Functional Link Artificial Neural Network (FLANN)

By

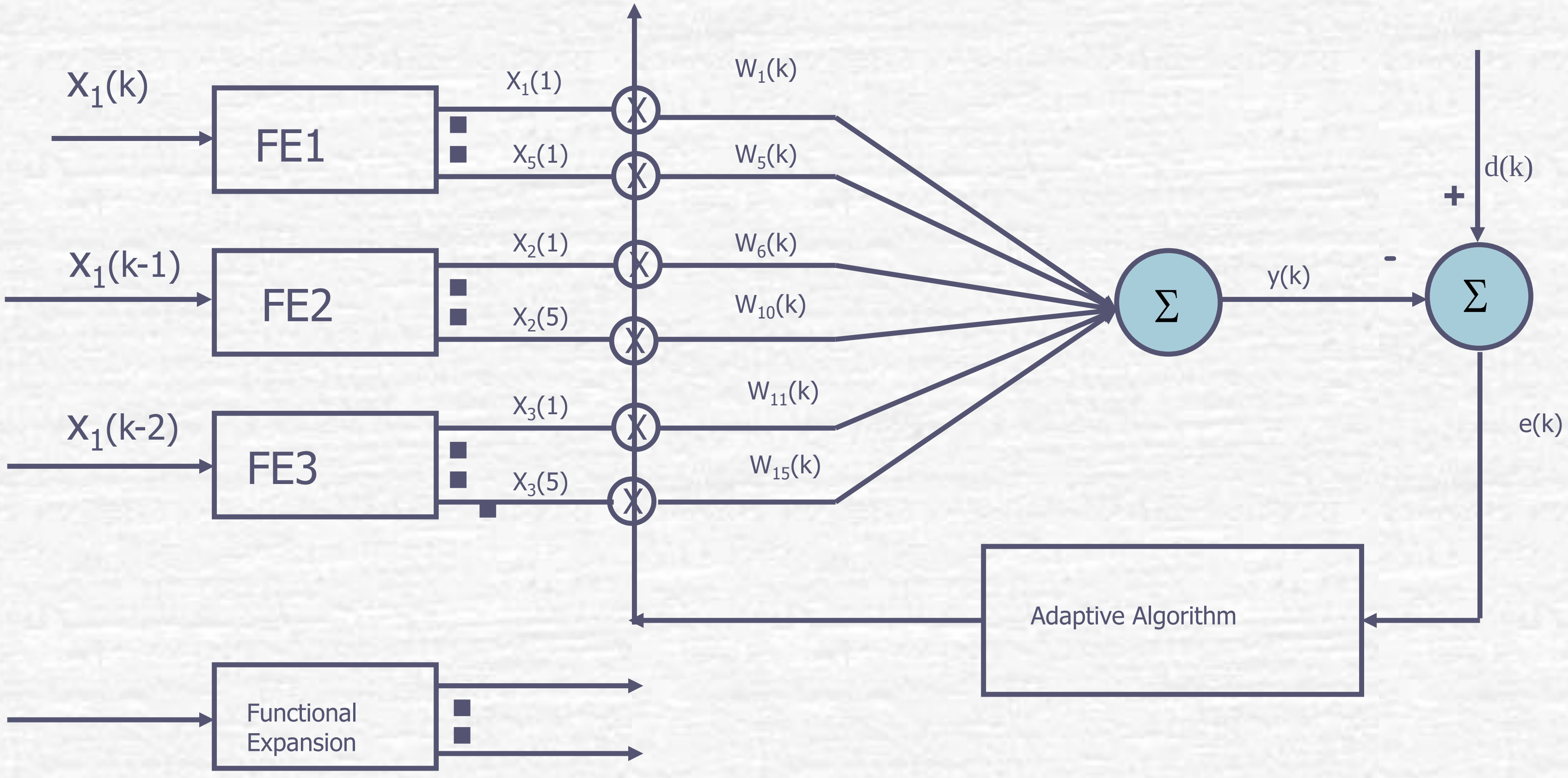
Prof. G. Panda, FNAE, FNASc.

IIT Bhubaneswar

FLANN

- The functional link ANN or Pao-network originally proposed by Pao.
- Single layer ANN structure
- Need of hidden layer is removed
- Capable of forming arbitrarily complex decision regions by generating nonlinear decision boundaries.
- Offers less computational complexity
- Higher convergence speed than MLP

