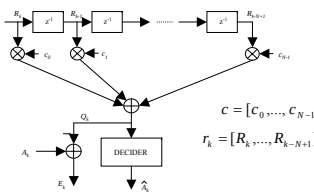
Equalization systems

LTE

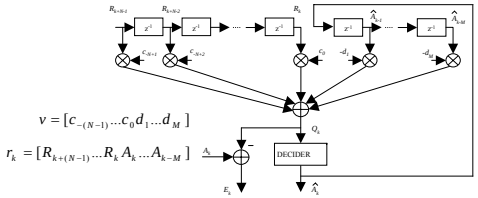


$c = [c_0, \ldots, c_{N-1}]$

$r_k = [R_k, \ldots, R_{k-N+1}]$

LMS: $Q_k = cr_k' \qquad E_k = A_k - Q_k \qquad c_{k+1} = c_k + \beta E_k r_k$

DFE

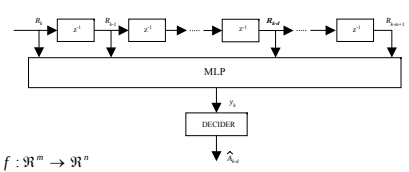


$v = [c_{-(N-1)} \ldots c_0 d_1 \ldots d_M]$

$r_k = [R_{k+(N-1)} \ldots R_k A_k \ldots A_{k-M}]$

LMS: $E_k = A_k - Q_k = A_k - v_k r_k' \qquad v_{k+1} = v_k + \beta E_k r_k$

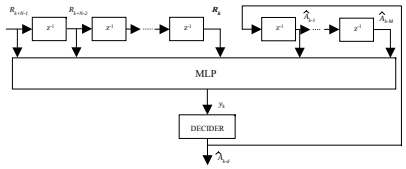
Forward MLP



$f : \Re^m \rightarrow \Re^n$

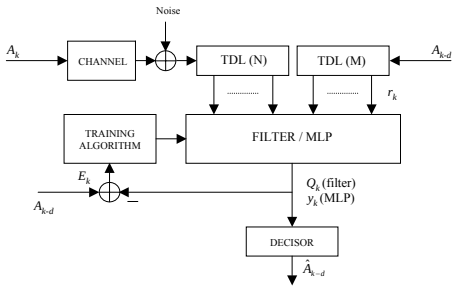
$y_l = [f(r_k)]_l = s\left(\sum_{j=1}^{h} w_{jl}^2 s\left(\sum_{i=1}^{m} w_{ij}^1 r_i + w_j^1\right) + w_l^2\right) \qquad l=1 \ldots n$

Feedback MLP



$r_k=[r_1 \ldots r_m]$ input vector at period (time) k,
$y_k=[y_1 \ldots y_n]$ output vector at period (time) k,
w_{ij}^1 e w_{jl}^2 first and second layer weights

Learning phase



training set → Training sequence $tr=[A_1, A_2, \ldots, A_k, \ldots, A_{Lt}]$

At each learning iteration k:

- *output channel sampling* r_k
- *estimation of the symbol* Q_k *(filter) or* y_k *(MLP)*
- *comparison with the target value* A_{k-d}
- *output error computation* E_k
- *weight update*

DFE and feedback MLP: feedback outputs → target A_{k-d}

During training: at each learning iteration k, computation of the MSE (E_k)

FILTER → $MSE_k = (Q_k - A_k)^2$

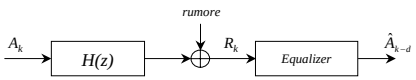
MLP → $MSE_k = \frac{(y_k - t_k)(y_k - t_k)^T}{n}$ (*n* uscite)

Performance evaluation: BER $\qquad BER = \frac{\sum_{i=1}^{r} d_i / nc}{r}$

d_i: # of errors (not correctly estimated symbols) at the receiver
nc: transmitted symbols
r: # of transmission experiments

Base band 2 levels PAM channel

Discrete time model



H(z) models:

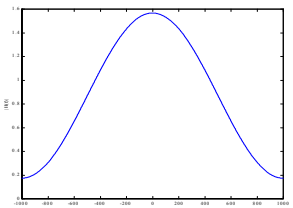
- transmitter filter
- channel
- receiver filter

