1 \\ v_1[l] &= R^*_x[l] + R^*_x[l-1] && \text{for } 1 \le l \le P \end{aligned}$$

SPLIT SCHUR RELATIONS (cont'd.)

SUPPORTING RELATIONS

$$\tau_p = v_p^*[p] \quad \Longrightarrow \quad \alpha_p = \frac{\tau_p^*}{\tau_{p-1}^*} = \frac{v_p[p]}{v_{p-1}[p-1]}$$

$$C_{p+1} = \beta_p C_p \qquad K_p = \frac{\xi_p}{\xi_{p+1}} C_{p+1}$$

$$\xi_{p+1} = (\beta_p + \beta_p^*)\xi_p - \beta_p \alpha_p \xi_{p-1} \qquad \text{(where } \xi_p = \boldsymbol{\Sigma}(\boldsymbol{s}_p)\text{)}$$

RECOVERY RELATIONS

$$\gamma_p = \frac{1}{K_p}\left(K_p^* - C_p^*\right) \qquad \sigma_p^2 = \frac{\tau_p}{K_p} = \frac{v_p^*[p]}{K_p}$$

SPLIT SCHUR ALGORITHM

INITIALIZATION

$$\gamma_0 = 0\,; \qquad \boldsymbol{v}_0^{(s)} = \begin{bmatrix} R_x[0] \\ 2R_x[1] \\ \vdots \\ 2R_x[P-1] \end{bmatrix}$$

$$\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{v}_1^{(s)} \\ \\ v_1[P] \end{bmatrix} = \begin{bmatrix} R_x[1] + R_x[0] \\ R_x[2] + R_x[1] \\ \vdots \\ ---------- \\ R_x[P] + R_x[P-1] \end{bmatrix}$$

SPLIT SCHUR ALGORITHM (cont'd.)

RECURSION*

$$\alpha_p = \text{top}(\boldsymbol{v}_p)/\text{top}(\boldsymbol{v}_{p-1}^{(s)})$$

$$\gamma_p = \frac{\alpha_p}{1-\gamma_{p-1}} - 1$$

$$\sigma_p^2 = (1-\gamma_p)\text{top}(\boldsymbol{v}_p)$$

$$\boldsymbol{v}_{p+1}\left\{ \begin{bmatrix} 0 \\ \boldsymbol{v}_{p+1}^{(s)} \\ \\ v_{p+1}[P] \end{bmatrix} \right. = \begin{bmatrix} \boldsymbol{v}_p^{(s)} \\ \\ \\ v_p[P] \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \\ \boldsymbol{v}_p^{(s)} \end{bmatrix} - \alpha_p \boldsymbol{v}_{p-1}^{(s)}$$

*Simplifications shown for the computation of γ_p and σ_p^2 arise in the real algorithm only.

SPLIT SCHUR ALGORITHM IN MATLAB

MAIN LOOP

```
gamma(1)=0
L=length(v);
for p=2:P+1;
    alpha = top(v)/top(vs_1);
    gamma(p) = alpha/(1-gamma(p-1)) -1;
    sigma(p) = (1-gamma(p))*top(v);
    if p < P+1;
        vs = v(1:L-1);
        vP = v(L);
        vector = [vs,vP] + [0,vs] - alpha*vs_1;
        v = vector(2:L);
        L=L-1;
        vs_1 = vs;
    end;
end;
```

COMPLEX SPLIT SCHUR ALGORITHM

INITIALIZATION

$$C_1 = 1\,; \qquad \xi_0 = \xi_1 = 2\,; \qquad \boldsymbol{v}_0^{(s)} = \begin{bmatrix} R_x[0] \\ 2R_x^*[1] \\ \vdots \\ 2R_x^*[P-1] \end{bmatrix}$$

$$\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{v}_1^{(s)} \\ \\ v_1[P] \end{bmatrix} = \begin{bmatrix} R_x^*[1] + R_x[0] \\ R_x^*[2] + R_x^*[1] \\ \vdots \\ - - - - - - - - - - \\ R_x^*[P] + R_x^*[P-1] \end{bmatrix}$$

COMPLEX SPLIT SCHUR ALGORITHM (cont'd.)

