Other Adaptive Algorithms

- LMS is the most popular algorithm

- What problems LMS potentially have?
  - slow convergence speed
  - choice of best step size

Normalized LMS

- LMS:

$$H(n+1) = H(n) + \delta X(n+1)e(n+1)$$

- NLMS: $H(n+1) = H(n) + \delta X(n+1)e(n+1)/\sigma_x^2$

signal power $\sigma_x{}^2$ can be estimated by

$$P_x(n) = P_0 + \frac{1}{N_0}\sum_{i=0}^{N_0} x^2(n-i)$$

or by

$$P_x(n) = (1-\gamma)P_x(n-1) + \gamma x^2(n)$$

Delayed LMS

- In the implementation, it is sometimes easier to update coefficients with some delay as

$$H(n+1) = H(n) + \delta X(n+1-d)e(n+1-d)$$

- The delay makes the stability condition more stringent.

Motivation for Recursive Algorithm

- **LMS**: simple but slow.

  Coefficient vector will "rattle around" the optimal rather than actually converge to it.
- **Another approach**: Use input data so as to ensure optimality at each step.

  If this can be done, the last vector is the overall optimal.

"Recursive-in-Time" Algorithm

Cost function

$$J(n) = \sum_{p=1}^{n} [y(p) - H^T(n-1)X(p)]^2$$

uses all the available data up to n.

$H(n)$ is optimized to minimize $J(n)$

- Some procedure by which $H(n)$ is updated to the new optimal vector $H(n+1)$ when new samples become available.

Simplest Update Formula

- Update R_N by $R_N(n+1) = R_N(n) + X(n+1)X^T(n+1)$
- Update r_{yx} by $r_{yx}(n+1) = r_{yx}(n) + X(n+1)y(n+1)$
- Invert $R_N(n+1)$
- Compute $H(n+1)$ by $H(n+1) = R_N^{-1}(n+1)r_{yx}(n+1)$

Direct, but computationally wasteful.

$O(N^3)$ multiplications

Matrix Inversion Lemma

Given matrices A,B,C and D satisfying

$$A = B + CDC^T$$

then inverse of A is

$$A^{-1} = B^{-1} - B^{-1}C(C^T B^{-1} C + D^{-1})^{-1} C^T B^{-1}$$

The Lemma is applied

$$B = R_N(n), \qquad C = X(n+1), \qquad D = 1$$

$$R_N^{-1}(n+1)$$
$$= R_N^{-1}(n) - \frac{R_N^{-1}(n)X(n+1)X^T(n+1)R_N^{-1}(n)}{1 + X^T(n+1)R_N^{-1}(n)X(n+1)}$$

We never compute $R_N(n+1)$,

nor do we invert it directly.

Optimal Coefficient Vector

$$H(n+1) = R_N^{-1}(n+1)r_{yx}(n+1)$$

$$= \left\{R_N^{-1}(n) - \frac{R_N^{-1}(n)X(n+1)X^T(n+1)R_N^{-1}(n)}{1 + X^T(n+1)R_N^{-1}(n)X(n+1)}\right\}\{r_{yx}(n) + X(n+1)y(n+1$$

$$= R_N^{-1}(n)r_{yx}(n) - \frac{R_N^{-1}(n)X(n+1)X^T(n+1)R_N^{-1}(n)r_{yx}(n)}{1 + X^T(n+1)R_N^{-1}(n)X(n+1)}$$

$$+ R_N^{-1}(n)X(n+1)y(n+1)$$

$$- \frac{y(n+1)X(n+1)R_N^{-1}(n)X(n+1)X^T(n+1)R_N^{-1}(n)}{1 + X^T(n+1)R_N^{-1}(n)X(n+1)}$$

Simplification

$Z(n+1) = R_N^{-1}(n)X(n+1)$: filtered information vector
$\hat{y}(n+1) = X^T(n+1)H(n)$: a priori output
$q = X^T(n+1)Z(n+1)$: signal power with normalization

$$
\begin{aligned}
&H(n+1) \\
&= H(n) - \frac{Z(n+1)\hat{y}(n+1)}{1 + X^T(n+1)Z(n+1)} + Z(n+1)y(n+1) - \frac{y(n+1)Z(n+1)X^T(n+1)Z(n+1)}{1 + X^T(n+1)Z(n+1)} \\
&= H(n) - \frac{Z(n+1)\hat{y}(n+1)}{1+q} + Z(n+1)y(n+1) - \frac{y(n+1)qZ(n+1)}{1+q} \\
&= H(n) - \frac{Z(n+1)\hat{y}(n+1)}{1+q} + \frac{y(n+1)Z(n+1)}{1+q} \\
&= H(n) + \frac{Z(n+1)\{y(n+1) - \hat{y}(n+1)\}}{1+q} \\
&= H(n) + \frac{e(n+1)Z(n+1)}{1+q}
\end{aligned}
$$

Recursive Least Squares algorithm

Correction Term to be Added to Update

$e(n+1) = y(n+1) - \hat{y}(n+1)$: a priori error

$Z(n+1)$: filtered information vector

$R_N^{-1}(n)$ acts to influence or "filter" the data vector

q: a measure of the input signal power
with normalization introduced by $R_N^{-1}(n)$

$$0 \le q \le 1$$

$O(N^2)$ multiplications

Initial Condition

The procedure assumes $R_N^{-1}(n)$ exists.

