Linear Prediction: Lattice filters

Overview

- New AR parametrization: Reflection coefficients;
- Fast computation of prediction errors;
- Direct and Inverse Lattice filters;
- Burg lattice parameter estimator;
- Gradient Adaptive Lattice filters;

Lattice Predictors

- **Order -Update Recursions for Prediction errors**

  Since the predictors obey the recursive–in–order equations

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

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

  it is natural that prediction errors can be expressed in recursive–in–order forms. These forms results considering the recursions for the vector $\underline{u}_{m+1}(n)$

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

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

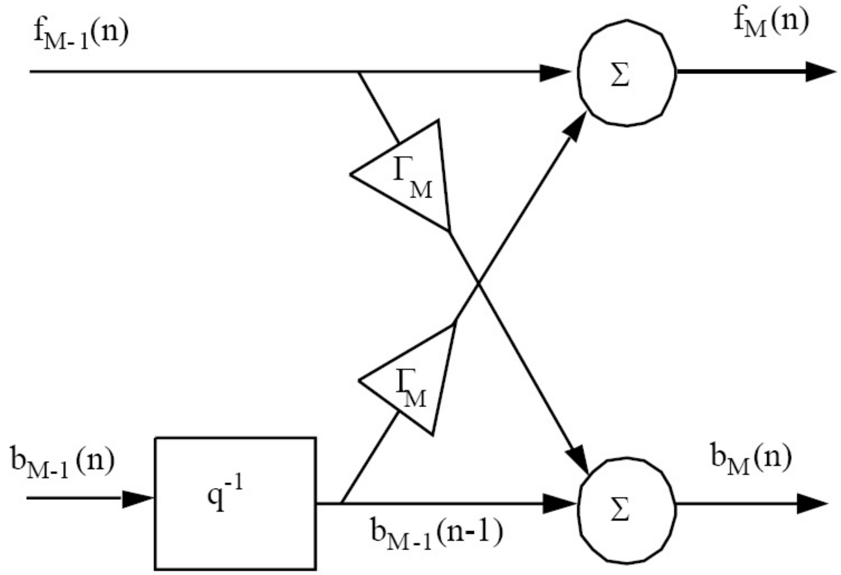
  Combining the equations we obtain

$$\begin{aligned} f_m(n) &= \underline{a}_m^T \underline{u}_{m+1}(n) = \begin{bmatrix} \underline{a}_{m-1}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_m(n) \\ u(n-m) \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 & (\underline{a}_{m-1}^B)^T \end{bmatrix} \begin{bmatrix} u(n) \\ \underline{u}_m(n-1) \end{bmatrix} = \\ &= \underline{a}_{m-1}^T \underline{u}_m(n) + \Gamma_m (\underline{a}_{m-1}^B)^T \underline{u}_m(n-1) = \\ &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \end{aligned}$$

$$
\begin{aligned}
b_m(n) &= (\underline{a}_m^B)^T \underline{u}_{m+1}(n) = \begin{bmatrix} 0 & (\underline{a}_{m-1}^B)^T \end{bmatrix} \begin{bmatrix} u(n) \\ \underline{u}_m(n-1) \end{bmatrix} + \Gamma_m \begin{bmatrix} (\underline{a}_{m-1})^T & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_m(n) \\ u(n-m) \end{bmatrix} = \\
&= (\underline{a}_{m-1}^B)^T \underline{u}_m(n-1) + \Gamma_m (\underline{a}_{m-1})^T \underline{u}_m(n) \\
&= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n)
\end{aligned}
$$

The order recursions of the errors can be represented as

$$
\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n-1) \end{bmatrix}
$$



$$\boxed{\begin{aligned} f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n) \end{aligned}}$$