RECURSION

$$\beta_p = \frac{\mathrm{top}(\boldsymbol{v}_{p-1}^{(s)})}{\mathrm{top}(\boldsymbol{v}_p)}\,; \qquad \xi_{p+1} = (2\mathrm{Re}\,\beta_p)\xi_p - \xi_{p-1}$$

$$C_{p+1} = \beta_p C_p\,; \qquad K_p = C_{p+1}\frac{\xi_p}{\xi_{p+1}}$$

$$\gamma_p = \frac{1}{K_p}(K_p^* - C_p^*)\,; \qquad \sigma_p^2 = \frac{1}{K_p}(\mathrm{top}(\boldsymbol{v}_p))^*$$

$$\boldsymbol{v}_{p+1}\left\{ \begin{bmatrix} 0 \\ \boldsymbol{v}_{p+1}^{(s)} \\ \\ v_{p+1}[P] \end{bmatrix} \right. = \beta_p \begin{bmatrix} \boldsymbol{v}_p^{(s)} \\ \\ \\ v_p[P] \end{bmatrix} + \beta_p^* \begin{bmatrix} 0 \\ \\ \\ \boldsymbol{v}_p^{(s)} \end{bmatrix} - \boldsymbol{v}_{p-1}^{(s)}$$

Insert Example 8.5 here.

RELATIONS BETWEEN LINEAR PREDICTION AND TRIANGULAR DECOMPOSITION

Matrix	Lower-upper decomposition	Upper-lower decomposition
$\mathbf{R}_{\boldsymbol{x}}$	$\mathbf{R}_{\boldsymbol{x}} = \mathbf{L}\mathbf{D}_L\mathbf{L}^{*T}$	$\mathbf{R}_{\boldsymbol{x}} = \mathbf{U}_1\mathbf{D}_U\mathbf{U}_1^{*T}$
$\mathbf{R}_{\boldsymbol{x}}^{-1}$	$\mathbf{R}_{\boldsymbol{x}}^{-1} = (\mathbf{U}_1^{-1})^{*T}\mathbf{D}_U^{-1}\mathbf{U}_1^{-1}$	$\mathbf{R}_{\boldsymbol{x}}^{-1} = (\mathbf{L}^{-1})^{*T}\mathbf{D}_L^{-1}\mathbf{L}^{-1}$

- Factors of $\mathbf{R}_{\boldsymbol{x}}^{-1}$ can be expressed in terms of linear prediction parameters. Factors of $\mathbf{R}_{\boldsymbol{x}}$ can be expressed in terms of Schur parameters.

STEPS IN DIAGONALIZATION OF $\mathbf{R}_{\boldsymbol{x}}$

$$\underbrace{\begin{bmatrix} R_x[0] & \cdots & R_x[P] \\ R_x[-1] & \cdots & R_x[P-1] \\ \vdots & \vdots & \vdots \\ R_x[-P] & \cdots & R_x[0] \end{bmatrix}}_{\tilde{\mathbf{R}}_{\boldsymbol{x}}} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^{(P)} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P^{(P)} & a_{P-1}^{(P-1)} & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \sigma_P^2 & \times & \cdots & \times \\ 0 & \sigma_{P-1}^2 & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_0^2 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} R_x[0] & \cdots & R_x[-P] \\ R_x[1] & \cdots & R_x[1-P] \\ \vdots & \vdots & \vdots \\ R_x[P] & \cdots & R_x[0] \end{bmatrix}}_{\mathbf{R}_{\boldsymbol{x}}} \begin{bmatrix} 1 & a_1^{(1)} & \cdots & a_P^{(P)} \\ 0 & 1 & \cdots & a_{P-1}^{(P)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ \times & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \sigma_P^2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^{(1)*} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P^{(P)*} & a_{P-1}^{(P)*} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ \times & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \sigma_P^2 \end{bmatrix} = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ ? & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ ? & ? & \cdots & \sigma_P^2 \end{bmatrix} \qquad ? = 0$$

