‘dfecolor’ Program Tutorial

Jungsub Byun

The previous developments of discrete-time equalization structures presumed that the equalizer could exist over an infinite interval in time. However equalization filters, w_k, are almost exclusively realized as finite impulse- response (FIR) filters in practice. Usually these structures have better numerical properties than IIR structures. In real implementations, non-causal filters are implemented by truncating and shifting the response and adding appropriate delay. The truncated and shifted response may not be optimal for that system delay. Hence, it is useful to find the optimal FIR implementation for a given delay, because FIR filters are more robust to quantization errors, stable, and easier for adaptive equalization. A continuous time filter is much harder to design accurately than a digital domain filter. Precise sampling is necessary at the output of matched filter to obtain optimum performance. Therefore most advanced transmission schemes implement the matched filter in digital domain.

A linear FIR equalizer uses a linear FIR feedforward filter to estimate the transmitted signal x_k.
System constraint I - FIR filter length
– The feedforward filter memory is limited to N_f symbol periods.
– This means FIR filter can only use the $N_f\,l$ samples given by

$$\mathbf{Y_k} \triangleq \begin{bmatrix} \mathbf{y_k} \\ \mathbf{y_{k-1}} \\ \vdots \\ \mathbf{y_{k-N_f+1}} \end{bmatrix}$$

System constraint II - Delay Δ
– At time instant k, the filter output should be the best estimate of $x_{k-\Delta}$
The estimate $\tilde{x}_{k-\Delta} = w\boldsymbol{Y}_k$ where w is the $N_f\,l$ tap linear filter expressed as a row vector.
w can be optimized to
– Maximize SNR_R - gives FIR MMSE-LE.
– Ignore noise and cancel ISI as far as possible - gives FIR ZFE.
– Because of FIR constraints FIR ZFE cannot completely eliminate ISI, unlike its unconstrained version.
- The output noise variance after sampler is $\frac{N_o}{2}l$ per dimension.

- Higher l implies better equalizer performance and lower sensitivity to sampling phase error. On the other hand, higher l means higher system complexity.
- This setup with digital domain matched-filter and higher sampling rate is called fractionally spaced equalizer (FSE) setup.

Minimum-Mean-Square-Error Decision Feedback Equalizer (MMSEDFE) makes use of previous decisions in attempting to estimate the current symbol (with an SBS detector). Any “trailing” intersymbol interference caused by previous symbols is reconstructed and then subtracted. The feedforward filter will try to shape the channel output signal so that it is a causal signal. The feedback section will then subtract (without noise enhancement) any trailing ISI.

For causality, the designer picks a channel-equalizer system delay $\Delta \cdot$ T symbol periods, $\Delta = (\upsilon + Nf)/2$ with the exact value Δ being of little consequence unless the equalizer length Nf is very short in the linear equalizer. The delay Δ is important in finite-length design because a non-causal filter cannot be implemented, and the delay Δ allows time for the transmit data symbol to reach the receiver. With infinite-length filters, the need for such a delay does not enter the mathematics because the infinite-length filters are not realizable in any case, so that infinite-length analysis simply provides best bounds on performance. Δ is approximately the sum of the channel and equalizer delays in symbol periods.

For showing various effects on various channels in Equalization part, the 'dfecolor' program Tutorial consists of

[1] Function description of dfecolor,

[2] 'dfecolor' examples,

[3] FIR equalizer performance (SNR) for various channels versus number of (feedforward tap, feedback tap, or delay factors) with(MMSE- DFE)/without(MMSE-LE) feedback tap,

[4] Comparison between MMSE-DFE and MMSE-LE - FIR equalizer performance (SNR) for 1+a*D^{-1} versus varying from 1-1*D^{-1} to 1+1*D^{-1} channels with fixed SNRmfb/noise variance (comparing between MMSE-DFE and MMSE-LE),

[5] Function description of dferake and examples.

[1] Function description of dfecolor, change Nf, Nb, and Δ for finding max SNR.

function [dfseSNR,w_t]=dfecolor(l,p,nff,nbb,delay,Ex,noise);

l = oversampling factor per symbol period

p = pulse response, oversampled at l (size)

nff = number of feedforward taps per symbol period

nbb = number of feedback taps per symbol period

delay = delay of system <= nff+length of p-2 - nbb for casality, Δ_{opt}

Ex = average energy of signals, Ex_bar

noise = noise autocorrelation vector (size l*nff), l[$N_0/2$ 0 0 0 … 0]

outputs: dfseSNR = equalizer SNR, unbiased in dB and w_t=[Wf, b]

This program has come to be used throughout the industry to compute/project equalizer performance. Difficult transmission channels may require large numbers of taps and considerable experimentation to find best settings.

[2] 'dfecolor' examples

DFE with Nf=2, delay = 1, Nb=1

[snr, W] = dfecolor(1, [.9 1], 2, 1, 1, 1, .181*[1 zeros(1,1)])

snr = 7.3911 , W = 0.1556 0.7668 -0.7668

DFE with Nf=3, delay=1, Nb=1

[snr, W] = dfecolor(1, [.9 1], 3, 1, 1, 1, .181*[1 zeros(1,2)])

snr = 7.3911

W = 0.1556 0.7668 0.0000 -0.7668

DFE with Nf=3, delay=2, Nb=1

[snr, W] = dfecolor(1, [.9 1], 3, 1, 2, 1, .181*[1 zeros(1,2)])

snr = 7.9148

W = -0.1077 0.2382 0.6919 -0.6919

DFE with Nf=7 delay=6, Nb=1
[snr, W] = dfecolor(1, [.9 1], 7, 1, 6, 1, .181*[1 zeros(1,6)])
snr = 8.3447
W = -0.0184 0.0408 -0.0718 0.1180 -0.1893 0.3008 0.6350 -0.6350