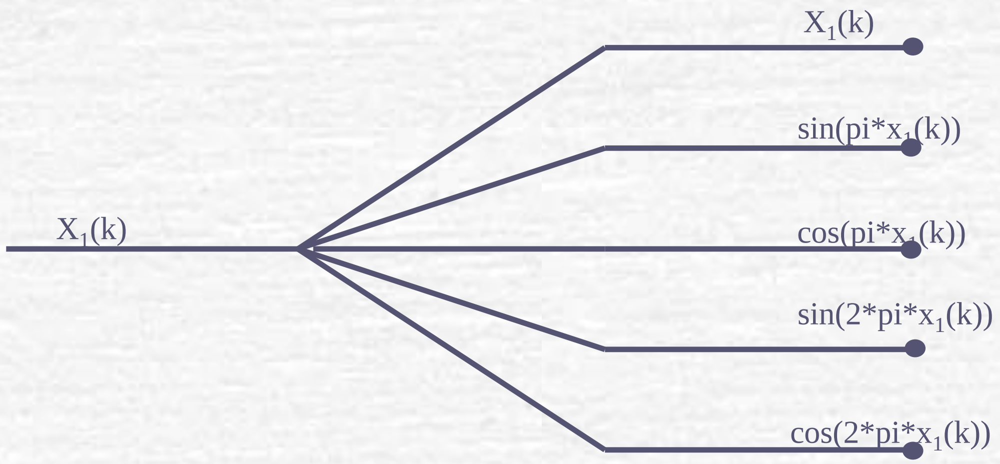
Structure of FLANN



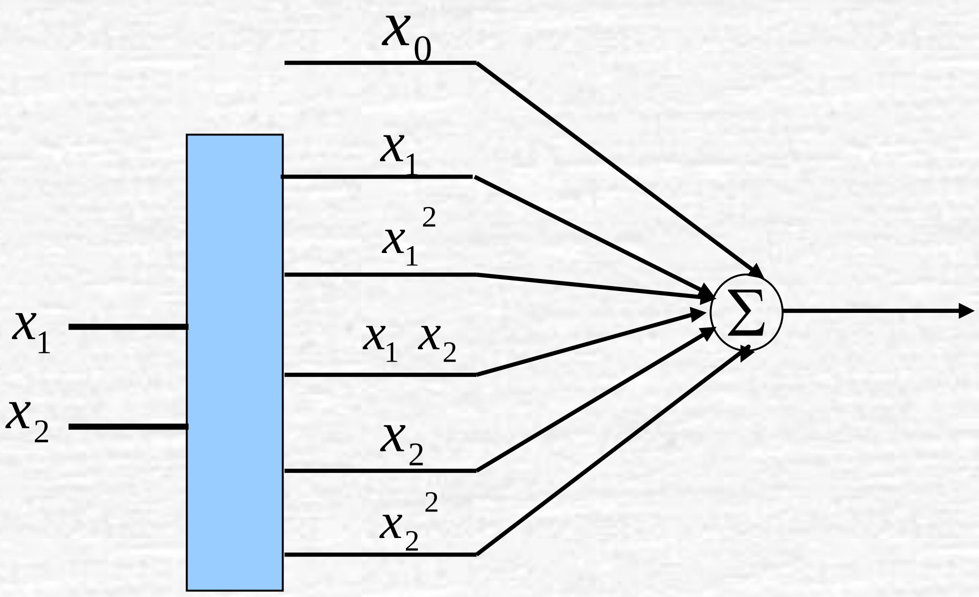
Types of Functional expansions

- Trigonometric
- Polynomial
- Exponential
- Chebyshev

Trigonometric Expansion



Polynomial Expansion



Chebyschev expansion:

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x)$$

The First Few Chebyschev Polynomials

$T_o(x) = 1$

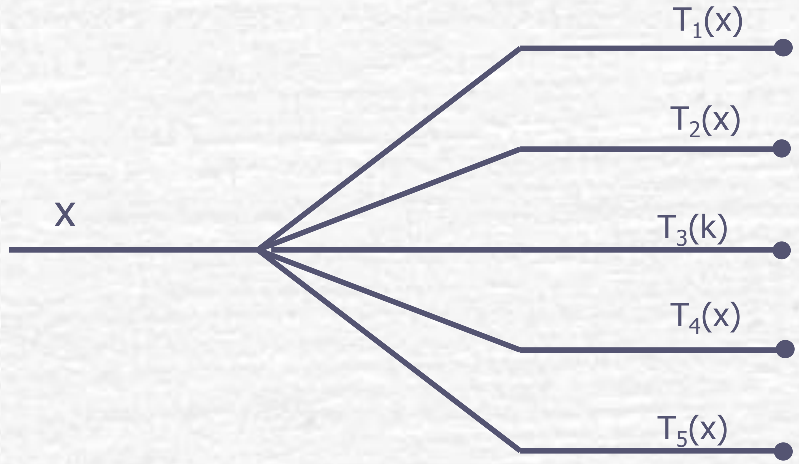
$T_1(x) = x$

$T_2(x) = 2x^2 - 1$

$T_3(x) = 4x^3 - 3x$

$T_4(x) = 8x^4 - 8x^2 + 1$

$T_5(x) = 16x^5 - 20x^3 + 5x$



Learning rule

$$Wt(k+1)=Wt(k) + mu * e(k)*X(k)$$

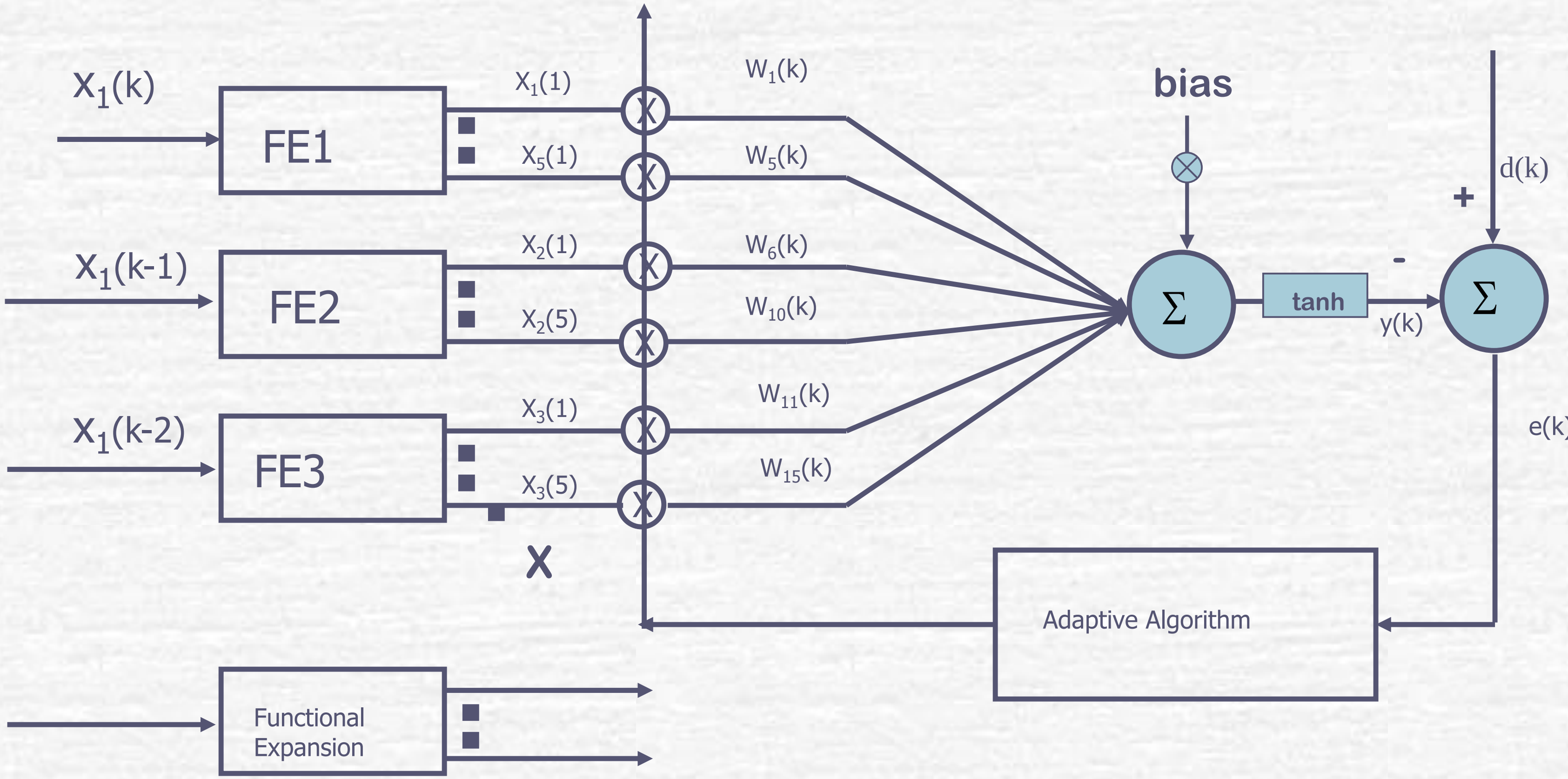
Where

mu=convergence coefficient

e(k) = error at kth instant

X(k) = expanded input vector at kth instant

FLANN Structure with sigmoid



Delta Learning

The error is given by

$$e(k) = d(k) - y(k)$$

Then the delta is calculated as

$$\delta(k) = (1 - y(k)^2) * e(k)$$

Now the weight update equation becomes

$$W(k+1) = W(k) + \mu\, \Delta(k) + \gamma\, \Delta(k-1)$$

where

$$\Delta(k) = \delta(k)\,[(X_k)]^T$$

Epoch based learning

Application of all N patterns constitutes one experiment.

At the end of each experiment N sets of $\underline{\Delta w}(i)$ are obtained

Then the average change of weight is computed as

$$\Delta w(i) = \frac{1}{N}\sum_{k=1}^{N} \Delta w\ (k)$$

The weights of the FLANN model is then updated according to the relation

$$\underline{w}(i+1) = \underline{w}(i) + \underline{\Delta w}(i)$$

Similarly the bias weight is updated using

$$w_b(i+1) = w_b(i) + \Delta w_b$$

Function Approximation

The two examples are

$$f_1(x) = x^3 + 0.3x^2 - 0.4x$$

$$f_2(x) = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x)$$

In both cases the input pattern is expanded using trigonometric expansion

Fifteen input nodes including a bias input are used

The nonlinearity associated is tanh( ) function.

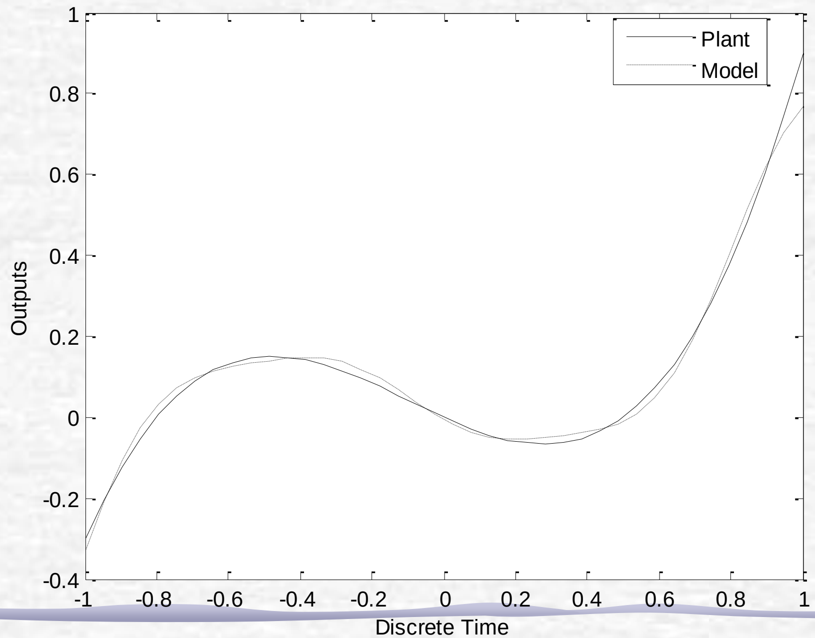
The convergence coefficient is set to 0.1

Training of the weights of FLANN model are carried out by using an uniformly distributed random signal over the interval [-1,1] as input.

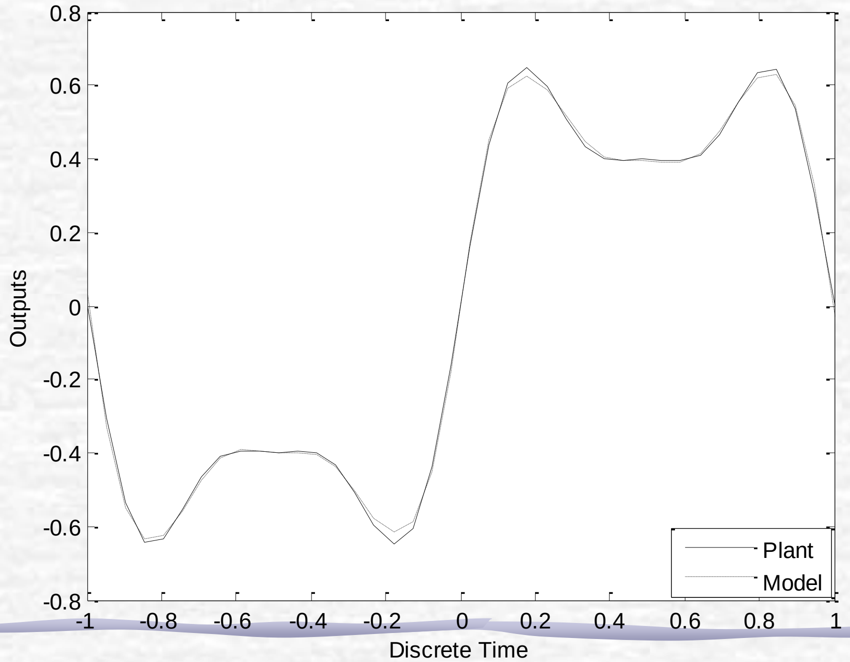
Function approximation

During testing the input to the identified model used is given by

$$x(k) = \begin{cases} \sin \dfrac{2\pi k}{250} & for\ k \le 250 \\ 0.8 \sin \dfrac{2\pi k}{250} + 0.2 \sin \dfrac{2\pi k}{25} & for\ k > 250 \end{cases}$$