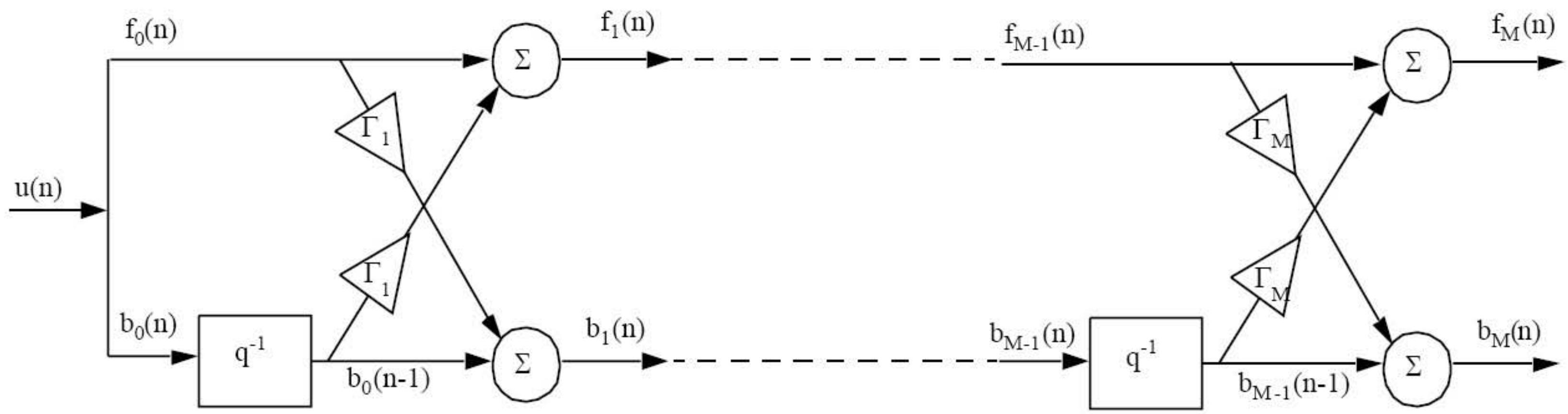
Using the time shifting operator q^{-1}, the prediction error recursions are given by

$$\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m q^{-1} \\ \Gamma_m & q^{-1} \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n) \end{bmatrix}$$

which can now be iterated for $m = 1, 2, \ldots, M$ to obtain

$$\begin{aligned} \begin{bmatrix} f_M(n) \\ b_M(n) \end{bmatrix} &= \begin{bmatrix} 1 & \Gamma_M q^{-1} \\ \Gamma_M & q^{-1} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{M-1} q^{-1} \\ \Gamma_{M-1} & q^{-1} \end{bmatrix} \cdots \begin{bmatrix} 1 & \Gamma_1 q^{-1} \\ \Gamma_1 & q^{-1} \end{bmatrix} \begin{bmatrix} f_0(n) \\ b_0(n) \end{bmatrix} \\ &= \begin{bmatrix} 1 & \Gamma_M q^{-1} \\ \Gamma_M & q^{-1} \end{bmatrix} \begin{bmatrix} 1 & \Gamma_{M-1} q^{-1} \\ \Gamma_{M-1} & q^{-1} \end{bmatrix} \cdots \begin{bmatrix} 1 & \Gamma_1 q^{-1} \\ \Gamma_1 & q^{-1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(n) \end{aligned}$$

Having available the reflexion coefficients, all prediction errors of order $m = 1, \ldots, M$ can be computed using the Lattice predictor, in $2M$ additions and $2M$ multiplications.



LATTICE PREDICTOR OF ORDER M

Some characteristics of the Lattice predictor:

1. It is the most efficient structure for generating simultaneously the forward and backward prediction errors.
2. The lattice structure is modular: increasing the order of the filter requires adding only one extra module, leaving all other modules the same.
3. The various stages of a lattice are decoupled from each other in the following sense: The memory of the lattice (storing $b_0(n-1), \ldots, b_{M-1}(n-1)$) contains orthogonal variables, thus the information contained in $u(n)$ is splitted in M pieces, which reduces gradually the redundancy of the signal.
4. The similar structure of the lattice filter stages makes the filter suitable for VLSI implementation.