MMSE-LE for Nf=5, different delays
[snr, W] = dfecolor(1, [.9 1], 5, 0, 1, 1, .181*[1 zeros(1,4)])
snr = 3.5936
W = 0.3069 0.4321 -0.2628 0.1493 -0.0675
[snr, W] = dfecolor(1, [.9 1], 5, 0, 2, 1, .181*[1 zeros(1,4)])
snr = 4.6558
W = -0.1973 0.4365 0.3427 -0.1947 0.0880

[snr, W] = dfecolor(1, [.9 1], 5, 0, 3, 1, .181*[1 zeros(1,4)])
snr = 4.9568
W = 0.1296 -0.2867 0.5046 0.2814 -0.1272

[snr, W] = dfecolor(1, [.9 1], 5, 0, 4, 1, .181*[1 zeros(1,4)])
snr = 4.6838
W = -0.0894 0.1977 -0.3480 0.5721 0.1934

[snr, W] = dfecolor(1, [.9 1], 5, 0, 5, 1, .181*[1 zeros(1,4)])
snr = 3.6663
W = 0.0681 -0.1507 0.2652 -0.4360 0.6994

DFE with different Nf taps, different delays, Nb=1
[snr, W] = dfecolor(1, [.9 1], 2, 1, 1, 1, .181*[1 zeros(1,1)])
snr = 7.3911
W = 0.1556 0.7668 -0.7668

[snr, W] = dfecolor(1, [.9 1], 3, 1, 2, 1, .181*[1 zeros(1,2)])
snr = 7.9148
W = -0.1077 0.2382 0.6919 -0.6919

[snr, W] = dfecolor(1, [.9 1], 4, 1, 3, 1, .181*[1 zeros(1,3)])
snr = 8.1689
W = 0.0708 -0.1567 0.2758 0.6577 -0.6577

[snr, W] = dfecolor(1, [.9 1], 5, 1, 4, 1, .181*[1 zeros(1,4)])
snr = 8.2798
W = -0.0456 0.1008 -0.1774 0.2917 0.6433 -0.6433

[snr, W] = dfecolor(1, [.9 1], 6, 1, 5, 1, .181*[1 zeros(1,5)])
snr = 8.3259
W = 0.0290 -0.0642 0.1131 -0.1859 0.2982 0.6374 -0.6374

[snr, W] = dfecolor(1, [.9 1], 7, 1, 0, 1, .181*[1 zeros(1,6)])
snr = 6.5081
W = 0.9082 0.0000 -0.0000 0 -0.0000 0.0000 -0.0000 -0.9082

[snr, W] = dfecolor(1, [.9 1], 7, 1, 1, 1, .181*[1 zeros(1,6)])
snr = 7.3911
W = 0.1556 0.7668 0.0000 -0.0000 -0.0000 0.0000 0.0000 -0.7668

[snr, W] = dfecolor(1, [.9 1], 7, 1, 3, 1, .181*[1 zeros(1,6)])
snr = 8.1689
W = 0.0708 -0.1567 0.2758 0.6577 0 0 0 -0.6577

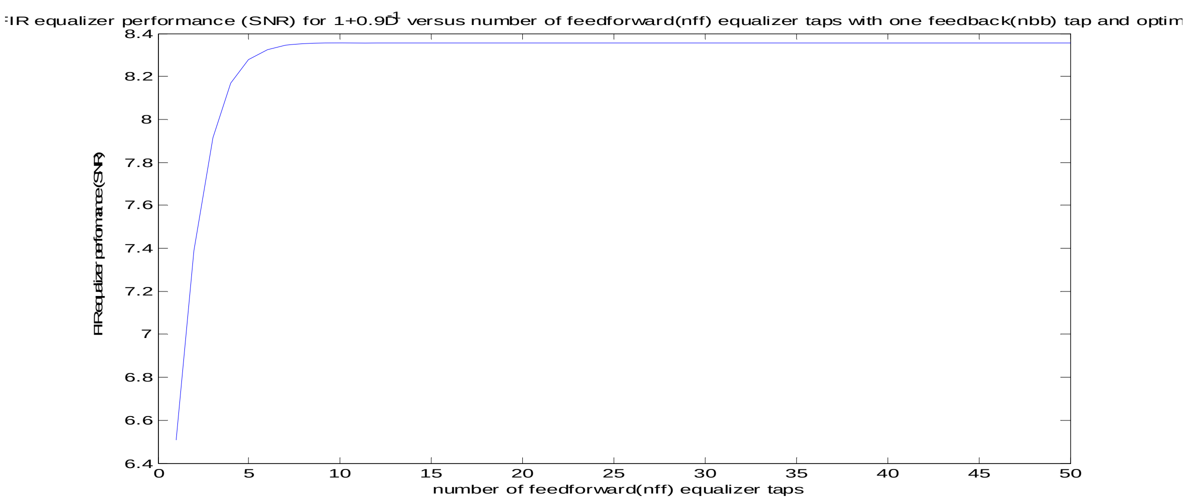
[snr, W] = dfecolor(1, [.9 1], 7, 1, 4, 1, .181*[1 zeros(1,6)])
snr = 8.2798
W = -0.0456 0.1008 -0.1774 0.2917 0.6433 -0.0000 -0.0000 -0.6433

[snr, W] = dfecolor(1, [.9 1], 7, 1, 5, 1, .181*[1 zeros(1,6)])
snr = 8.3259
W = 0.0290 -0.0642 0.1131 -0.1859 0.2982 0.6374 0.0000 -0.6374

