A Proposal for MLSE in COM Reference Receiver for KR/CR (AUI?)

Hossein Shakiba
Huawei Technologies Canada
January 04, 2024

Introduction

- Defining MLSE for COM reference receivers was highlighted as one of the priorities in phase 1 (lusted_3dj_elec_01_231207.pdf)

- MLSE seems necessary for KR/CR receivers

- MLSE may be necessary for AUIs, depending on the loss target (TBD)

- Some options are:

  1) Include MLSE COM calculations based on the existing proposal
  2) Use MLSE coding gain as a rough estimate (costs accuracy)
  3) Further simplify and relax COM margin by a constant amount (costs more accuracy)
  4) Find a better replacement for MLSE (currently no clear path)
  5) Ignore MLSE for channel compliance (channels need to become better)

History

- A detailed proposal for calculating the MLSE COM improvement using simple equations and compatible with the COM flow was proposed first in November 2022 in the IEEE Plenary (shakiba_3df_01a_2211.pdf)
  - Calculates real SNR improvement of MLSE over DFE on a channel-by-channel basis
  - COM flow (Annex 93A) compatible
  - COM function friendly and fast and only needs few lines of code to post-process COM-generated data
- The proposal was further detailed and presented in January 2023 802.3 Joint Task Force Session (shakiba_3dj_01_230116.pdf) and in February 2023 802.3 dj Electrical ad hoc Meeting (shakiba_3dj_elec_01_230223.pdf)
- This proposal was implemented in the COM Matlab function ver4.0 in February 2023 (mellitz_3dj_elec_01a_230223.pdf)

History

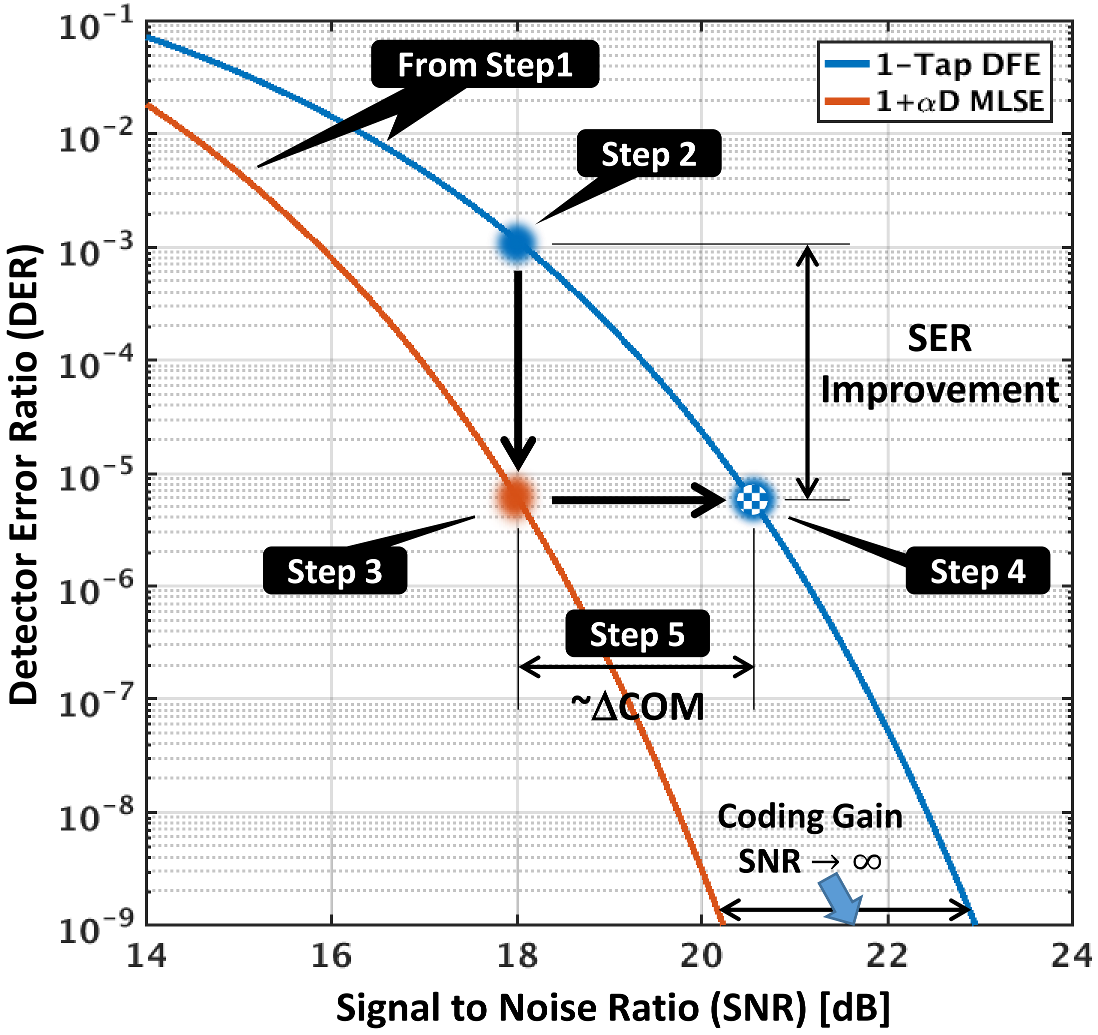
● Based on the feedback, two updates have been provided since then:

1) April 2023 802.3 dj Electrical ad hoc Meeting (shakiba_3dj_elec_01_230420.pdf):
   a) COM improvement calculation was updated based on the target detector error ratio (DER0)
   b) A more comprehensive and accurate method for calculating MLSE noise PDF was explained
   c) Effect of noise coloring was analyzed and added
2) April 2023 802.3 dj Electrical ad hoc Meeting (shakiba_3dj_elec_02_230420.pdf):
   a) Error propagation in MLSE was analyzed and shown to be always better than DFE (approaches DFE as α approaches 1)

Proposal Recap

- The proposal specified following steps:

1) Use COM analysis to find DFE tap, α
2) From COM data calculate SNR_{DFE}
3) Use analysis to calculate DER_{MLSE} at SNR_{DFE}
4) Use analysis to calculate $SNR_{DFE,\ equivalent}$ for the same DFE that yields the same DER_{MLSE}
5) Increase from SNR_{DFE} to $SNR_{DFE,\ equivalent}$ gives a good estimate of COM improvement of MLSE



What was Added in Matlab COM Function v4.0 Beta1L

- After COM calculation (no change) the following values are passed to the MLSE calculator:
  a) DFE tap value (α)
  b) Signal amplitude (A_s)
  c) Total noise PDF and CDF (PDF_{noise} and CDF_{noise})

  } Step 1

- MLSE calculator calculates:
  a) Signal: $main = (L-1)A_s$ (where $L = 4$ for PAM4)
  b) Noise: σ_{noise} from noise PDF
  c) $SNR_{DFE} = \frac{1}{3}\frac{L+1}{L-1}\frac{main^2}{\sigma_{noise}{}^2}$

  } Step 2

  d) $SER \approx 2\sum_{j=1}^{\infty} j\left(\frac{L-1}{L}\right)^j\left(1 - CDF_{noise}\left(\frac{main}{L-1}\sqrt{1+(j-1)(1-\alpha)^2+\alpha^2}\right)\right)$

  } Step 3

  e) $SNR_{DFE,equivalent} \approx \left(\frac{L-1}{main}CDF_{noise}^{-1}\left(1-\frac{1}{2}SER_{MLSE}\left(\frac{1}{L-1}+CDF_{noise}\left(\frac{main}{L-1}(1-2\alpha)\right)\right)\right)\right)^2 SNR_{DFE}$

  } Step 4

  f) $\Delta COM \approx 10\log_{10}\left(\frac{SNR_{DFE,equivalent}}{SNR_{DFE}}\right)$

  } Step 5

Updates since then:

1) A minor tweak to change SER to DER
2) A more comprehensive method to calculate MLSE sequence noise
3) A new way to calculate noise to include coloring