Two methods;

- Acquire N input samples and $R_N^{-1}(n)$ is computed directly.
- $R_N^{-1}(0) = \gamma I_N,\ \ \gamma$: large positive constant

  inaccurate but simple

  $R_N^{-1}(n)$ is gradually corrected.

Multirate Adaptive Filters

- Sampling Frequencies of input and reference signals can be different.

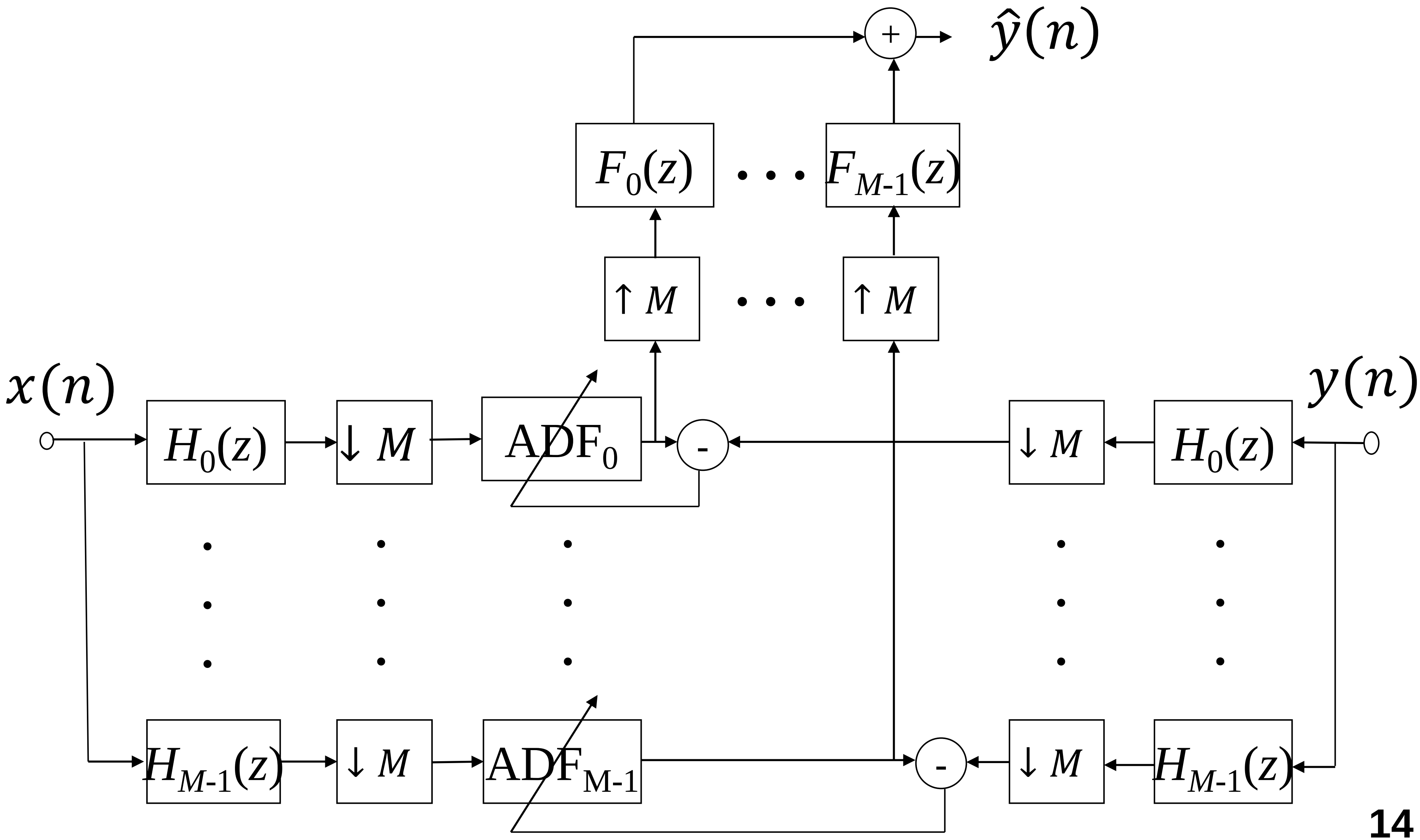
Sampling rate conversion

- For colored input signals, convergence speed becomes slow.