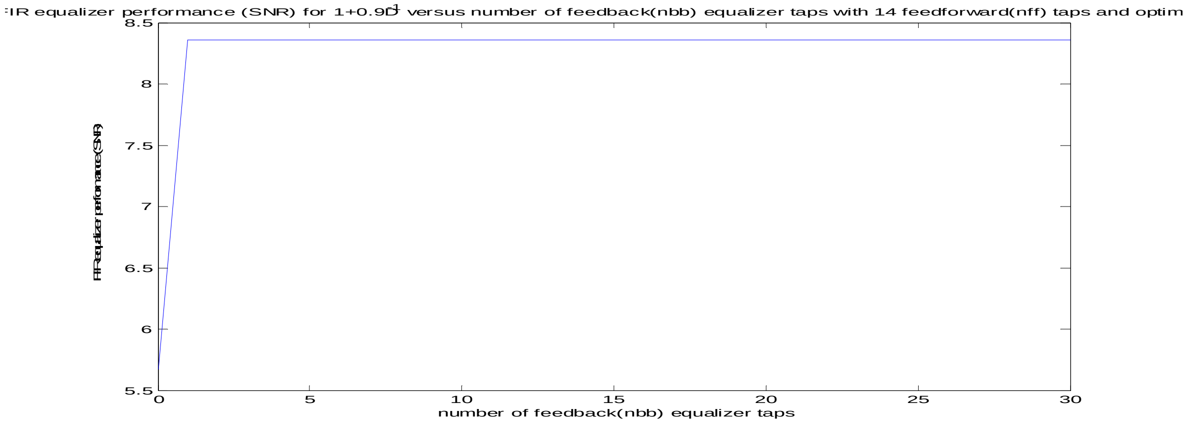
[snr, W] = dfecolor(1, [.9 1], 7, 1, 6, 1, .181*[1 zeros(1,6)])
snr = 8.3447
W = -0.0184 0.0408 -0.0718 0.1180 -0.1893 0.3008 0.6350 -0.6350

[3] FIR equalizer performance (SNR) for various channels versus number of (feedforward(nff) tap, feedback(nbb) tap, or delay factors) with(MMSE-DFE)/without(MMSE-LE) feedback tap

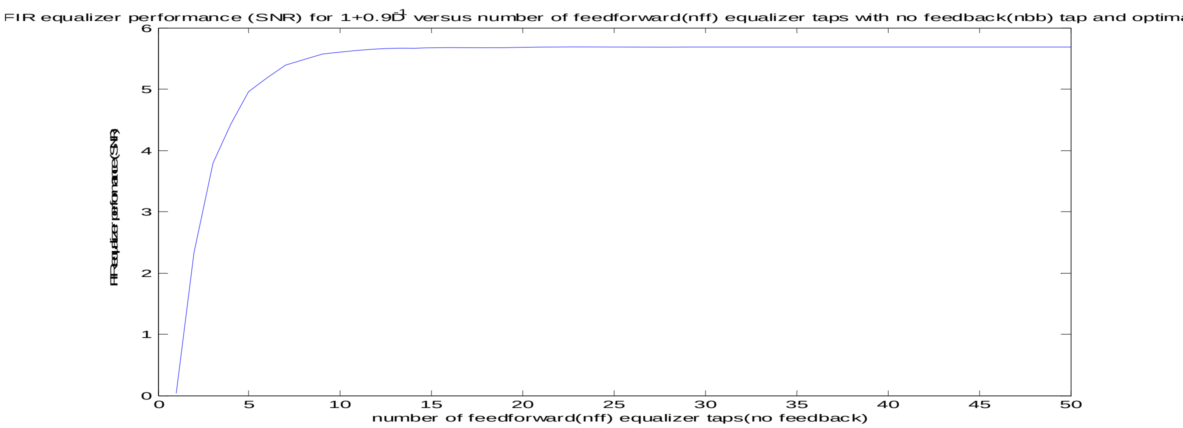
a. Fix SNRmfb=10dB and noise variance=0.181 for $1+0.9D^{-1}$channel (Low pass channel)
FIR equalizer performance (SNR) for $1+0.9D^{-1}$ versus number of feedforward(nff) equalizer taps with one feedback(nbb) tap and optimal delay (nbb=1).



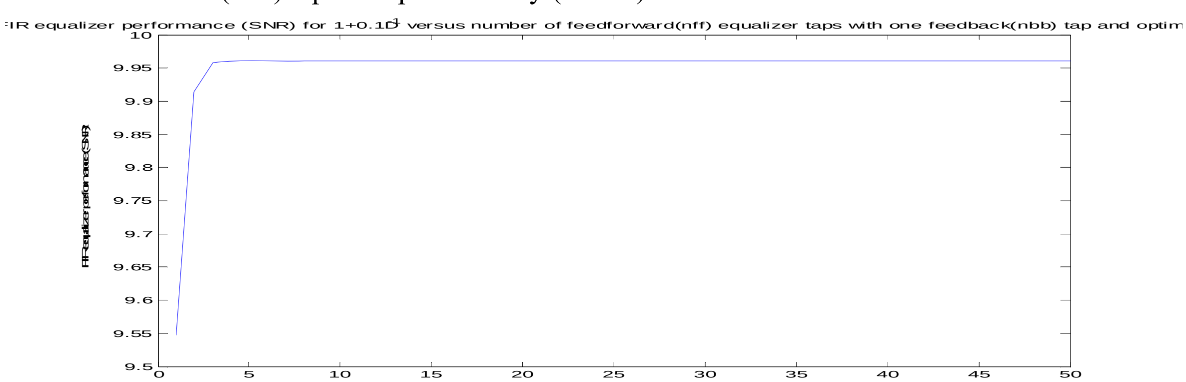
b. Fix SNRmfb=10dB and noise variance=0.181 for $1+0.9D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.9D^{-1}$ versus number of feedback(nbb) equalizer taps with 14 feedforward(nff) taps and optimal delay (nff=14).



