LRA
DSP

Prediction

Forward and Backward Linear Prediction

Forward Linear Prediction

- Given: WSS random process whose last *p* values are known
- Problem: predict next value

$$\hat{x}(n) = -\sum_{k=1}^{p} a_p(k)\, x(n-k)$$

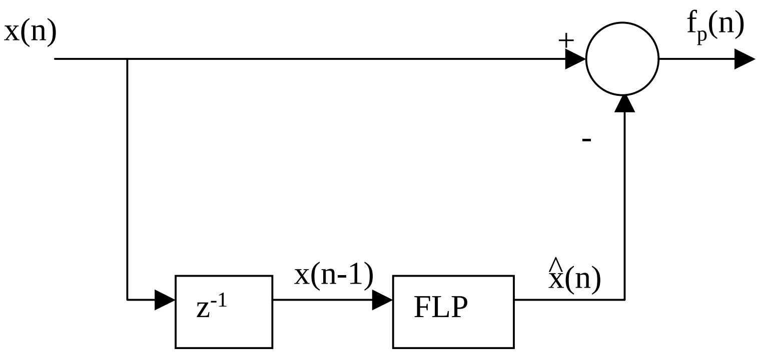
  - $\{a_p(k)\}$: (forward) prediction coefficients
  - Predictor as a linear filter
- Forward linear prediction (FLP) error:

$$f_p(n) = x(n) - \hat{x}(n)$$

$$= x(n) + \sum_{k=1}^{p} a_p(k)\, x(n-k)$$

FLP Error Filter

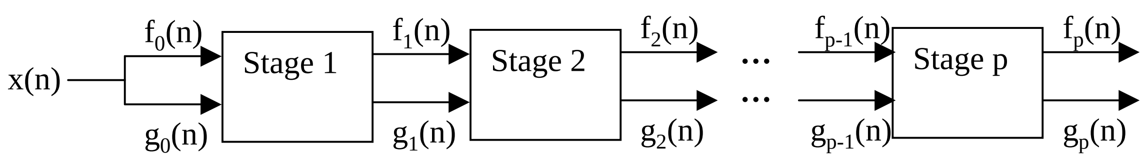
- Prediction-error filter



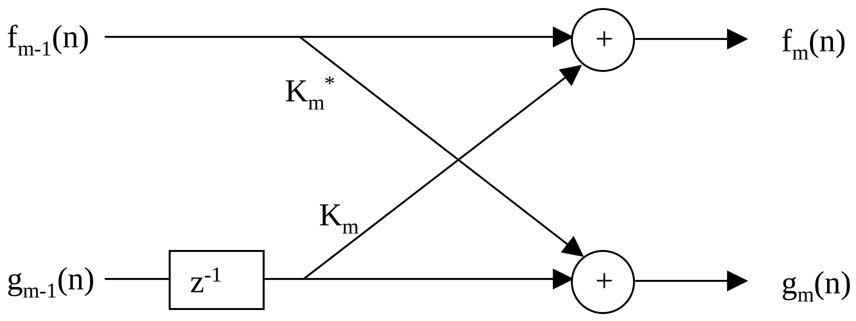
- Realisations
  - Direct-form implementation: $A_p(z) = \sum_{k=0}^{p} a_p(k)\, z^{-k}$
  - Lattice filter

FLP Lattice Filter

- Implementation:



- Individual stage (both FLP and BLP):



LRA DSP

Order-Recursive Equations

- Order-recursive equations:

$$f_0(n) = g_0(n) = x(n)$$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1), \quad m = 1,2,\ldots,p$$

$$g_m(n) = K_m^* f_{m-1}(n) + g_{m-1}(n-1), \quad m = 1,2,\ldots,p$$

  - K_m: reflection coefficients
  - g_m(n): backward prediction error

FLP Normal Equations

- Output of *p*-stage filter:

$$f_p(n) = \sum_{k=0}^{p} a_p(k)\, x(n-k), \qquad a_p(0) = 1$$

  - z-transformation:

$$F_p(z) = A_p(z)\, X(z) \;\Rightarrow\; A_p(z) = \frac{F_p(z)}{F_0(z)}$$

- MSE for FLP:

$$\varepsilon_{FLP}^2 = E[|\, f_p\,|^2] = \gamma_{xx}(0) + 2\,\mathrm{Re}\left[\sum_{k=1}^{p} a_p^*(k)\,\gamma_{xx}(k)\right] + \sum_{k=1}^{p}\sum_{l=1}^{p} a_p^*(l)\, a_p(k)\,\gamma_{xx}(l-k)$$