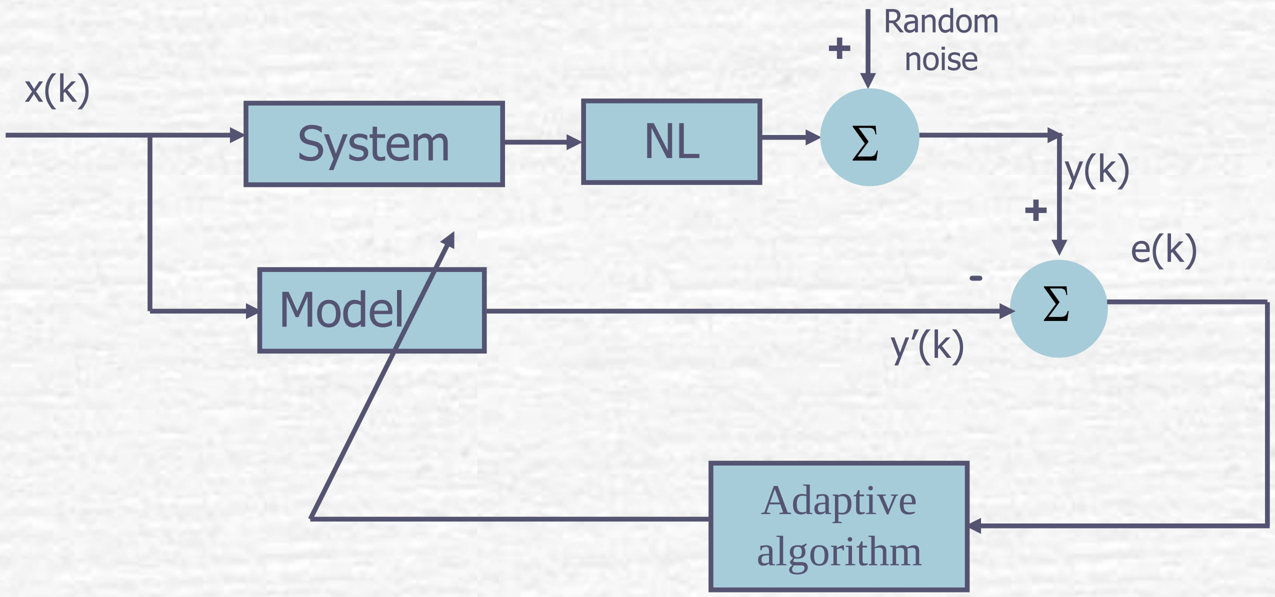


Example-1



Example-2

System Identification



Block Diagram for System Identification model

Simulation study:

- Input signal : uniformly distributed random signal [-.5,.5].
- SNR : 30dB
- H(z) : $0.260 + 0.930z^{-1} + 0.260z^{-2}$
- The learning parameter μ : 0.02
- Iteration : 2000

  Averaging : 50 times

- For channel equalization delay : 3
- Non linearity
- NL = 0 : b(k)= a(k)
- NL = 1 : y= tanh(x)
- NL = 2 :   y = x + 0.2 * (x^2) - 0.1*(x^3)
- NL = 3 :   y= x - 0.9 * (x.^3)
- NL = 4 : y= x + 0.2 * (x^2) - 0.1 * (x^3)+ 0.5*cos(pi*x)
- A linear channel is modeled by NL = 0

System Identification

y=x

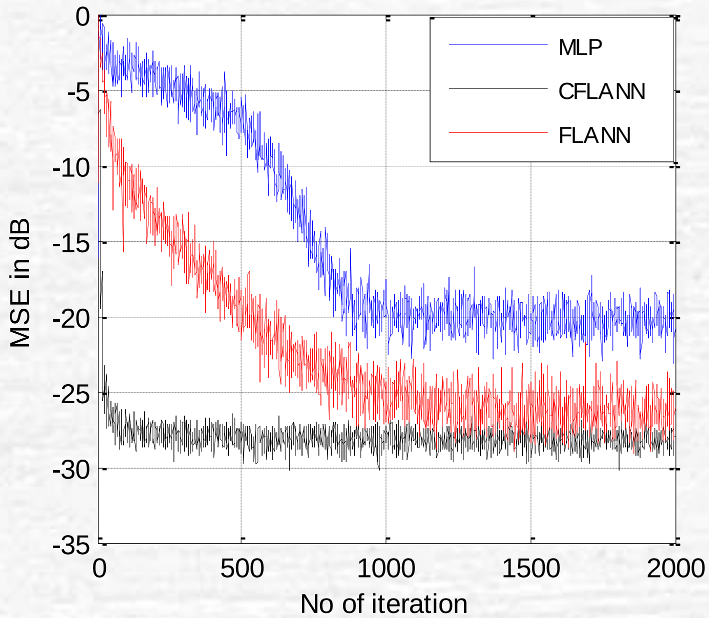


Fig. 6(a)

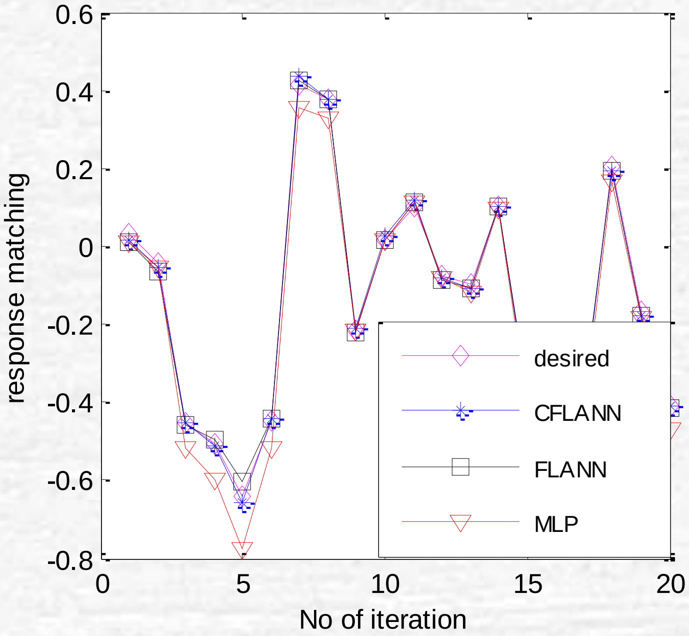


Fig. 6(b)