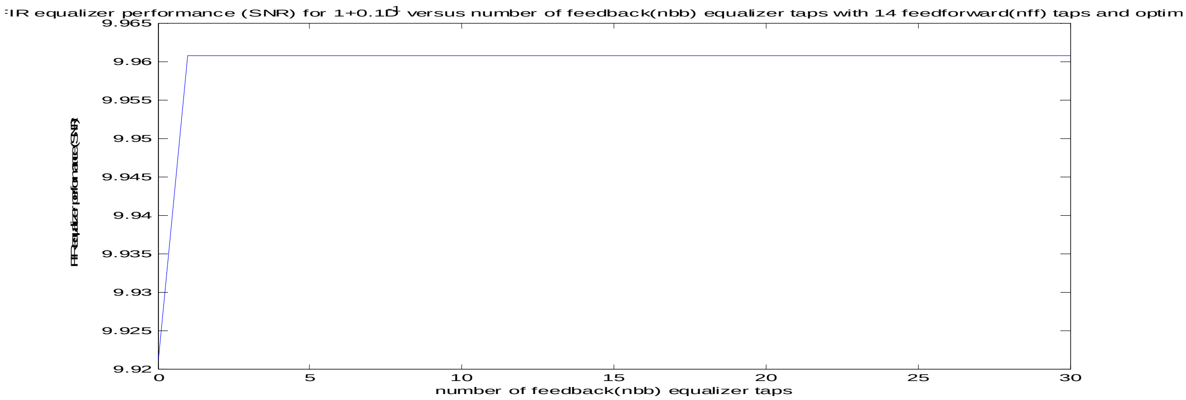
c. Fix SNRmfb=10dB and noise variance=0.181 for $1+0.9D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.9D^{-1}$ versus number of feedforward(nff) equalizer taps with no feedback(nbb) tap and optimal delay(nbb=0)



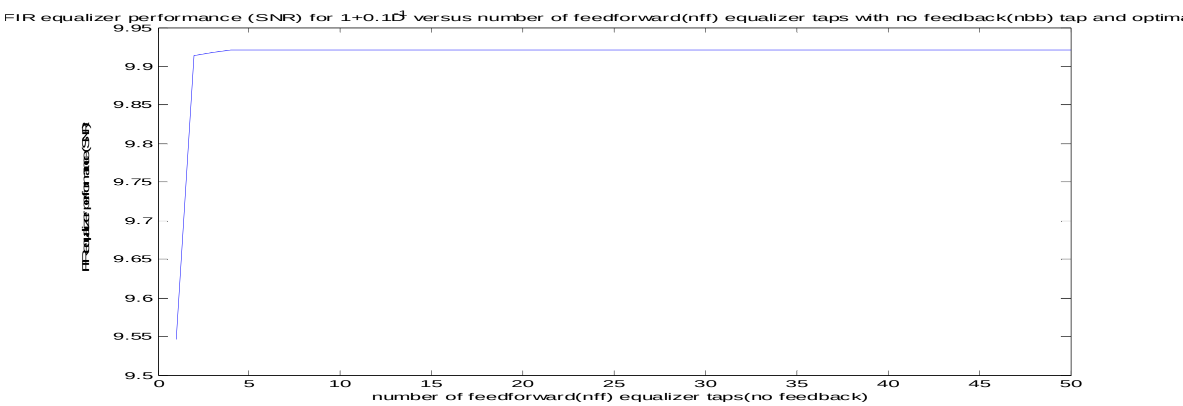
d. Fix SNRmfb=10dB and noise variance=0.101 for $1+0.1D^{-1}$ channel ['less ISI']
FIR equalizer performance (SNR) for $1+0.1D^{-1}$ versus number of feedforward(nff) equalizer taps with one feedback(nbb) tap and optimal delay (nbb=1).



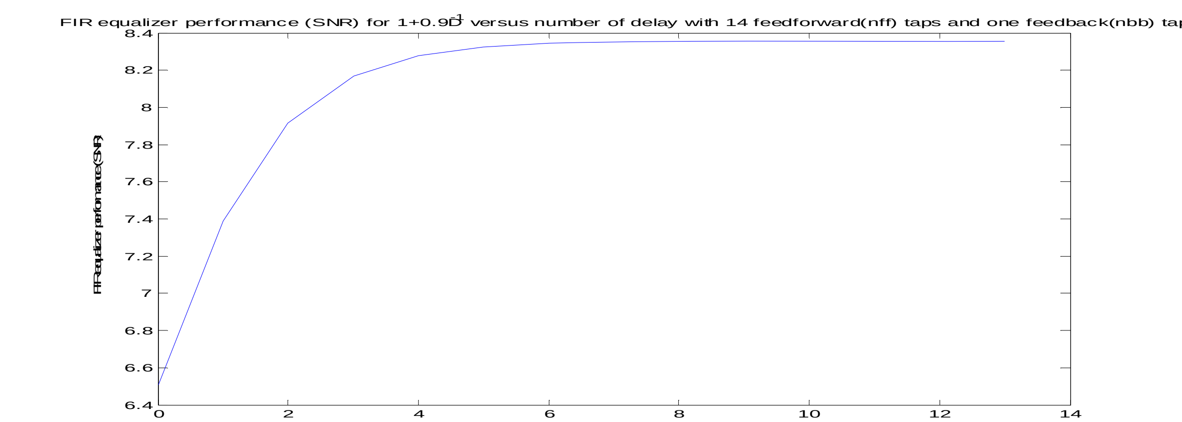
e. Fix SNRmfb=10dB and noise variance=0.101 for $1+0.1D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.1D^{-1}$ versus number of feedback(nbb) equalizer taps with 14 feedforward(nff) taps and optimal delay (nff=14).