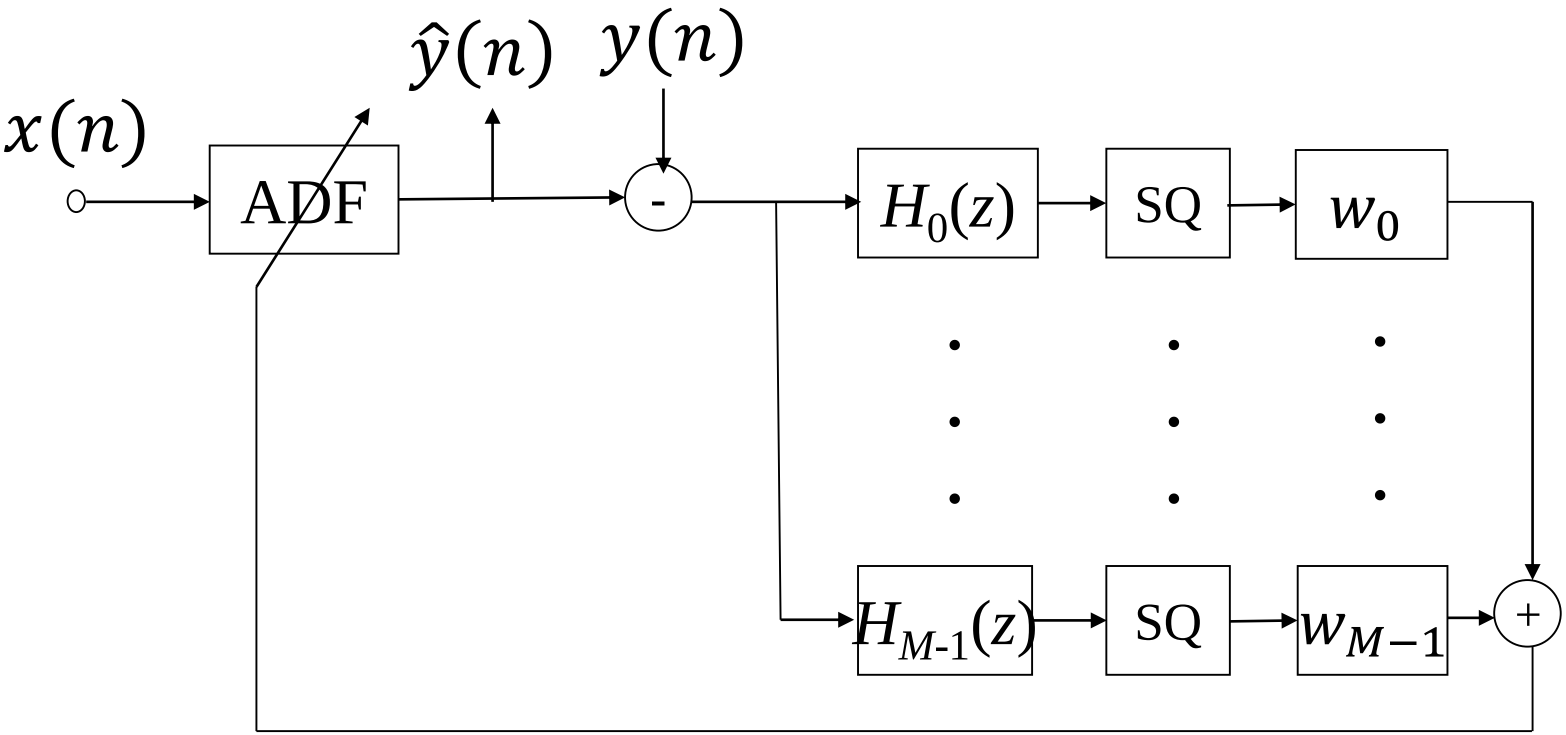
Subband decomposition

In each band, unevenness in the spectrum is reduced.