System Identification

y= tanh(x)

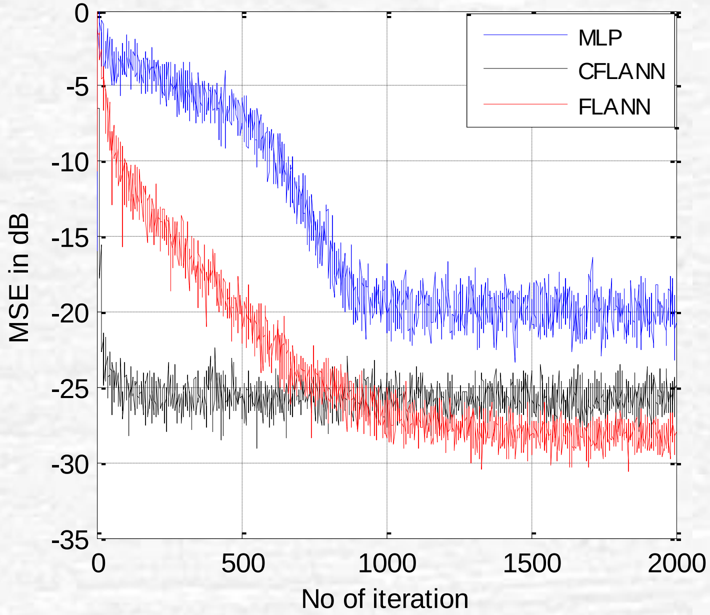


Fig. 6(c)

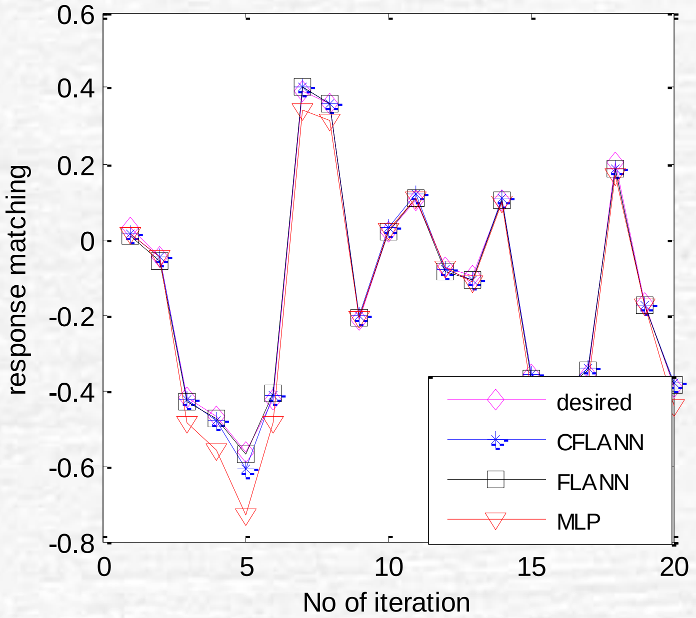


Fig. 6(d)

System Identification

y = x + 0.2 * (x^2) - 0.1*(x^3)

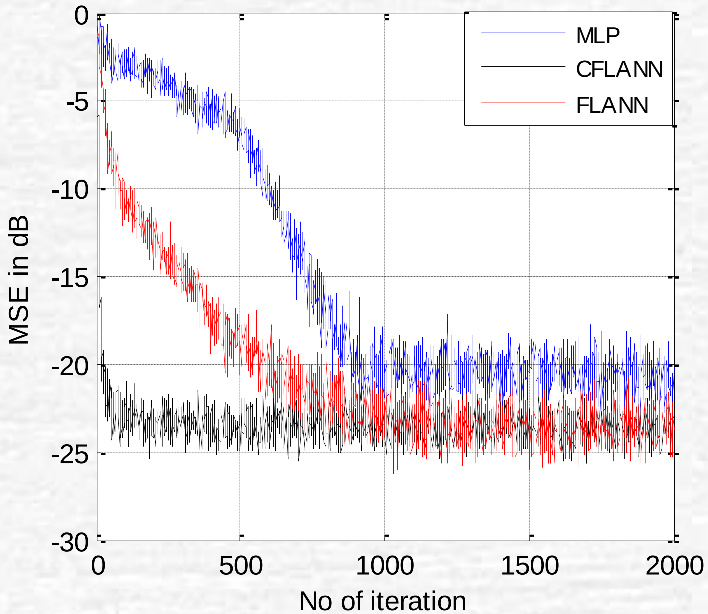


Fig. 6(e)

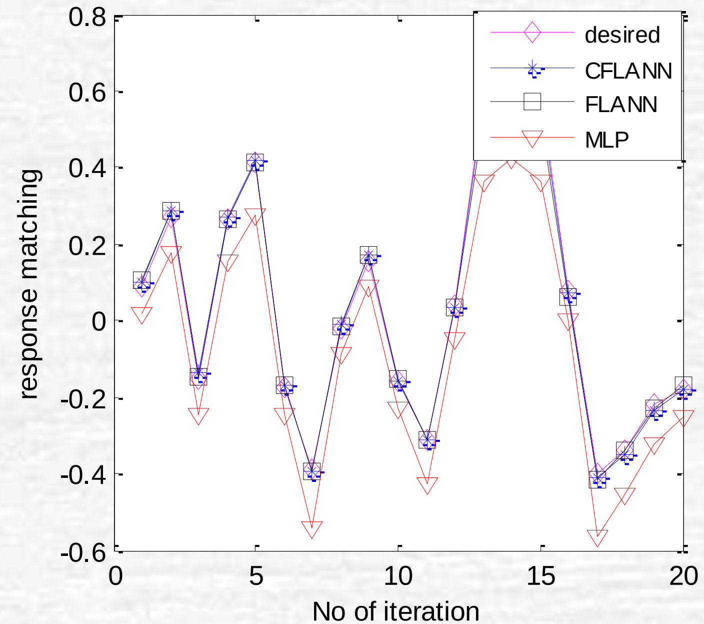


Fig. 6(f)