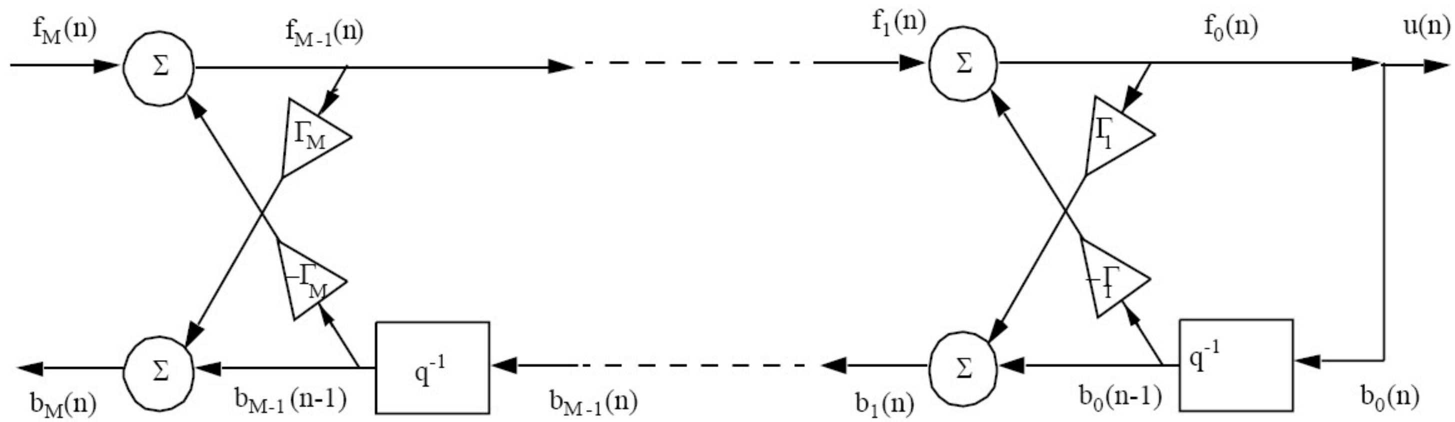
- **Lattice Inverse filters** The basic equations for one stage of the lattice are

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= \Gamma_m f_{m-1}(n) + b_{m-1}(n-1) \end{aligned} \tag{1}$$

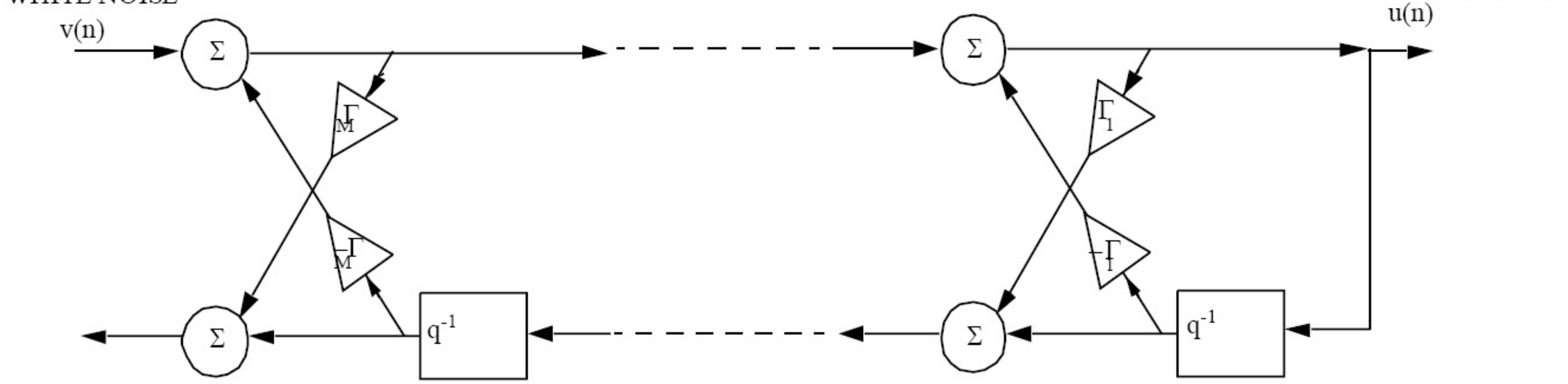
and simply rewriting the first equation

$$\begin{aligned} f_{m-1}(n) &= f_m(n) - \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= \Gamma_m f_{m-1}(n) + b_{m-1}(n-1) \end{aligned}$$

we obtain the basic stage of the Lattice inverse filter representation.



LATTICE INVERSE FILTER OF ORDER M



- **Joint–process estimation**

  Find the optimal (in MSE sense) filter recovering a desired signal $d(n)$ from the signal $u(n)$

  – not using directly the observations $u(n), u(n-1), \ldots, u(n-m)$ as in FIR filtering

  – but using instead the samples $b_0(n), b_1(n), \ldots, b_M(n)$ which comes from the orthogonalization of $u(n)$ using a lattice filter.