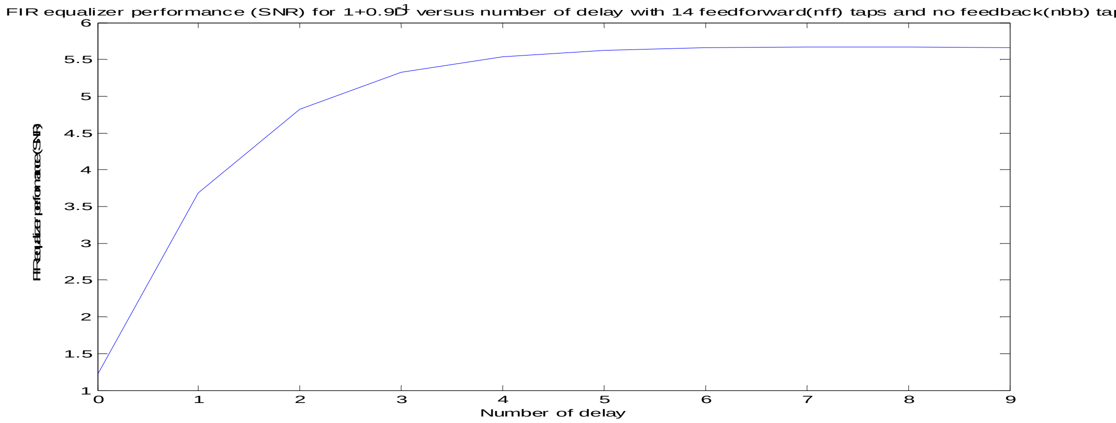
f. Fix SNRmfb=10dB and noise variance=0.101 for $1+0.1D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.1D^{-1}$ versus number of feedforward(nff) equalizer taps with no feedback(nbb) tap and optimal delay (nbb=0).



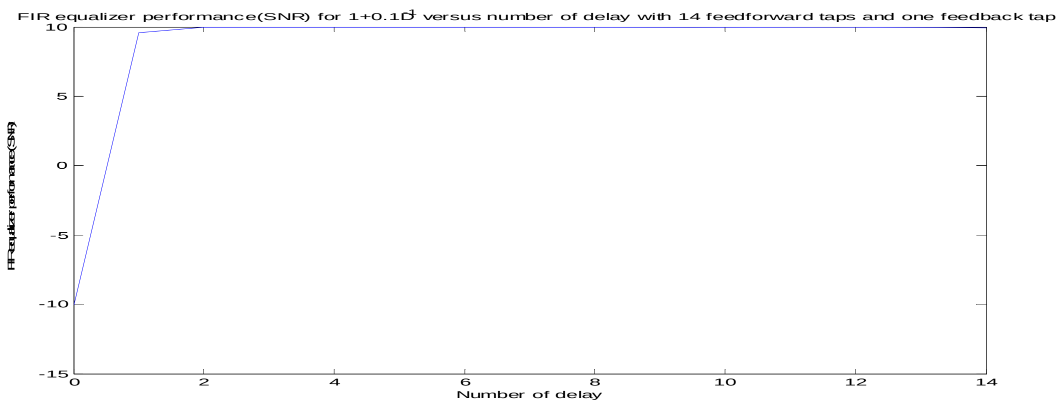
g. Fix SNRmfb=10dB and noise variance=0.181 for $1+0.9D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.9D^{-1}$ versus number of delay with 14 feedforward(nff) taps and one feedback(nbb) tap (nff=14, nbb=1, MMSE-DFE).



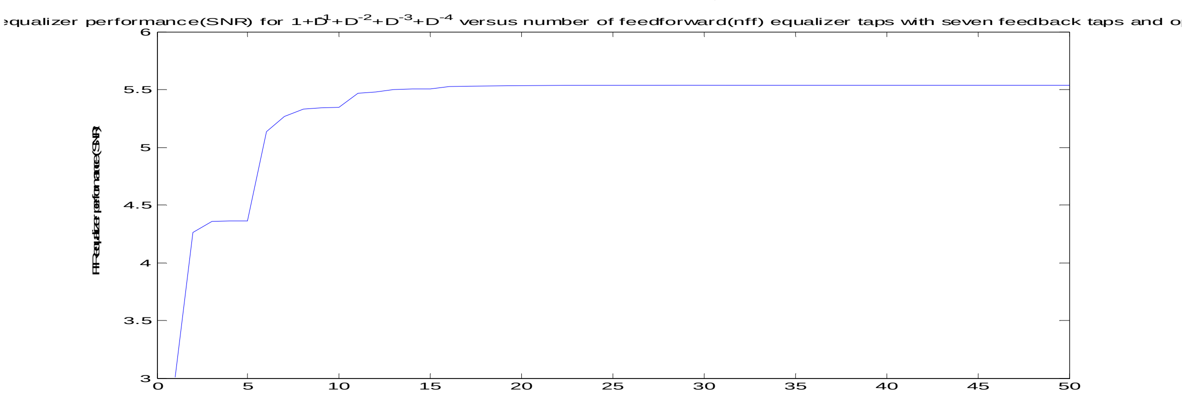
h. Fix SNRmfb=10dB and noise variance=0.181 for $1+0.9D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.9D^{-1}$ versus number of delay with 14 feedforward(nff) taps and one feedback(nbb) tap (nff=14, nbb=0, MMSE-LE).



