Adaptive Filters

filter : extract a prescribed quantity of interest from noisy data

usual assumption:

- stationary
- all the pertinent statistics are known
- linear

least squares solution: Wiener filter

Wiener, 1949

Kolmogorov, 1941

non stationary environment → adaptive filtering

self-designing system

Classification of AF

- linear
- non linear

- open loop — statistics analyzed are used in a separate extraction module
- closed loop — filter parameters are updated using extracted information

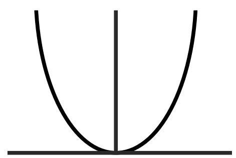
- supervised — training sequence available
- unsupervised

Further Classification

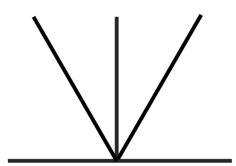
- FIR
- IIR

cost function

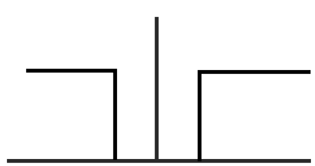
- LMS : least mean squares



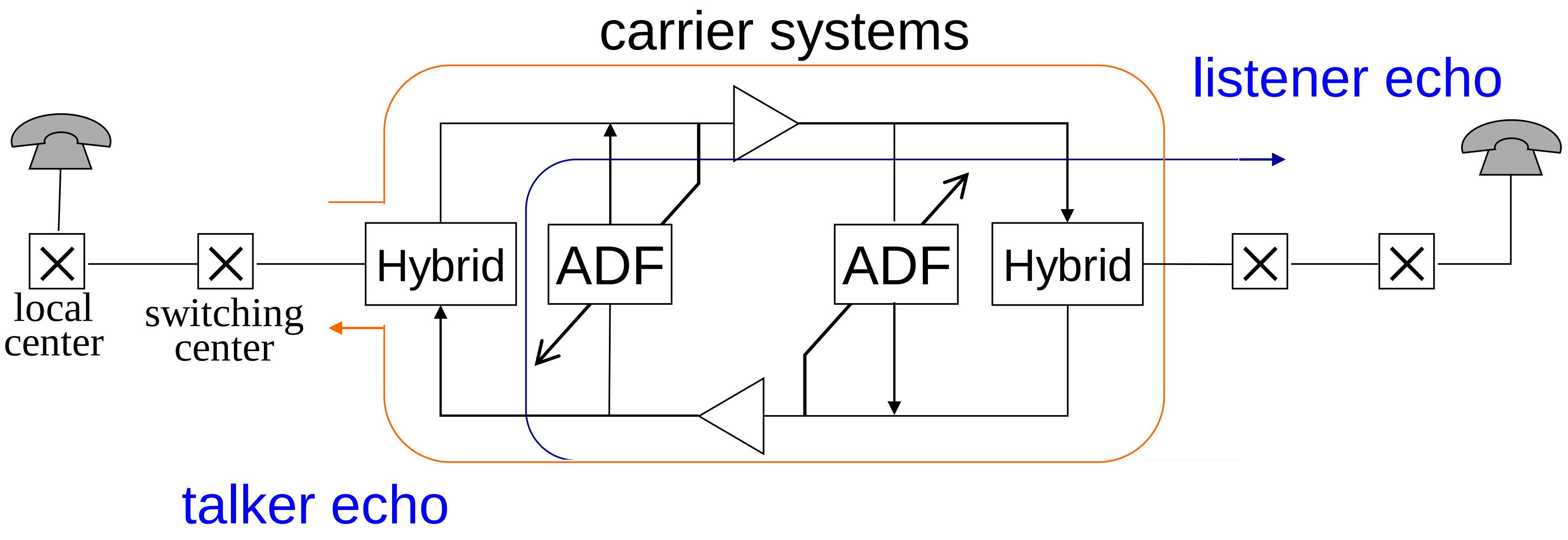
- modulus



- threshold

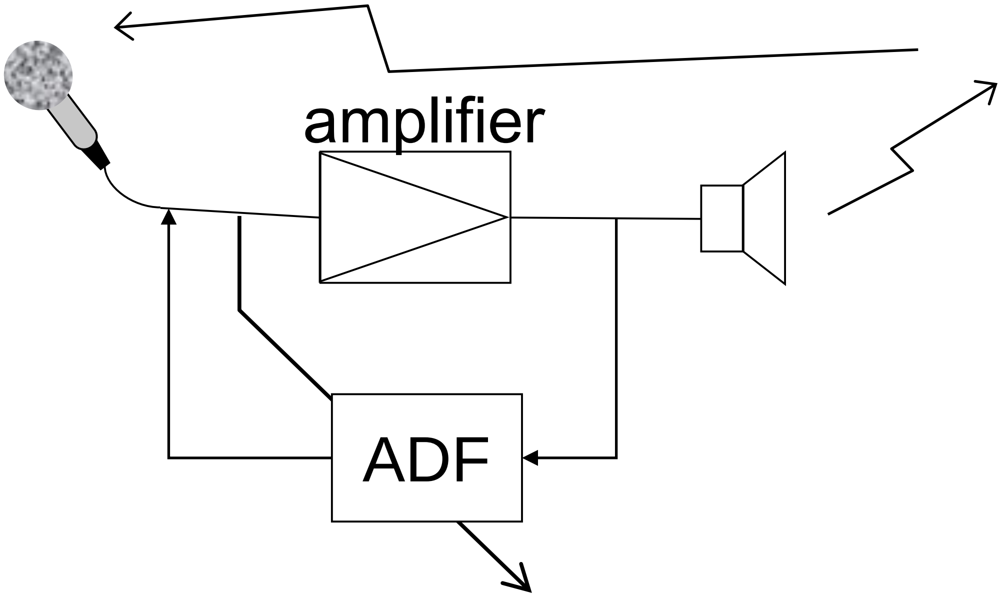


Applications



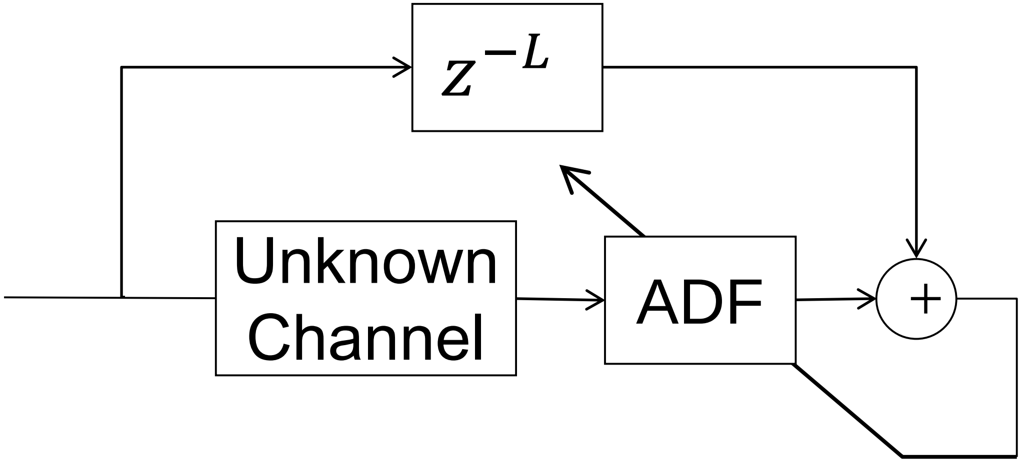
echo canceller

Applications (cont'd)



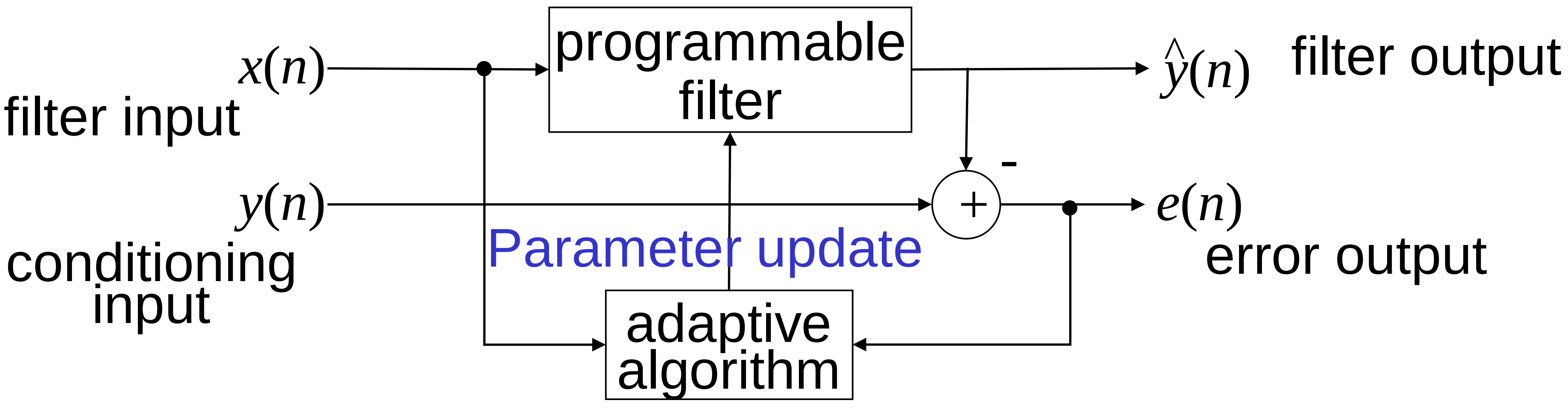
howling canceller

Applications (cont'd)