  The structure of the filter comprises two sections:

  – one *lattice predictor* section with reflection coefficients $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$, transforming the observations $u(n), u(n-1), \ldots, u(n-m)$ into the sequence of uncorrelated errors $b_0(n).b_1(n), \ldots, b_M(n)$;

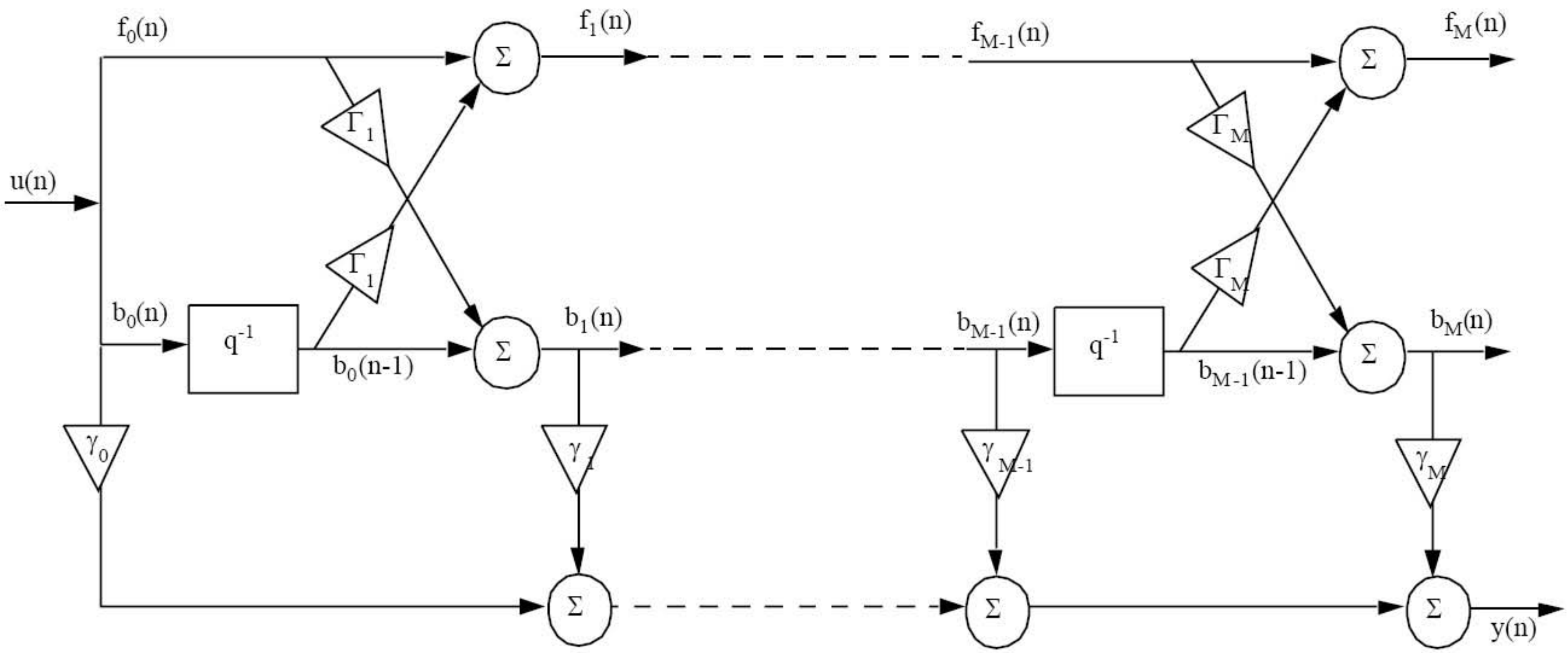
  – a *multiple regression filter*, with parameters $\gamma_0, \gamma_1, \ldots, \gamma_M$ which uses as observations the samples $b_0(n).b_1(n), \ldots, b_M(n)$ to compute the output of the filter $y(n)$.

  Denoting

$$
\begin{aligned}
\underline{b}(n) &= \left[\begin{array}{cccc} b_0(n) & b_1(n) & \ldots & b_M(n) \end{array}\right]^T \\
\underline{\gamma} &= \left[\begin{array}{cccc} \gamma_0 & \gamma_1 & \ldots & \gamma_M \end{array}\right]^T
\end{aligned}
$$

  we can write the optimal Wiener filter

$$
\underline{\gamma} = [E\underline{b}(n)\underline{b}(n)]^{-1} E\underline{b}(n)d(n)
$$



LATTICE FILTER BASED JOINT - PROCESS ESTIMATION

Relationship between Lattice parameters and optimal (direct) FIR filter parameters