- MMSE: normal equations for FLP cffs.:

$$\gamma_{xx}(l) = -\sum_{k=1}^{p} a_p(k)\,\gamma_{xx}(l-k), \qquad l = 1,2,\ldots, p$$

Backward Linear Prediction

- One-step BLP of order *p*:

$$\hat{x}(n-p) = -\sum_{k=0}^{p-1} b_p(k)\, x(n-k)$$

- Backward prediction error

$$\begin{aligned} g_p(n) &= x(n-p) - \hat{x}(n-p) \\ &= x(n-p) + \sum_{k=0}^{p-1} b_p(k)\, x(n-k) \\ &= \sum_{k=0}^{p} b_p(k)\, x(n-k), \qquad b_p(p) = 1 \end{aligned}$$

BLP to FLP Relationship

- One-step BLP of order *p*:

$$b_p(k) = a_p^*(p-k), \quad k = 0,1.,\dots, p$$

- z-domain:

$$B_p(z) = \sum_{k=0}^{p} b_p(k) z^{-k} = \sum_{k=0}^{p} a_p^*(p-k) z^{-k} = z^{-p} \sum_{k=0}^{p} a_p^*(k) z^{+k}$$
$$= z^{-p} A_p^*(z^{-1})$$

i.e., zeroes of FIR BLP are complex conjugate reciprocals of zeroes of FIR FLP

Lattice Filter Design

- Reverse polynomial cffs. (z-domain):

$$G_p(z) = B_p(z)\,X(z) \quad \Rightarrow \quad B_p(z) = \frac{G_p(z)}{G_0(z)}$$

- Order-recursive equation cffs. (z-domain):

$$F_0(z) = G_0(z) = X(z)$$

$$F_m(z) = F_{m-1}(z) + K_m z^{-1} G_{m-1}(z), \qquad m = 1,2,\ldots, p$$

$$G_m(z) = K_m^* F_{m-1}(z) + z^{-1} G_{m-1}(z), \qquad m = 1,2,\ldots, p$$

hence

$$A_0(z) = B_0(z) = 1$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z), \qquad m = 1,2,\ldots, p$$

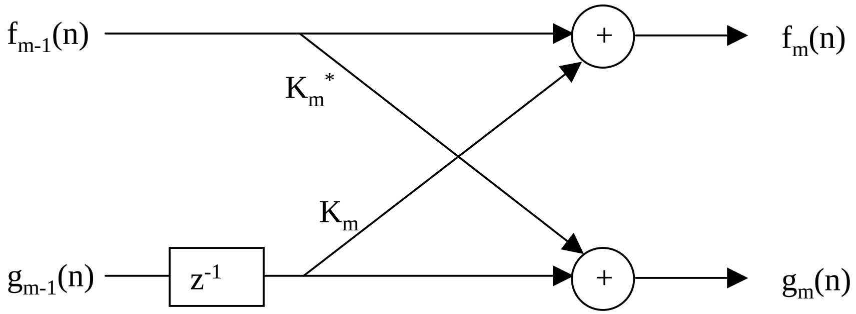
$$B_m(z) = K_m^* A_{m-1}(z) + z^{-1} B_{m-1}(z), \qquad m = 1,2,\ldots, p$$

Lattice Filter Design

- Linear prediction transfer matrix:

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m z^{-1} \\ K_m^* & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} A_{m-1}(z) \\ B_{m-1}(z) \end{bmatrix}$$

  - Permits design of lattice-form reflection cffs. from direct-form FIR filter cffs.



Solution of Normal Equations

- Two computationally efficient methods:
  - Levinson–Durbin algorithm
    - For serial processing
    - Complexity:
      - O(p^2) operations for prediction cffs. (serial),
      - O(p ln p) (semi-parallel for multiplications & additions)
  - Schur algorithm
    - For parallel processing
      - pipeline architecture: each stage has two parallel processing blocks
    - Complexity: O(p)
- Both methods exploit Toeplitz structure of γ_{xx}

Properties of FLP/BLP Filters

- FLP filter is minimum-phase
- BLP filter is maximum-phase
- Whitening: FLP whitens input WSS process $x(n)$
- Orthogonality: BLP errors from different stages are orthogonal