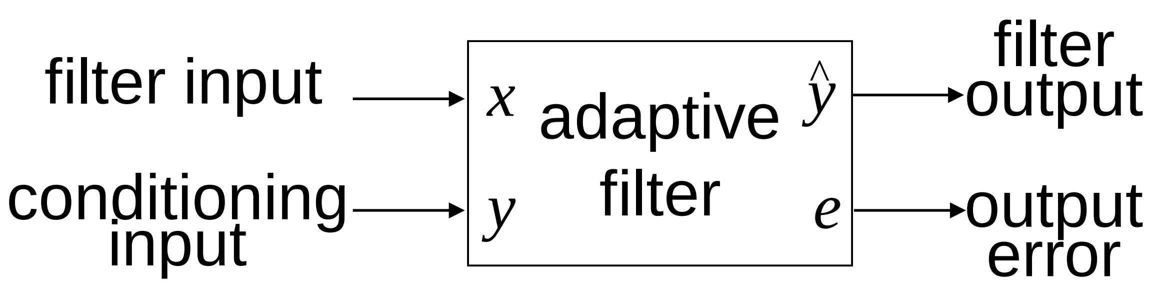


channel equalizer

Adaptive Filters

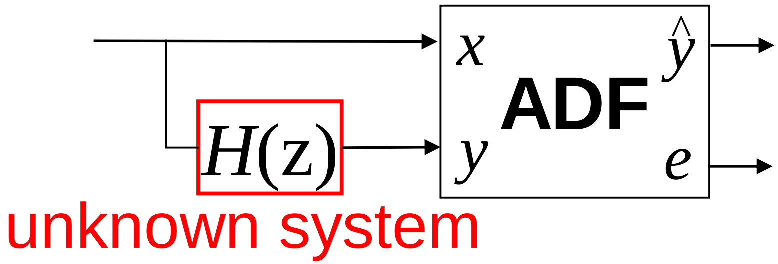


general schematic

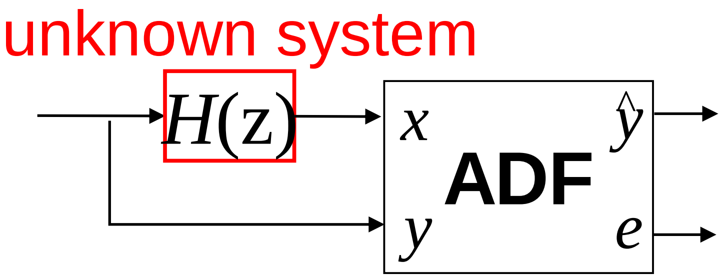


two-input, two-output system

System Modeling



direct modeling



inverse modeling

Error Output and Cost Function

$$H(n) = \begin{bmatrix} h_0(n) \\ h_1(n) \\ \vdots \\ h_{N-1}(n) \end{bmatrix} \qquad X(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix}$$

N-tap FIR time-varying

N most recent input

error signal $\varepsilon(n) = y(n) - H^T(n)X(n)$ a posteriori

At each time index, cost function is minimized

weighted sum of squared error

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

W: weight close to 1

Normal Equation

find $H(n)$ which minimizes $J(n)$

$$\frac{\partial J(n)}{\partial h_i(n)} = -2\sum_{p=1}^{n} W^{n-p}\{y(p) - H^T(n)X(p)\}X(p) = 0$$

$$\sum_{p=1}^{n} W^{n-p}\, X(p)X^T(p)H(n) = \sum_{p=1}^{n} W^{n-p}\, X(p)y(p)$$

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

$$\begin{cases} R(n) = \sum_{p=1}^{n} W^{n-p}\, X(p)X^T(p) \\ r_{yx}(n) = \sum_{p=1}^{n} W^{n-p}\, X(p)y(p) \end{cases}$$

Normal Equation (cont'd)