System Identification

y= x - 0.9 * (x.^3)

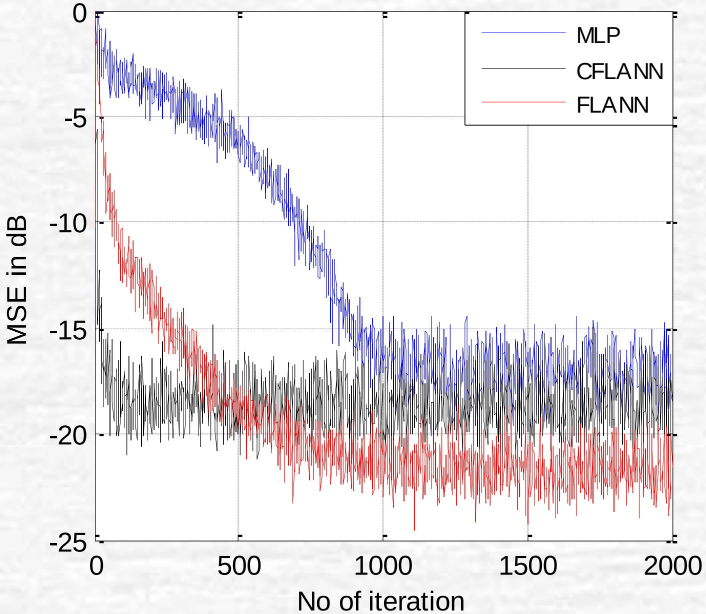


Fig. 6(g)

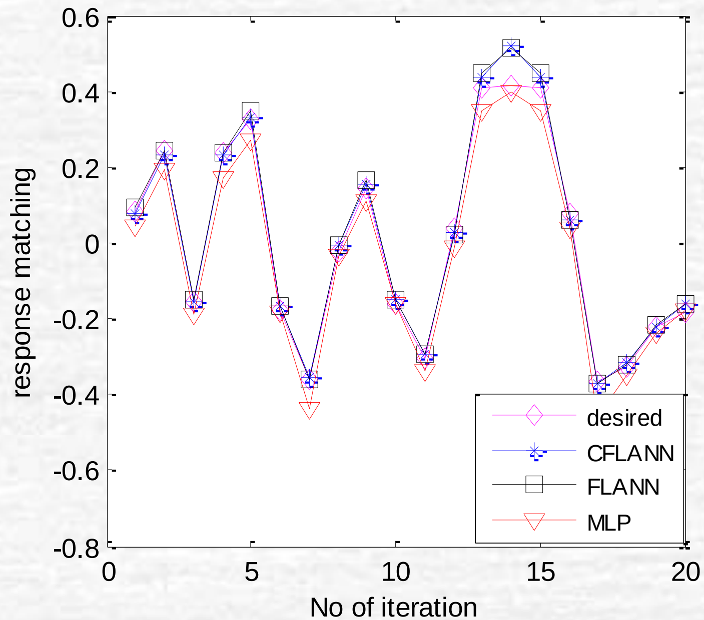


Fig. 6(h)

System Identification

y= x + 0.2 * (x^2) - 0.1 * (x^3)+ 0.5* cos(pi*x)

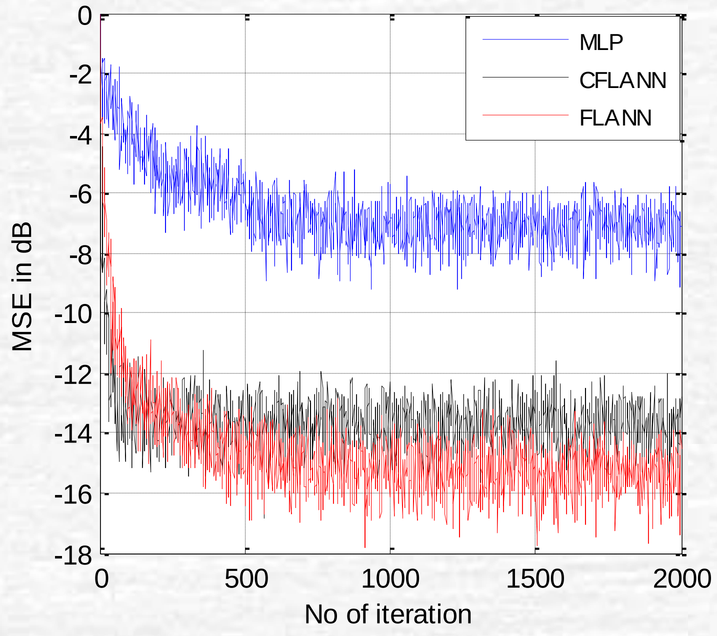


Fig. 6(i)