Input stream: random stream of equal-probability and statistically indipendent symbols A_k: {-1,1}

AWGN noise with variance σ^2 → $SNR = \frac{\bar{f} \cdot \bar{f}^T}{\sigma^2}$ at the equalizer input

$\bar{f}$ H(z) coefficients vector

$H(z) = 0.3482 + 0.8704\, z^{-1} + 0.3482\, z^{-2}$



Frequency response magnitude

Decider: signum function

MLP: 10 hidden neurons, 1 output neuron

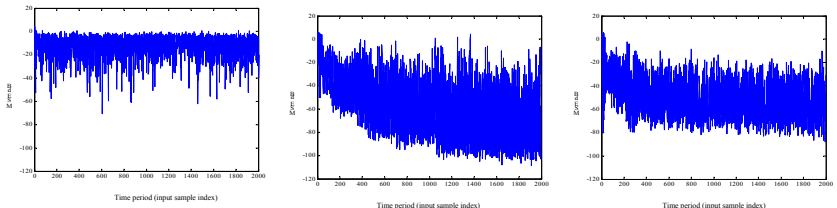
MLP → BP: η=0.07, η_b=0.05, α=0.3, a_b=0

DFE → LMS: β=0.035

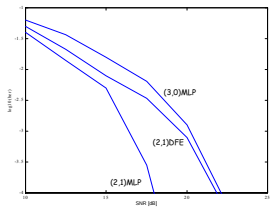
✓ DFE: reference structure for the comparison with the NN

DFE - MLP performance comparison

Training sequence length: 2000 samples

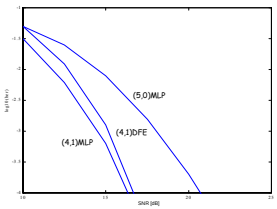


(2,1)DFE **(3,0)MLP** **(2,1)MLP**



m=3

Higher order equalizer

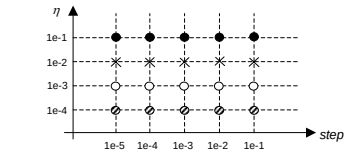


m=5

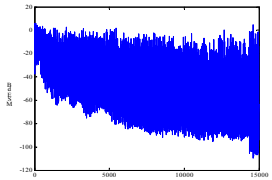
NNs give better performance with respect to DFE filters when the equalizer order is close to the channel filter order

BP - WP comparison

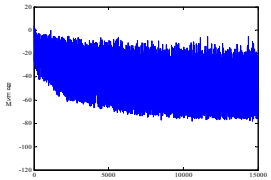
WP: training convergence analysis



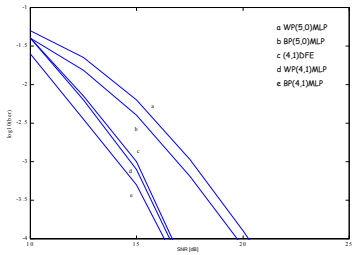
$step=10^{-3}$
$\eta=10^{-2}$
$Lt=15000$



(3,0)MLP



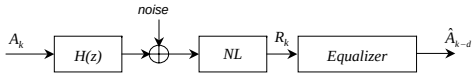
(2,1)MLP



The performance are comparable unless the training sequence of WP length is 5-fold the one of BP.

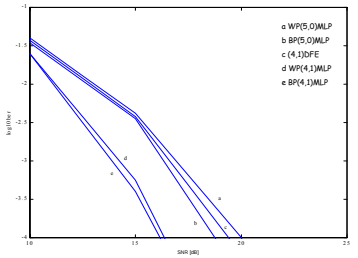
M. Valle

Non-linear channel

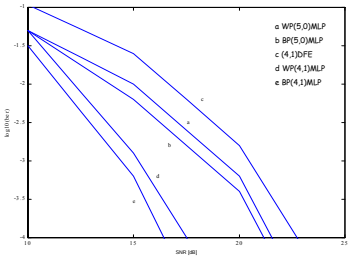


$$H(z) = 0.3482 + 0.8704\,z^{-1} + 0.3482\,z^{-2}$$

NL = hyperbolic tangent: *g(x)=tanh(x)*



NL = exponential: *g(x)=exp(x)-1*



In case of non-linear systems/channels, NNs achieve better performance in particular for low SNR values