$R_{xx} = E[X(p)X^T(p)]$ signal autocorrelation matrix

$r_{yx} = E[X(p)y\,(p)]$ cross-correlation between input and reference

$$E[R_N(n)] = \frac{1 - W^n}{1 - W} R_{xx}$$

$$E\left[r_{yx}(n)\right] = \frac{1 - W^n}{1 - W} r_{yx}$$

Optimal coefficient vector

$n \to \infty$

$$H_{opt} = R_{xx}^{-1} r_{yx}$$

normal equation

Yule-Walker equation

Recursive algorithms

update $H(n+1)$ from $H(n)$

$$\begin{cases} R(n+1) = WR(n) + X(n+1)X^T(n+1) \\ r_{yx}(n+1) = Wr_{yx}(n) + X(n+1)y(n+1) \end{cases}$$

Now

$$\begin{aligned} H(n+1) &= R^{-1}(n+1)r_{yx}(n+1) \\ &= R^{-1}(n+1)\left[Wr_{yx}(n) + X(n+1)y(n+1)\right] \end{aligned}$$

But

$$\begin{aligned} Wr_{yx}(n) &= WR(n)H(n) \\ &= [R(n+1) - X(n+1)X^T(n+1)]H(n) \end{aligned}$$

and

$$\begin{aligned} H(n+1) &= R^{-1}(n+1)[\{R(n+1) - X(n+1)X^T(n+1)\}H(n) + X(n+1)y(n+1)] \\ &= H(n) + R^{-1}(n+1)X(n+1)\underbrace{[y(n+1) - X^T(n+1)H(n)]}_{e(n+1)} \end{aligned}$$

$e(n+1)$ a priori error

Gradient - LMS Algorithm

- simple
- flexible
- robust
- easy to design
- performance is well characterized

Most widely used in all technical fields such as communications and control

Looped structure makes the exact analysis extremely difficult.

Under restrictive hypothesis not verified in practice.

Approximate investigations provide sufficient results.

LMS Algorithm

to avoid complicated matrix manipulation

$$R^{-1}(n+1) \cong \delta I$$

Then

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

$$e(n+1) = y(n+1) - X^T(n+1)H(n)$$

$$\frac{\partial e^2(n+1)}{\partial h_i(n)} = \underbrace{-2x(n-i+1)e(n+1)}_{\text{gradient}}$$

δ: adaptation step size

LMS Algorithm (cont'd)

$$h_i(n+1) = h_i(n) - \delta \frac{\partial e(n+1)}{\partial h_i(n)} e(n+1)$$

$\frac{\partial e(n+1)}{\partial h_i(n)} e(n+1)$ is the gradient of $\frac{1}{2} e^2(n+1)$

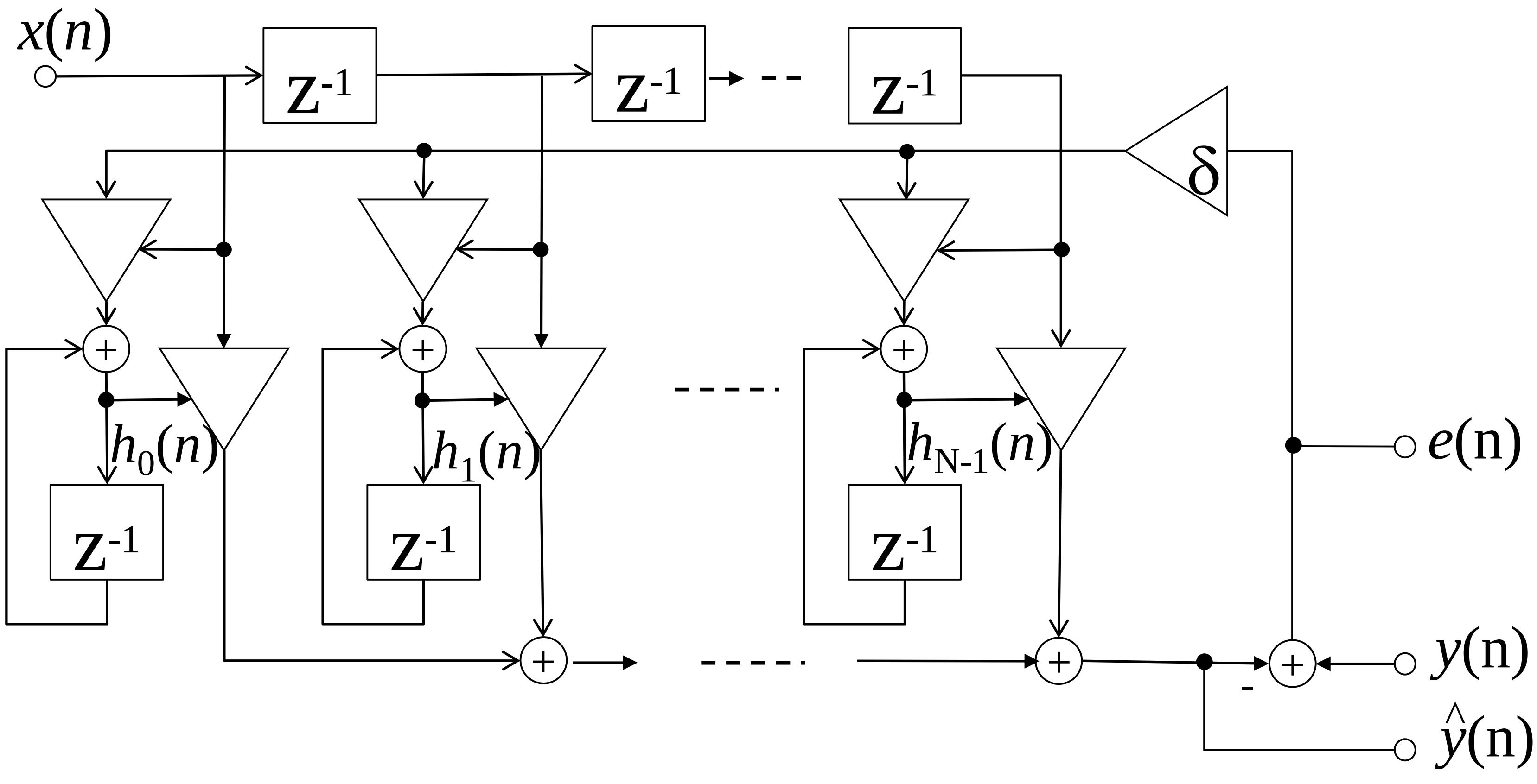
minimize the error power in the mean, thus LMS