We found the autocorrelation matrix of backward errors to be

$$E[\underline{b}(n)\underline{b}(n)^T] = \begin{bmatrix} P_0 & 0 & \dots & 0 \\ 0 & P_1 & \dots & 0 \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ 0 & 0 & \dots & P_M \end{bmatrix} = D$$

and from $\underline{b}(n) = L\underline{u}(n)$ we found

$$E[\underline{b}(n)\underline{b}(n)^T] = LE[\underline{u}(n)\underline{u}(n)^T]L^T = LRL^T = D$$

We can now compute the optimal $\underline{\gamma}$ parameters as

$$\underline{\gamma} \;=\; [E\underline{b}(n)\underline{b}(n)]^{-1}E\underline{b}(n)d(n) = D^{-1}E\underline{b}(n)d(n) = D^{-1}EL\underline{u}(n)d(n) = D^{-1}L\underline{p} = D^{-1}LR\underline{w}_o$$

Multiplying both sides with L^T and recalling $R^{-1} = L^T D^{-1} L$ we obtain

$$L^T\underline{\gamma} \;=\; \underline{w}_o$$

Thus we have a one-to-one correspondence between the parameters of the optimal FIR filter, $\underline{w}_o$ and the parameters of the optimal lattice filter.

- **Burg estimation algorithm**

  The optimum design of the lattice filter is a decoupled problem.

At stage m the optimality criterion is:

$$J_m = E[f_m^2(n)] + E[b_m^2(n)]$$

and using the stage m equations

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + \Gamma_m b_{m-1}(n-1) \\ b_m(n) &= b_{m-1}(n-1) + \Gamma_m f_{m-1}(n) \end{aligned}$$

$$\begin{aligned} J_m &= E[f_m^2(n)] + E[b_m^2(n)] = E[(f_{m-1}(n) + \Gamma_m b_{m-1}(n-1))^2] + E[(b_{m-1}(n-1) + \Gamma_m f_{m-1}(n))^2] \\ &= E[(f_{m-1}^2(n) + b_{m-1}^2(n-1)](1+\Gamma_m^2) + 4\Gamma_m E[b_{m-1}(n-1) f_{m-1}(n)] \end{aligned}$$

Taking now the derivative with respect to Γ_m of the above criterion we obtain

$$\frac{d(J_m)}{d\Gamma_m} = 2E[(f_{m-1}^2(n) + b_{m-1}^2(n-1)]\Gamma_m + 4E[b_{m-1}(n-1) f_{m-1}(n)] = 0$$

and therefore

$$\Gamma_m^* = -\frac{2E[b_{m-1}(n-1) f_{m-1}(n)]}{E[(f_{m-1}^2(n)] + E[b_{m-1}^2(n-1)]}$$

Replacing the expectation operator E with time average operator $\frac{1}{N}\sum_{n=1}^{N}$ we obtain one direct way to estimate the parameters of the lattice filter, starting from the data available in lattice filter:

$$\Gamma_m = -\frac{2\sum_{n=1}^{N} b_{m-1}(n-1) f_{m-1}(n)}{\sum_{n=1}^{N}[(f_{m-1}^2(n) + b_{m-1}^2(n-1)]}$$

The parameters $\Gamma_1, \ldots, \Gamma_M$ can be found solving first for Γ_1, then using Γ_1 to filter the data $u(n)$ and obtain $f_1(n)$ and $b_1(n)$, then find the estimate of Γ_2 ....

There are other possible estimators, but Burg estimator ensures the condition $|\Gamma| < 1$ which is required for the stability of the lattice filter.

- **Gradient Adaptive Lattice Filters**

  Imposing the same optimality criterion as in Burg method

$$J_m = E[f_m^2(n)] + E[b_m^2(n)]$$

the gradient method applied to the lattice filter parameter at stage m is

$$\frac{d(J_m)}{d\Gamma_m} = 2E[f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n)]$$

and can be approximated (as usually in LMS algorithms) by

$$\hat{\nabla} J_m \approx 2[f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n)]$$

We obtain the updating equation for the parameter Γ_m

$$\Gamma_m(n+1) = \Gamma_m(n) - \frac{1}{2}\mu_m(n)\hat{\nabla} J_m = \Gamma_m(n) - \mu_m(n)(f_m(n)b_{m-1}(n-1) + f_{m-1}(n)b_m(n))$$