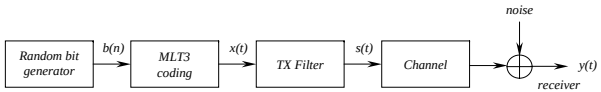
M. Valle

Fast Ethernet: 100BASE-TX

100BASE-TX (IEEE 802.3):

- ✓ 2 twisted pairs CAT5 (1 per TX e 1 per RX)
- ✓ input bit stream @ 100 Mbps, 4b/5b coding
- ✓ input data stream b(n) @ 125 Mbps, MLT3 coding

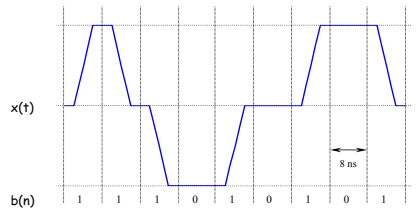
MATLAB model:

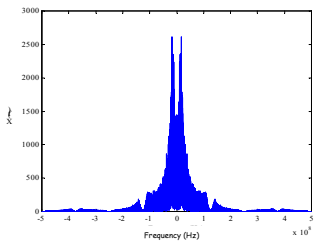


- ✓ *b(n)* random equiprobable bit stream @ 125Mbps
- ✓ MLT3 coding: 3-level (1, 0,-1) baseband signal

1 ➡ transition

0 ➡ no-transition



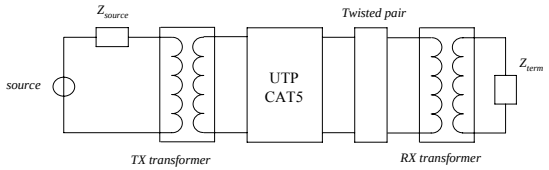


✓TX filter: 4th order BESSEL

M. Valle

Channel model

- ✓ Channel (ANSI/TIA/EIA-568-A):

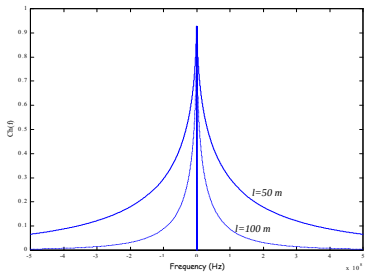


UTP CAT5

$$H_{dB}(f,l,C) = A_0 \cdot \left(\frac{2j \cdot f}{F_0}\right)^{0.5} \cdot \frac{l}{X_0} \cdot (1 + 0.003 \cdot (C - 20))$$

$F_0 = 31.25 \cdot 10^6$	*Hz*	*l*: twisted pair length
$A_0 = -36$	*dB*	*C*: temperature
$X_0 = 305$	*m*	

Channel frequency response magnitude



- ✓ @ 100 MHz ➡ 0.3 l= 50 m, 0.1 l= 100 m
- ✓ transformer ➡ high pass (low cut-off freq = 50 kHz)