Subband Adaptive Filters

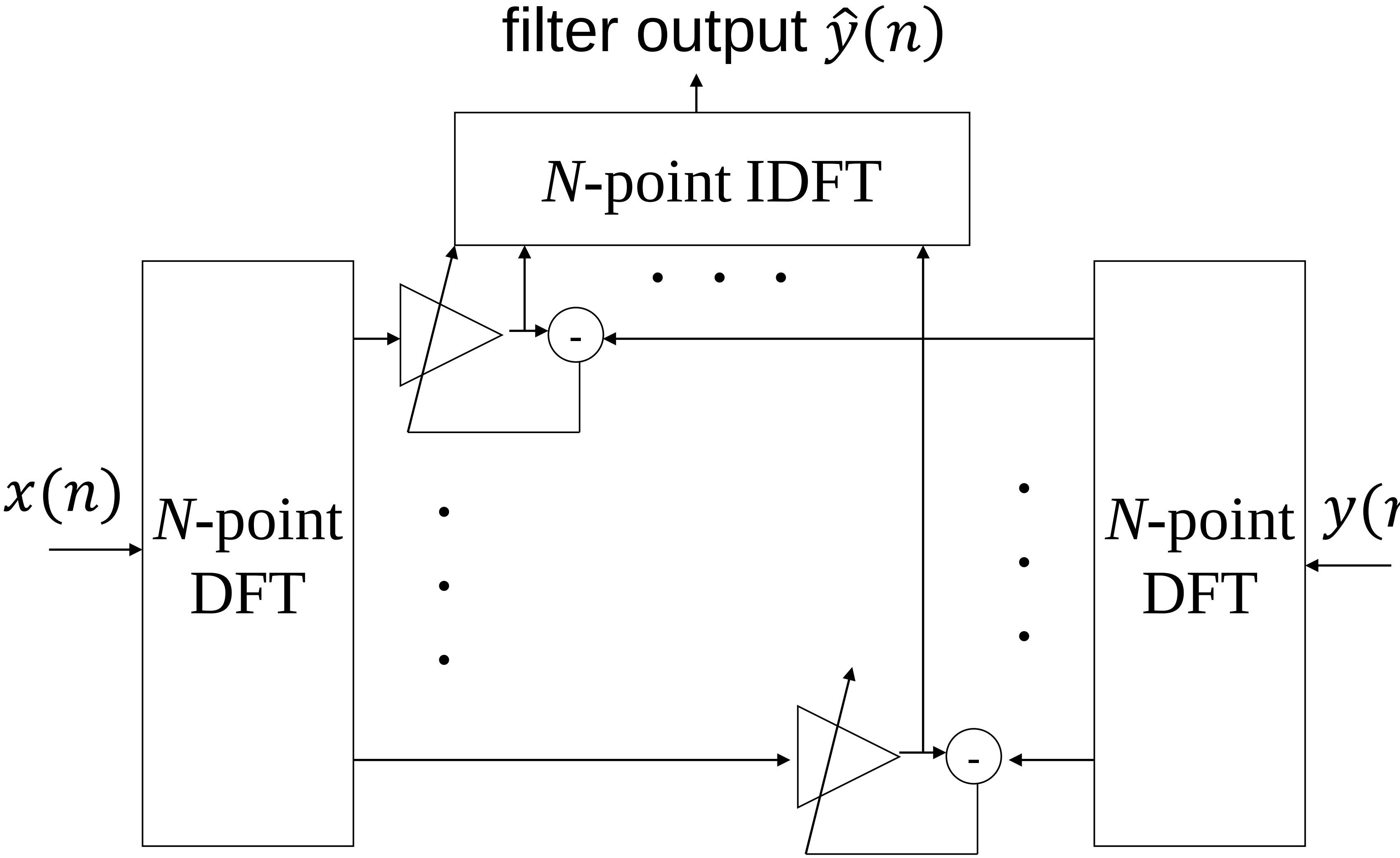


Multi-Band Decomposition of Error



w_i : window functions
long for low frequency and
short for high frequency

Frequency Domain Adaptive Filters



Frequency Domain Adaptive Filters

N-tap adaptive filter ⇨ A set of separate 1-tap filters

Efficient FFT algorithm

Real-valued transforms such as DCT can also be used.

Exercise 10

1. Adaptive filters have coefficient update equations given by

   [new coefficient vector] =[old coefficient vector]

   + [step size] × [input data vector] × [innovation signal].

   a. What will happen if the step size parameter is too large?
   b. What will happen if the step size parameter is too small?
   c. If the step size parameter can also be varied, what will be a suitable strategy to change the parameter?

2. Autocorrelation matrix is defined as

$$R(n) = \sum_{p=n-L+1}^{n} X(p)X^T(p)$$

   where $X^T(n) = [x(n)\ x(n-1) \cdots x(n-N+1)]$.

   a. Show the expression for R_{ij}, ij-th entry of $R(n)$, in terms of $x(n)$,
   b. State the physical meanings of R_{ii} and R_{ij}.
   c. What specific feature does $R(n)$ have in general and in the case where $x(n)$ is stationary?

3. A measurement gives the signal autocorrelation values $r(0) = 5.0, r(1) = 3.0, r(2) = 0.4$. Calculate the two coefficients of the second-order linear predictor and the prediction error power. Give the corresponding signal power spectrum.