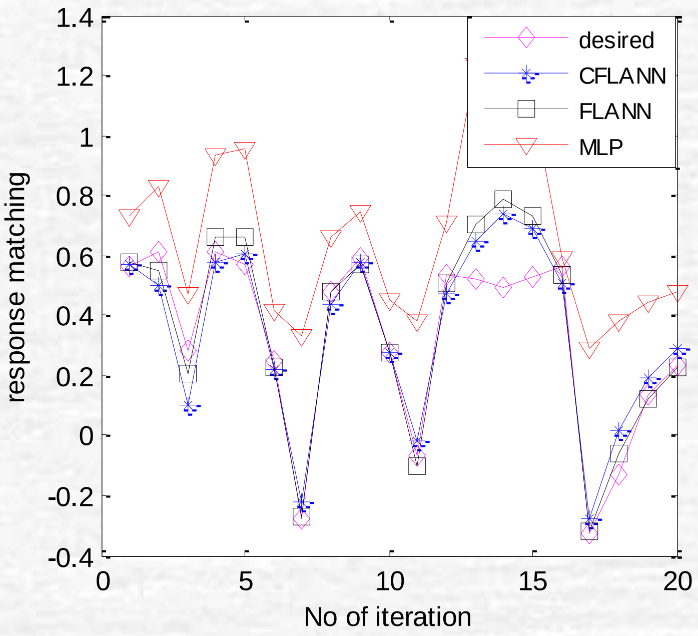
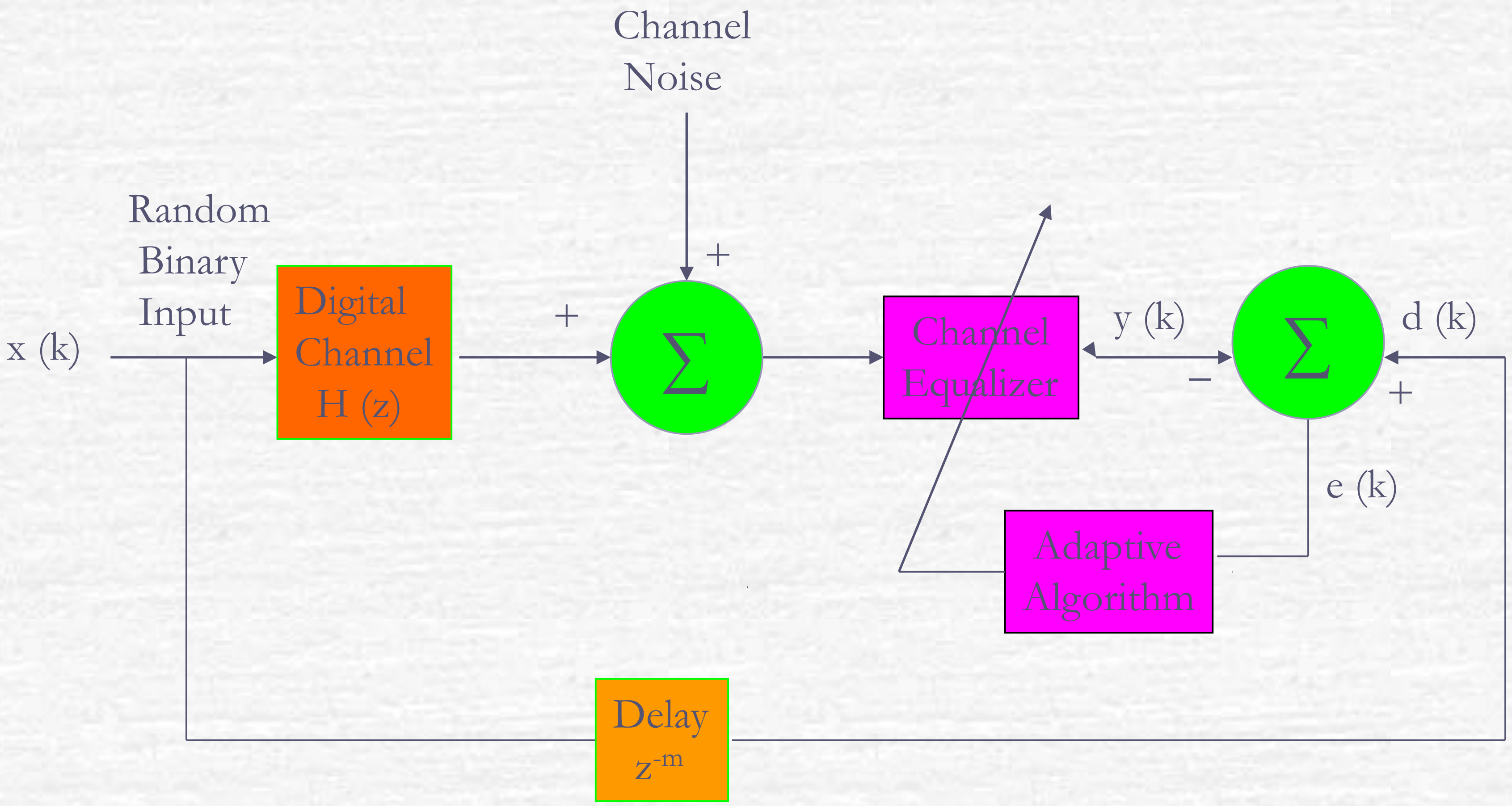


Fig. 6(j)

Block diagram for channel equalization



Channel Equalization

Fig. 7(a):y=x
Fig. 7(b): y= tanh(x)

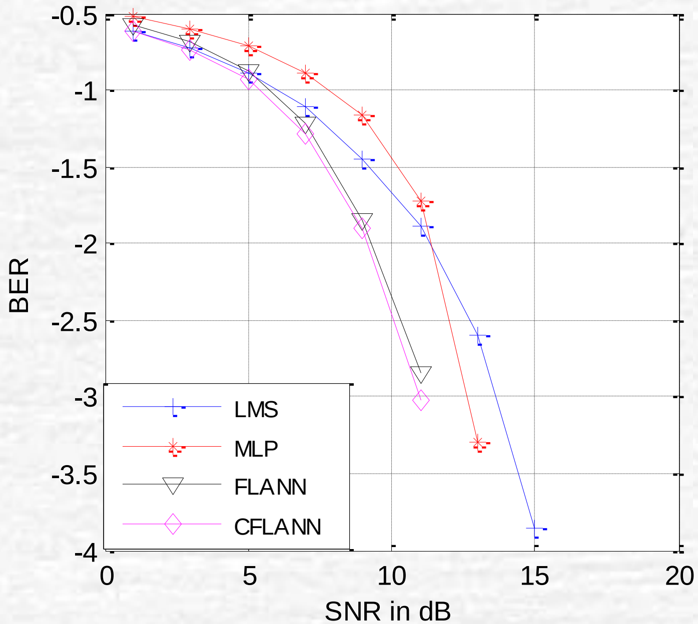


Fig. 7(a)