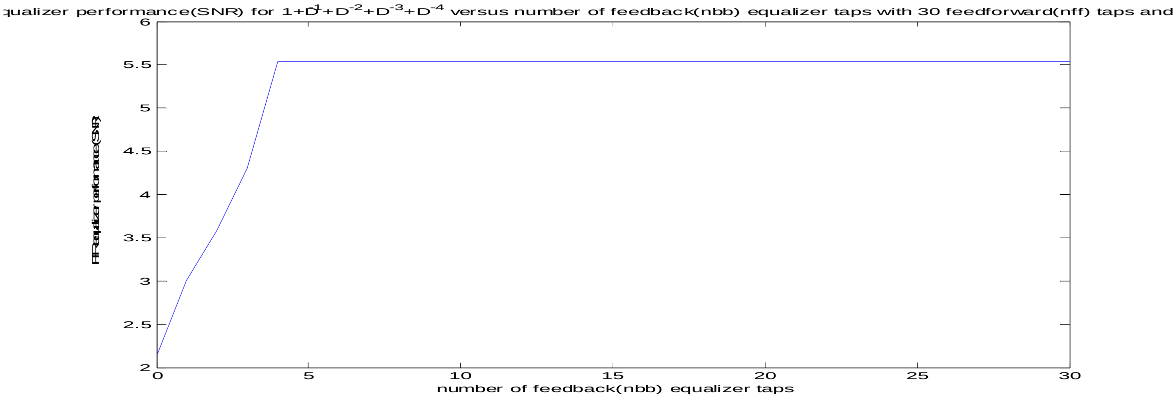
i. Fix SNRmfb=10dB and noise variance=0.101 for $1+0.1D^{-1}$ channel
FIR equalizer performance (SNR) for $1+0.1D^{-1}$ versus number of delay with 14 feedforward(nff) taps and one feedback(nbb) tap (nff=14, nbb=1, MMSE-DFE).



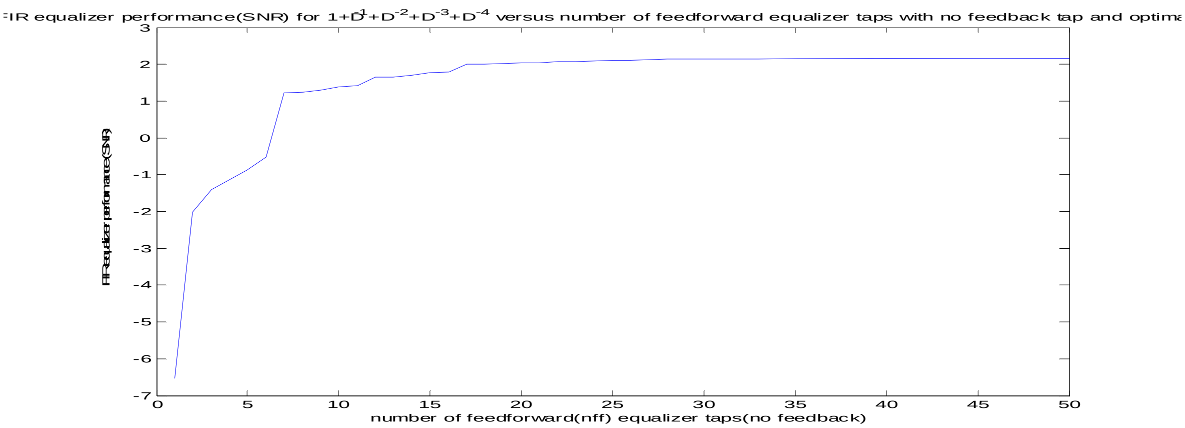
j. Fix SNRmfb=10dB and noise variance=0.5 for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ channel (4 notches channel)
FIR equalizer performance (SNR) for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ versus number of feedforward(nff) equalizer taps with five feedback(nbb) taps and optimal delay (nbb=5).



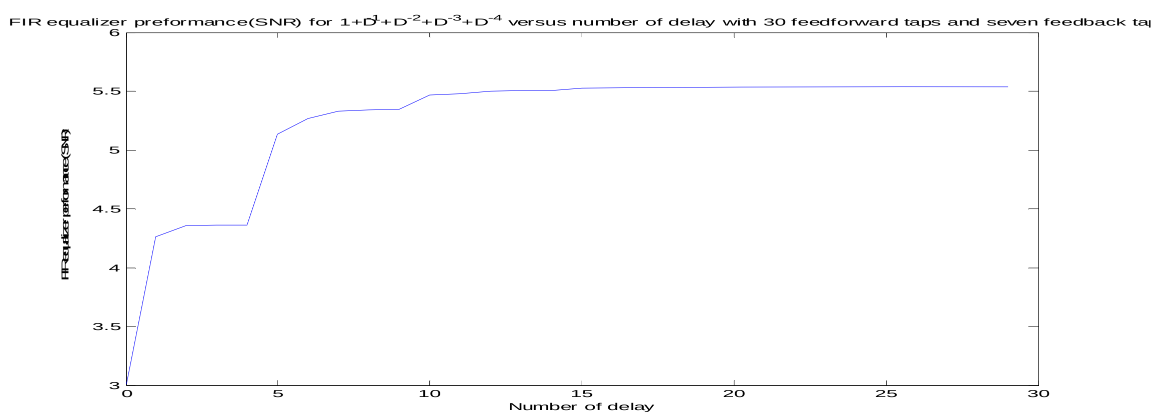
k. Fix SNRmfb=10dB and noise variance=0.5 for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ channel
FIR equalizer performance (SNR) for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ versus number of feedback(nbb) equalizer taps with 30 feedforward(nff) taps and optimal delay (nff=30).