✓AWGN noise

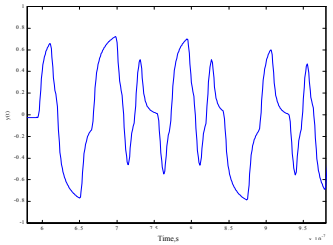
M. Valle

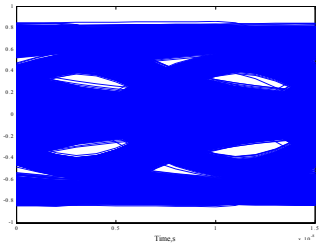
Waveforms

Channel output

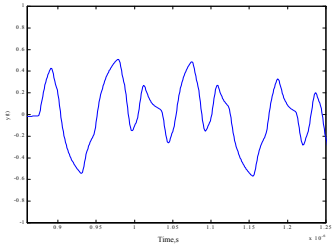
Eye diagram

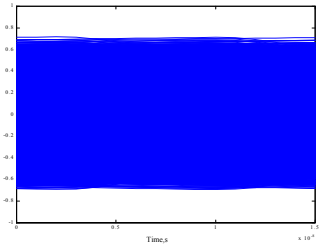
l = 50 m



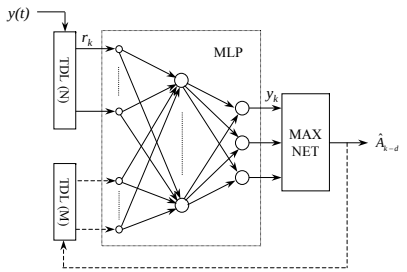


l = 100 m





Equalizer and training phase



3-level MLT3 signal → 3 output neurons

Target coding:

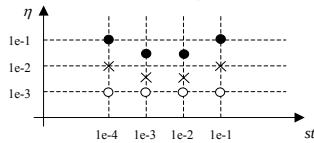
level 1 → y_k=[1,-1,-1]

level 0 → y_k=[-1,1,-1]

level -1 → y_k=[-1,-1,1]

WP training convergence analysis:

5-2-3 topology, d=2

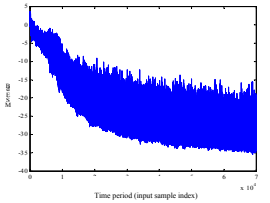


● N convergence × convergence

○ Slow convergence

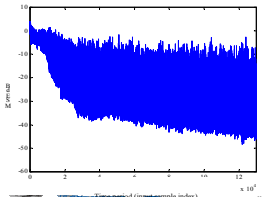
step = 10^{-2}

$\eta = 5 \cdot 10^{-3}$



l = 50 m

Lt=70000



l = 100 m

Lt=130000

Performance

BER value averaged on 50 different 10000-symbol sequences

Analysis on the equalizer structure

- ✓ input neurons (i.e. equalizer order)*(m)* → Non convergence if $m > 10$
- ✓ hidden neurons *(h)* → No performance enhancement if $h > 2$
- ✓ extimation delay *(d)* → Performance enhancement when d approaches 0

Forward MLP equalizer

- ✓ TDL 5 taps with 8 ns delay
- ✓ MLP topology 5 × 2 × 3
- ✓ Estimation delay d=0

Training

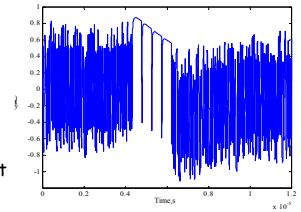
- ✓ Weight Perturbation $\eta = 5 \cdot 10^{-3}$ *step*=10^{-2}
- ✓ Training sequence ≥ 70000 l= 50 m and training sequence ≥ 130000 l = 100 m

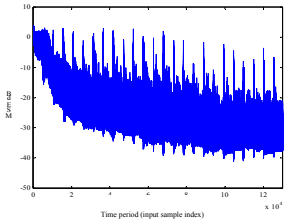
Performance

- ✓ BER < 10^{-4} @ SNR ≥ 30 dB, l ∈ [0÷100] m
- ✓ BER=10^{-3} @ SNR=20 dB, l=50 m
- ✓ BER=10^{-3} @ SNR=25 dB, l=100 m

Non-ideal effects: baseline wander and jitter

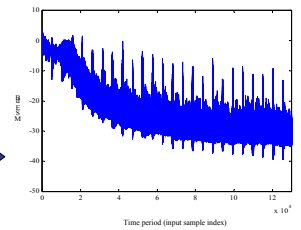
baseline wander: dc cut off due to transformer + long transmitted zero-sequeces →





Training with baseline wander

✓ the peaks of the MSE indicates that the network cannot learn the level transitions



One more input signal which takes into account the baseline wander

Training with on more signal which is the output of a low pass filter whose input is the input sequence

jitter: non-ideal sampling time

➢ random sampling in an interval of 2d length centered on the ideal sampling time

- ✓ BER < 10^{-4} if $\delta \leq 1$ ns → Sampling circuit specifications
- ✓ BER < 10^{-3} if $\delta \leq 2$ ns