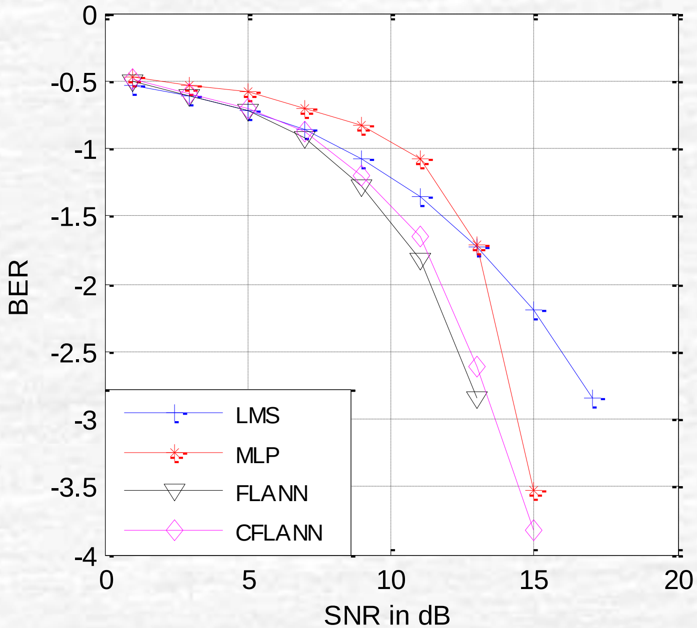


Fig. 7(b)

Channel Equalization

Fig. 7(c): y = x + 0.2 * (x^2) - 0.1*(x^3)
Fig. 7(d):y= x - 0.9 * (x.^3)

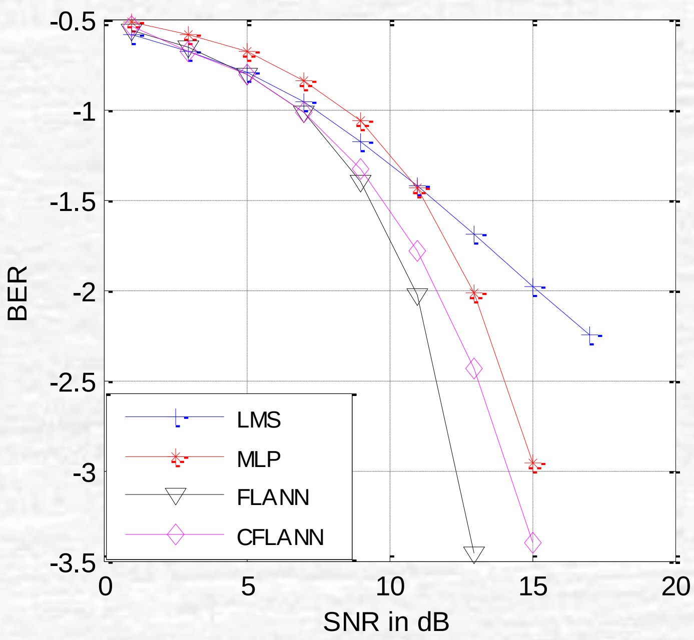


Fig. 7(c)

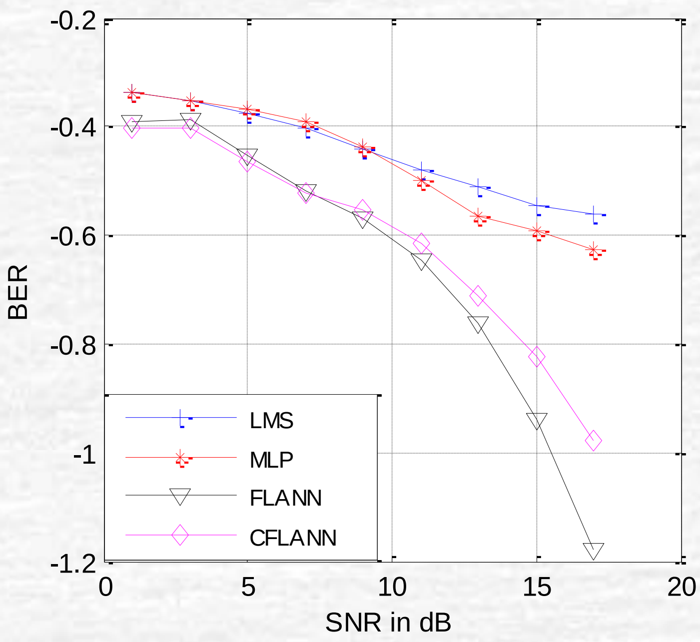


Fig. 7(d)

Channel Equalization

Fig. 7(e): y= x + 0.2 * (x^2) - 0.1 * (x^3)+ 0.5* cos(pi*x)

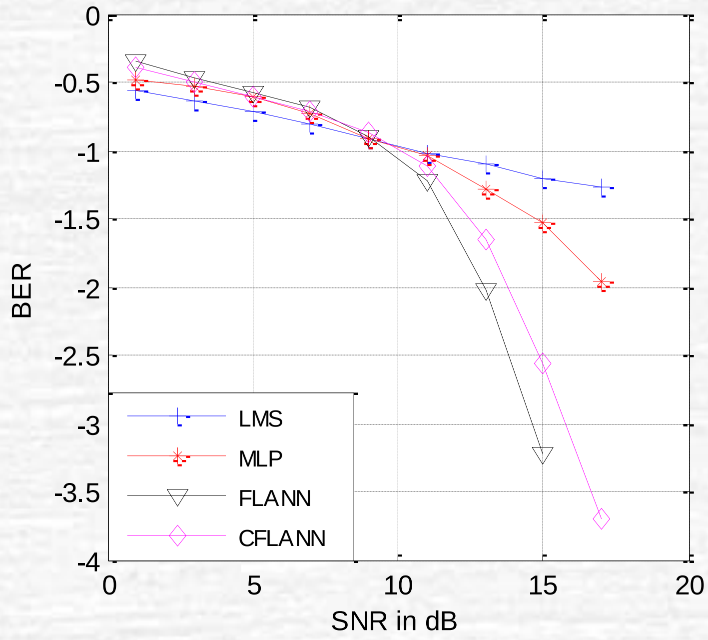


Fig. 7(e)

Conclusion:

The Chebyshev FLANN (CFLANN) structure is a better candidate in comparison to other nonlinear structures like MLP and FLANN in terms of

- better and faster convergence
- less mathematical complexity
- less no input sample.

Unlike other algorithms it is observed that the CFLANN structure exhibits **learning-while functioning** instead of **learning then functioning** hence it is suitable for **On-line identification.**

THANK YOU