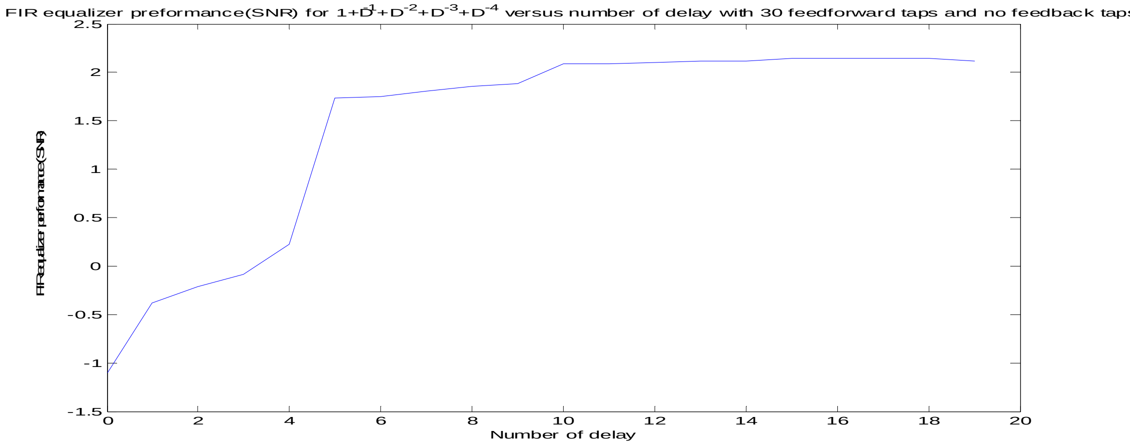
l. Fix SNRmfb=10dB and noise variance=0.5 for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ channel
FIR equalizer performance (SNR) for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ versus number of feedforward(nff) equalizer taps with no feedback(nbb) tap and optimal delay (nbb=0).



m. Fix SNRmfb=10dB and noise variance=0.5 for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ channel
FIR equalizer performance (SNR) for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ versus number of delay with 30 feedforward(nff) taps and five feedback(nbb) taps (nff=30, nbb=5, MMSE-DFE)

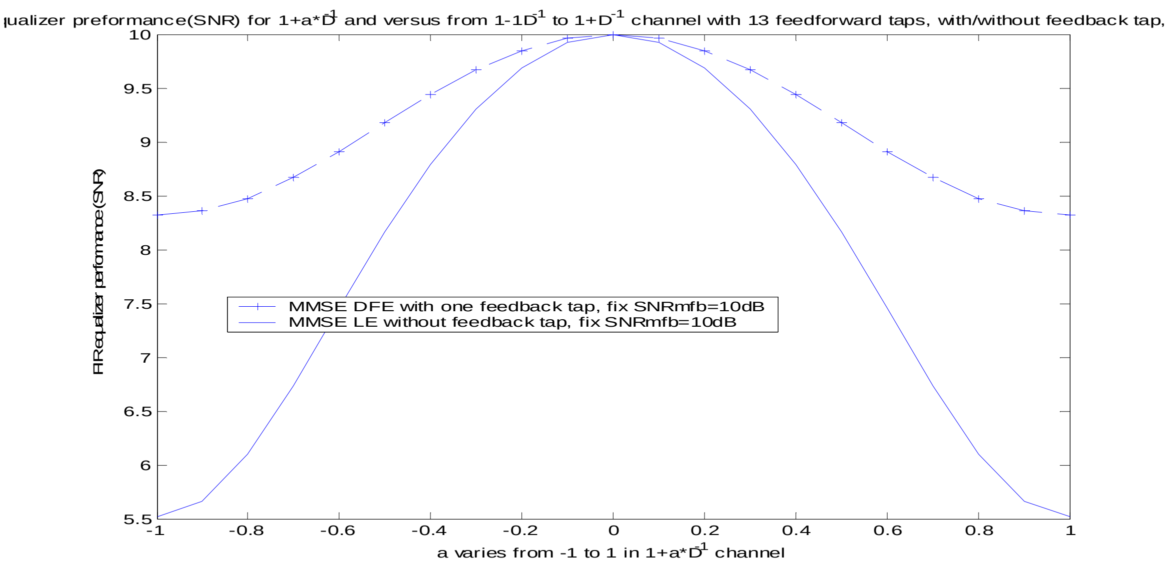


n. Fix SNRmfb=10dB and noise variance=0.5 for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ channel
FIR equalizer performance (SNR) for $1+D^{-1}+D^{-2}+D^{-3}+D^{-4}$ versus number of delay with 30 feedforward(nff) taps and five feedback(nbb) taps (nff=30, nbb=0, MMSE-LE).