for FIR filter $e(n+1) = y(n+1) - X^T(n+1)H(n)$

and

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

LMS Adaptive FIR Filter



Stability Condition

a priori error $e(n+1) = y(n+1) - H^T(n)X(n+1)$

coefficients before updating

a posteriori error $\varepsilon(n+1) = y(n+1) - H^T(n+1)X(n+1)$

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

$$\varepsilon(n+1) = e(n+1)[1 - \delta X^T(n+1)X(n+1)]$$

system is stable if
a posteriori error is smaller than a priori error

$$|1 - \delta E[X^T(n+1)X(n+1)]| < 1$$

$$0 < \delta < \frac{2}{N\sigma_x^2}$$ some margin necessary

System Gain and Time Constant

Two main specifications

- system gain $G_s^2 = \dfrac{E[y^2(n)]}{E[e^2(n)]}$

- time constant

$$\left(E[e^2(0)] - E[e^2(\infty)]\right)e^{-\frac{2}{\tau}} = E[e^2(1)] - E[e^2(\infty)]$$

τ is related to δ

Steady State

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

$$e(n+1) = y(n+1) - X^T(n+1)H(n)$$

$$H(n+1) = [1 - \delta X(n+1)X^T(n+1)]H(n) + \delta X(n+1)y(n+1)$$

$$n \to \infty$$

$$E[H(\infty)] = R^{-1}r_{yx} = H_{opt}$$

the gradient algorithm provides
the optimal coefficient H_{opt}

Leakage Factor

When input vanishes, coefficients are locked
It might be preferable to have them return to zero

$$H(n+1) = (1-\gamma)H(n) + \delta X(n+1)e(n+1)$$
$$= [(1-\gamma)I - \delta X(n+1)X^T(n+1)]H(n) + \delta y(n+1)X(n+1$$

after convergence

$$H_\infty = E[H(\infty)] = \left[R + \frac{\gamma}{\delta} I\right]^{-1} r_{yx}$$

γ introduces a bias on the coefficients

Exercise 8

1. Evaluate the mean and variance associated with the uniform probability density function on the interval $[x_1, x_2]$.
2. Find the first three terms of the ACF of the AR signal $x(n) = 1.27x(n-1) - 0.81x(n-2) + e(n)$, where $e(n)$ is a unit-power zero-mean white noise.
3. Consider the signal $x(n) = 0.8x(n-1) + e(n)$ for $n \geq 0$ $(x(n) = 0$ for $n < 0)$, where $e(n)$ is a stationary zero mean random sequence with power $\sigma^2 = 0.5$. The initial condition is $x(0)=1$.

   Calculate the mean sequence $m_n = \mathrm{E}[x(n)]$. Give the recursion for the variance sequence. What is the stationary solution. Calculate the ACF of the stationary signal.
4. Read the following paper;

   B. Widrow and M.E. Hoff, Jr., Adaptive Switching Circuits, *IRE WESCON Convention Record*, 4:96-104, August 1960.