TRIANGULAR DECOMPOSITION RELATIONS

The previous steps show that

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^{(1)*} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P^{(P)*} & a_{P-1}^{(P)*} & \cdots & 1 \end{bmatrix} \mathbf{R}_{\boldsymbol{x}} \begin{bmatrix} 1 & a_1^{(1)} & \cdots & a_P^{(P)} \\ 0 & 1 & \cdots & a_{P-1}^{(P)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_P^2 \end{bmatrix}$$

Since the decomposition $\mathbf{R}_{\boldsymbol{x}} = \mathbf{L}\mathbf{D}_L\mathbf{L}^{*T}$ or $\mathbf{L}^{-1}\mathbf{R}_{\boldsymbol{x}}(\mathbf{L}^{-1})^{*T} = \mathbf{D}_L$ is *unique*, it follows that

$$\mathbf{L}^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_1^{(1)*} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_P^{(P)*} & a_{P-1}^{(P)*} & \cdots & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{D}_L = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_P^2 \end{bmatrix}$$

GAPPED FUNCTION AND LP PARAMETERS

$$
\begin{array}{l}
\text{Impulse response terms of } p^{\text{th}}\text{-order AR model} \\
g_p[0] = \sigma_p^2 \\
g_p[p+1] = \Delta_{p+1}
\end{array}
\quad
\begin{bmatrix} g_p[-N] \\ \vdots \\ g_p[-2] \\ g_p[-1] \\ g_p[0] \\ 0 \\ \vdots \\ 0 \\ g_p[p+1] \\ g_p[p+2] \\ \vdots \\ g_p[N] \end{bmatrix}
=
\underbrace{\begin{bmatrix}
R_x[0] \cdots & & & \cdots R_x[2N] \\
\vdots \quad \ddots & & & \vdots \\
 & \cdots R_x[0] & R_x[1] \cdots \quad R_x[p+1] & \\
 & & R_x[0] \cdots \quad R_x[p] & \\
 & & \vdots \quad \ddots \quad \vdots & \\
 & & R_x[-p] \cdots \quad R_x[0] & \\
 & & & R_x[0] \cdots \\
 & & & \vdots \quad \ddots \\
\vdots & & & \vdots \\
R_x[-2N] \cdots & & & \cdots R_x[0]
\end{bmatrix}}_{(2N+1) \times (2N+1) \text{ reversed correlation matrix}}
\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

STEPS IN DECOMPOSITION OF $\mathbf{R}_{\boldsymbol{x}}$

$$\mathbf{R}_{\boldsymbol{x}} \begin{bmatrix} 1 & \tilde{\boldsymbol{a}}_1 & & \uparrow \\ 0 & \downarrow & \cdots & \tilde{\boldsymbol{a}}_P \\ \vdots & \vdots & \ddots & | \\ 0 & 0 & \cdots & \downarrow \end{bmatrix} = \begin{bmatrix} \uparrow & 0 & \cdots & 0 \\ \tilde{\boldsymbol{g}}_0 & \uparrow & & \vdots \\ | & \tilde{\boldsymbol{g}}_1 & \ddots & 0 \\ \downarrow & \downarrow & \cdots & g_P[P] \end{bmatrix}$$

$$\mathbf{R}_{\boldsymbol{x}} =$$

$$\underbrace{\begin{bmatrix} \uparrow & 0 & \cdots & 0 \\ \tilde{\boldsymbol{g}}_0/\sigma_0^2 & \uparrow & & \vdots \\ | & \tilde{\boldsymbol{g}}_1/\sigma_1^2 & \ddots & 0 \\ \downarrow & \downarrow & \cdots & g_P[P]/\sigma_P^2 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_P^2 \end{bmatrix}}_{\mathbf{D}_L} \underbrace{\begin{bmatrix} 1 & \tilde{\boldsymbol{a}}_1 & & \uparrow \\ 0 & \downarrow & \cdots & \tilde{\boldsymbol{a}}_P \\ \vdots & \vdots & \ddots & | \\ 0 & 0 & \cdots & \downarrow \end{bmatrix}^{-1}}_{\mathbf{L}^{*T}}$$