In order to normalize the adaptation step, the following value of $\mu_m(n)$ was suggested

$$\mu_m(n) = \frac{1}{\xi_{m-1}(n)}$$

where

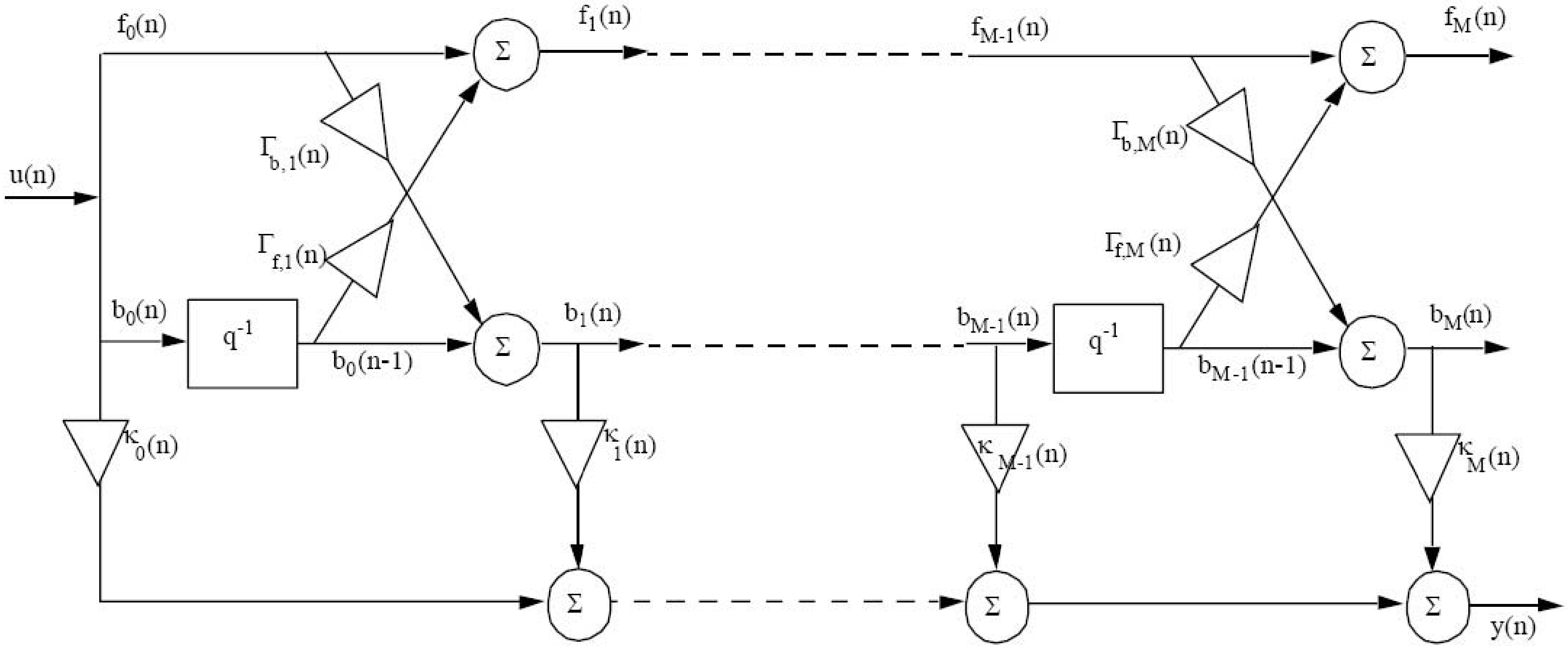
$$\xi_{m-1}(N) = \sum_{i=1}^{N}[(f_{m-1}^2(i) + b_{m-1}^2(i-1)] = \xi_{m-1}(N-1) + f_{m-1}^2(N) + b_{m-1}^2(N-1)$$

represents the total energy of forward and backward prediction errors.

We can introduce a forgetting factor using

$$\xi_{m-1}(n) = \beta\xi_{m-1}(n-1) + (1-\beta)[f_{m-1}^2(n) + b_{m-1}^2(n-1)]$$

with the forgetting factor close to 1, but $0 < \beta < 1$ allowing to forget the old history, which may be irrelevant if the filtered signal is nonstationary.



LEAST SQUARES LATTICE FILTER BASED JOINT - PROCESS ESTIMATION