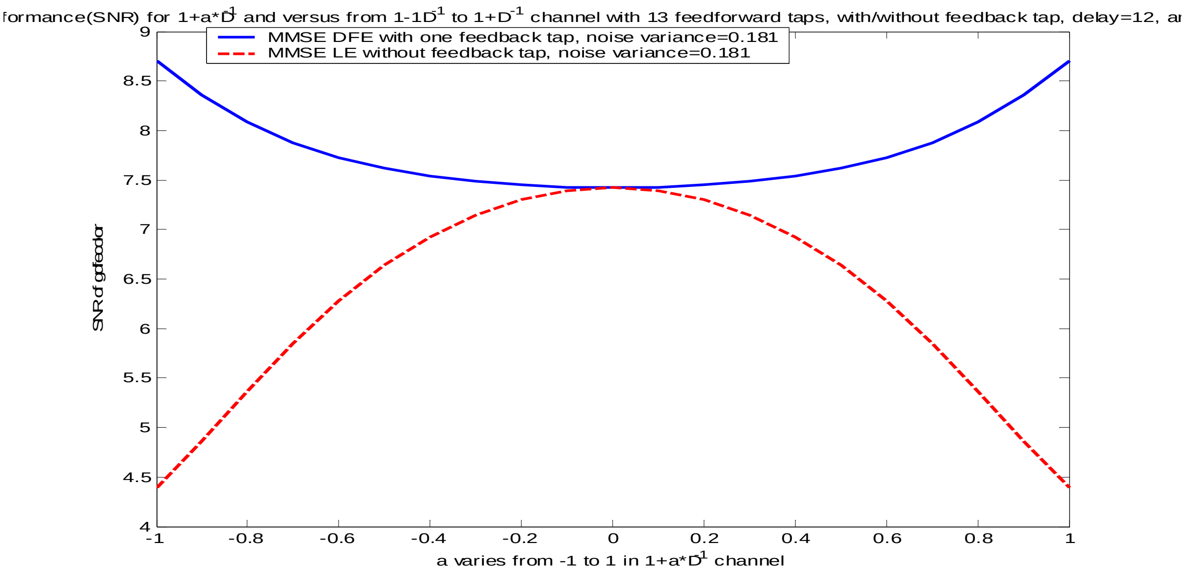


4] Comparison between MMSE DFE and MMSE LE - FIR equalizer performance(SNR) for $1+a*D^{-1}$ versus varying from $1-1*D^{-1}$ to $1+1*D^{-1}$ channels with fixed SNRmfb/noise variance (comparing between MMSE DFE and MMSE LE)

a. Fix SNRmfb=10dB, Noise variance=$(1+a^2)/10$<=[$1+a*D^{-1}$] channel
'FIR equalizer performance (SNR) for $1+a*D^{-1}$ versus from $1-1*D^{-1}$ to $1+1*D^{-1}$ channels with 13 feedforward(nff) taps, with(MMSE DFE)/without(MMSE LE) feedback(nbb) tap, and delay=12'



b. Fix noise variance=0.181
'FIR equalizer performance(SNR) for $1+a*D^{-1}$ versus from $1-1D^{-1}$ to $1+D^{-1}$ channel with 13 feedforward(nff) taps, with(MMSE DFE)/without(MMSE LE) feedback(nbb) tap, delay=12, and noise variance=0.181'



[5] Function description of dferake and examples.

function [dfseSNR,W,b]=dferake(l,p,nff,nbb,delay,Ex,noise);

**** only computes SNR ****

l = oversampling factor

L = No. of fingers in RAKE

p = pulse response matrix, oversampled at l (size), each row corresponding to a diversity path

nff = number of feedforward taps for each RAKE finger

nbb = number of feedback taps

delay = delay of system <= nff+length of p - 2 - nbb

if delay = -1, then choose best delay

Ex = average energy of signals

noise = noise autocorrelation vector (size L*l*nff) - Careful here - nominally only white noise (JC - 2003)

NOTE: noise is assumed to be stationary

outputs:

dfseSNR = equalizer SNR, unbiased in dB

%oversampling factor is one

l = 1

%define p matrix for the example

p = [1 .9; sqrt(1.81/1.64) sqrt(1.81/1.64)*.8]

p = 1.0000 0.9000

1.0506 0.8404

1. Find the DFE with 6 feedforward taps per finger and 1 feedback tap
Try different delays for best performance
(a) Delay = 4
[dfseSNR,W,b]=dferake(1,p,6,1,4,1,.181*[1 zeros(1,11)])
dfseSNR = 11.1498
W = -0.0177 0.0320 -0.0506 0.0758 0.3938 0.0000
-0.0031 0.0126 -0.0239 0.0383 0.4137 -0.0000
b = 0.7020
(b) Delay = 5
[dfseSNR,W,b]=dferake(1,p,6,1,5,1,.181*[1 zeros(1,11)])
dfseSNR = 11.1676
W = 0.0124 -0.0224 0.0354 -0.0530 0.0777 0.3923
0.0021 -0.0088 0.0167 -0.0268 0.0404 0.4121
b = 0.6994
(c) Delay = 3
[dfseSNR,W,b]=dferake(1,p,6,1,3,1,.181*[1 zeros(1,11)])
dfseSNR = 11.1138
W = 0.0252 -0.0456 0.0721 0.3968 0.0000 0.0000
0.0043 -0.0180 0.0340 0.4169 -0.0000 -0.0000
b = 0.7075
Delay 5 has best performance
%%%%%%%%%%%%%%%%%%%%%%%%
2. Try 12 forward taps
[dfseSNR,W,b]=dferake(1,p,12,1,11,1,.181*[1 zeros(1,23)])
dfseSNR = 11.1843
W = 0.0014 -0.0026 0.0041 -0.0061 0.0089 -0.0129 0.0186 -0.0268 0.0385 -0.0553 0.0794 0.3909
0.0002 -0.0010 0.0019 -0.0031 0.0046 -0.0068 0.0099 -0.0143 0.0205 -0.0295 0.0423 0.4106
b = 0.6969
This is very close to the best possible with MMSE-DFE. Nff=6, Nb=1, delay = 5 is almost as good
Note that best possible ZF-DFE performance is 10.8dB
%%%%%%%%%%%%%%%%%%%%%%%
3. For ZF-DFE filters, use very small noise. Only thing to note that the SNR value
output is meaningless. (why ?)
[dfseSNR,W,b]=dferake(1,p,12,1,11,1,.0001*[1 zeros(1,23)])
dfseSNR = 43.2669
W =**0.0044 -0.0054 0.0066 -0.0080 0.0097 -0.0116 0.0139 -0.0166 0.0198 -0.0236 0.0281 0.4713**
-0.0042 0.0048 -0.0054 0.0063 -0.0073 0.0086 -0.0101 0.0120 -0.0142 0.0168 -0.0200 0.4951
b = 0.8403