SUMMARY OF RELATIONS IN $\mathrm{R}_{\boldsymbol{x}} = \mathbf{L}\mathbf{D}_L\mathbf{L}^{*T}$

$$\mathbf{L} = \begin{bmatrix} \uparrow & 0 & \cdots & 0 \\ \tilde{\boldsymbol{g}}_0/\sigma_0^2 & \uparrow & & \vdots \\ | & \tilde{\boldsymbol{g}}_1/\sigma_1^2 & & \vdots \\ | & | & \ddots & 0 \\ \downarrow & \downarrow & \cdots & g_P[P]/\sigma_P^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \tilde{\boldsymbol{a}}_1^{*T} & \rightarrow & \cdots & 0 \\ & \vdots & \ddots & \vdots \\ \leftarrow & \tilde{\boldsymbol{a}}_{P-1}^{*T} & \rightarrow & 0 \\ \leftarrow & -- & \tilde{\boldsymbol{a}}_P^{*T} & \rightarrow \end{bmatrix}^{-1}$$

$$\mathbf{D}_L = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_P^2 \end{bmatrix}$$

SUMMARY OF RELATIONS IN $\mathbf{R}_{\boldsymbol{x}} = \mathbf{U}_1 \mathbf{D}_U \mathbf{U}_1^{*T}$

$$\mathbf{U}_1 = \begin{bmatrix} g_P^b[P]/\sigma_P'^2 & & \uparrow & \uparrow \\ 0 & \cdots & \tilde{\boldsymbol{g}}_1^b/\sigma_1'^2 & | \\ \vdots & \ddots & \downarrow & \tilde{\boldsymbol{g}}_0^b/\sigma_0'^2 \\ 0 & \cdots & 0 & \downarrow \end{bmatrix} = \begin{bmatrix} \leftarrow & \boldsymbol{b}_P^{*T} & -- & \rightarrow \\ 0 & \leftarrow & \boldsymbol{b}_{P-1}^{*T} & \rightarrow \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \leftarrow & \boldsymbol{b}_1^{*T} \\ 0 & 0 & \cdots & 1 \end{bmatrix}^{-1}$$

$$\mathbf{D}_U = \begin{bmatrix} \sigma_P'^2 & 0 & \cdots & 0 \\ 0 & \sigma_{P-1}'^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_0'^2 \end{bmatrix}$$

In addition:

$$\mathbf{U}_1 = \tilde{\mathbf{L}}^* \quad \text{and} \quad \mathbf{D}_U = \tilde{\mathbf{D}}_L$$

All factors can thus be written in terms of either the forward or the backward parameters.

IMPLICATION OF THE RELATIONS

- For a *regular* process $\sigma_p^2 > 0$ for all values of p.

- This implies that $\mathbf{R}_{\boldsymbol{x}}$ is *strictly* positive definite.

- For a "predictable" process there is some value P_{O} such that $\sigma_p^2 = 0$ for $p \geq P_{\mathrm{O}}$.

- This implies that $\mathbf{R}_{\boldsymbol{x}}$ is only positive semidefinite.

LATTICE FORM FOR WIENER FILTER

Start with the Wiener-Hopf equation

$$\mathbf{R}_{\boldsymbol{x}}\,\mathbf{h} = \tilde{\mathbf{r}}_{d\boldsymbol{x}} \quad \Longleftrightarrow \quad \mathbf{R}_{\boldsymbol{x}}\tilde{\mathbf{h}}^* = \mathbf{r}^*_{d\boldsymbol{x}}$$

and equation for the filter output

$$\widehat{d}[n] = \mathbf{h}^T\tilde{\boldsymbol{x}}[n] = \tilde{\mathbf{h}}^T\boldsymbol{x}[n]$$

Solve for filter parameters and insert the result:

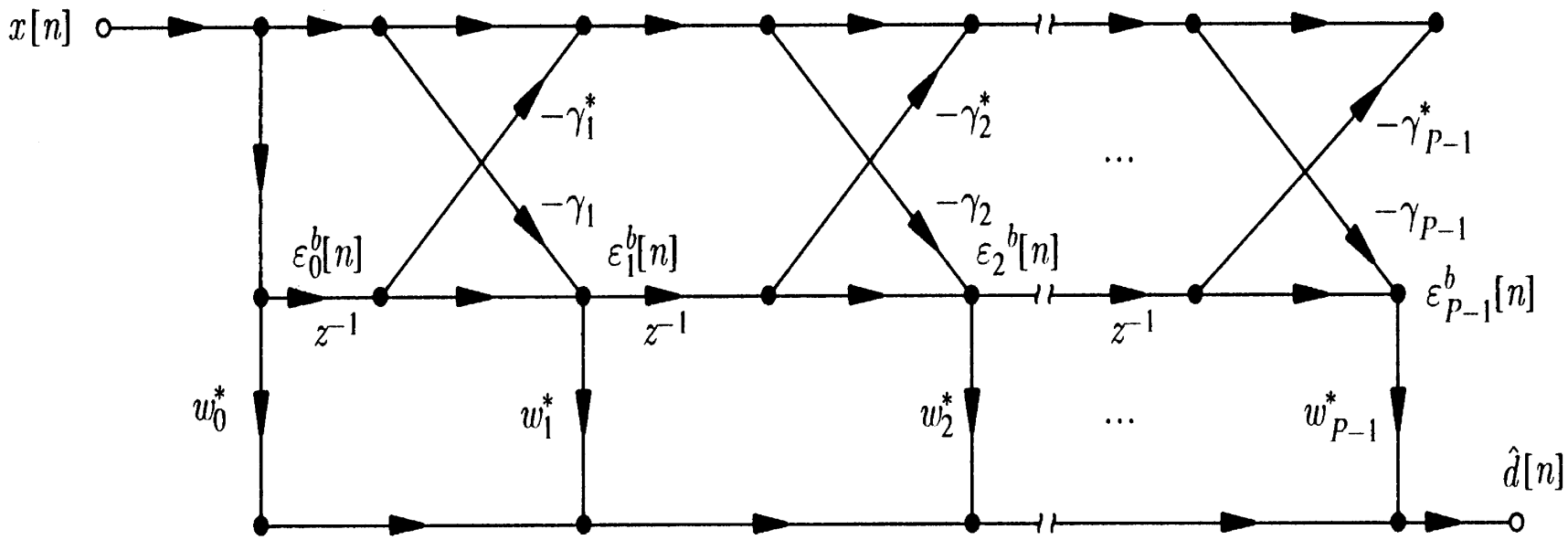
$$\widehat{d}[n] = \left(\mathbf{R}_{\boldsymbol{x}}^{-1}\mathbf{r}^*_{d\boldsymbol{x}}\right)^{*T}\boldsymbol{x}[n] = \underbrace{\mathbf{r}^T_{d\boldsymbol{x}}(\mathbf{U}_1^{-1})^{*T}\mathbf{D}_U^{-1}}_{\tilde{\mathbf{w}}^{*T}}\underbrace{\mathbf{U}_1^{-1}\boldsymbol{x}[n]}_{\boldsymbol{\varepsilon}^{\boldsymbol{b}}[n]}$$

LATTICE FORM FOR WIENER FILTER (cont'd.)

The Wiener filter output is expressed as

$$\hat{d}[n] = \tilde{\mathbf{w}}^{*T}\boldsymbol{\varepsilon}^{\boldsymbol{b}}[n] \qquad \text{with} \quad \tilde{\mathbf{w}} = \mathbf{D}_U^{-1}\mathbf{U}_1^{-1}\mathbf{r}_{d\boldsymbol{x}}^*$$

and

$$\boldsymbol{\varepsilon}^{\boldsymbol{b}}[n] = \begin{bmatrix} \varepsilon_{P-1}^b[n] \\ \varepsilon_{P-2}^b[n] \\ \vdots \\ \varepsilon_0^b[n] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & b_1^{(P-1)*} & \cdots & b_{P-1}^{(P-1)*} \\ 0 & 1 & \cdots & b_{P-2}^{(P-2)*} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\mathbf{U}_1^{-1}} \begin{bmatrix} x[n-P+1] \\ \vdots \\ x[n-1] \\ x[n] \end{bmatrix}$$

WIENER FILTER IN LATTICE FORM