MLSE Function in the COM Matlab Code (v4.0 Beta1L):

```
1936 function [MLSE_results] = MLSE(param,alpha,A_s,A_ni,PDF,CDF)
1937 % OP.MLSE= 1 ... COM and VEC will be adjusted with MLSE CDF
1938 % OP.MLSE= 2 ... COM and VEC will be adjusted with MLSE Gaussian assumptions
1939 % Based on oif2022.580.00 / IEEE802. shakiba_3dj_01_230116 by Hossein Shakiba
1940 %% step 0
1941     COM_from_matlab=20*log10(A_s/A_ni);
1942     L=param.levels;
1943     DERO=param.specBER;
1944 %% step 1 from slide 6/5
1945     A_peak=(L-1)*A_s; % slide 6 A_s is main in appendix a
1946     main=A_peak;
1947     k_DER=qfuncinv(param.specBER);
1948     sigma_noise=sqrt(sum(PDF.y.*PDF.x.^2));
1949     SNR_dB=10*log10( 1/3*(L+1)/(L-1)*(A_peak^2)/sigma_noise^2) ;
1950     COM=SNR_dB-10*log10((L^2-1)/3*k_DER^2);
1951 %     sprintf('COM from Matlab %g dB\n COM from slide 6 using Gaussian asumptions %g dB\n', COM_from_matlab ,COM)
1952 if A_s >= A_ni
1953 %% step 2 slide 10/8
1954     SNR_DFE=1/3*(L+1)/(L-1)*(A_peak^2)/sigma_noise^2;
1955 %% step 2 slide 10/8
1956 %     DER_DFE=    2/ ( L/(L-1) -qfunc(   (1-2*alpha)*main/(L-1)/sigma_noise )   )*(qfunc(main/(L-1)/sigma_noise));
1957 %     DER_DFE_CDF=2/ ( L/(L-1)-CDF_ev(   (1-2*alpha)*main/(L-1),PDF,CDF )       )*CDF_ev((main/(L-1)),PDF,CDF);
1958 %% step 3 side 11/9
1959     j=1:200;
1960     DER_MLSE=2*sum( j .* ((L-1)/L).^j .* qfunc( sqrt(1+(j-1)*(1-alpha)^2+alpha^2).* main/((L-1)*sigma_noise )       ));
1961     DER_MLSE_CDF=0; jj=1;
1962     DER_delta = inf;
1963     while DER_delta > .001
1964         last_DER_MLSE_CDF=DER_MLSE_CDF;
1965         DER_MLSE_CDF=2*( jj .* ((L-1)/L).^jj .* CDF_ev( sqrt(1+(jj-1)*(1-alpha)^2+alpha^2).* main/((L-1) ),PDF,CDF       ))+DER_MLSE_CDF;
1966         DER_delta= 1-last_DER_MLSE_CDF/DER_MLSE_CDF;
1967         jj=jj+1;
1968     end
1969 %% step 4 slide 12/10
1970     SNR_DFE_eqivalent=SNR_DFE*(...
1971         (L-1)*sigma_noise/main *    qfuncinv(...
1972         1/2 *DER_MLSE*(L/(L-1) - qfunc((1-2*alpha)*main/(L-1)*sigma_noise )) ...
1973         )   ...
1974         )^2;
1975     SNR_DFE_eqivalent_CDF=SNR_DFE*(...
1976         (L-1)/main *    CDF_inv_ev(...
1977         1/2 *DER_MLSE_CDF*(L/(L-1) - CDF_ev((1-2*alpha)*main/(L-1),PDF,CDF )) ...
1978         ,PDF, CDF )    ...
1979         )^2;
1980
1981 %% step 5 slide 13/11
1982 delta_com=10*log10(SNR_DFE_eqivalent/SNR_DFE);
1983 delta_com_CDF=10*log10(SNR_DFE_eqivalent_CDF/SNR_DFE);
1984 new_com_CDF=COM_from_matlab+delta_com_CDF;
1985 else
1986     warning('MLSE not applied because there is more noise than signal')
1987     DER_MLSE=[];
1988     DER_MLSE_CDF=[];
1989     SNR_DFE_eqivalent=[];
1990     SNR_DFE_eqivalent_CDF=[];
1991     new_com_CDF=COM_from_matlab;
1992     delta_com_CDF=0;
1993     delta_com=0;
1994     SNR_DFE=[];
1995 end
```

<u>This fix in needed:</u>

must be placed in brackets (similar to line 1960)

Only affects calculations with Gaussian noise assumption

What is Really Being Calculated

- COM code directly followed the steps by which the MLSE COM analysis was conducted and presented
- The interim steps that were purposely included for clarity of the analysis description can be skipped, boiling down the MLSE COM calculation to:
  - ❖ Passing α, A_s, and CDF_{noise} to the MLSE calculator
  - ❖ Calculating:

$$DER_{MLSE} \approx 2\sum_{j=1}^{\infty} j\left(\frac{3}{4}\right)^{j}\left(1 - CDF_{noise}\left(A_s\sqrt{1 + (j-1)(1-\alpha)^2 + \alpha^2}\right)\right)$$

$$\boxed{\Delta COM \approx 20\log_{10}\left(\frac{1}{A_s} CDF_{noise}^{-1}\left(1 - \frac{1}{2} DER_{MLSE}\left(\frac{1}{3} + CDF_{noise}\left((1-2\alpha)A_s\right)\right)\right)\right)}$$

This is already implemented in the COM Matlab function. It is simple and costs almost no extra run time.

Presented Updates (1st, Must)

- Change the target error from SER (symbol error rate) to DER (detector error ratio)
- The MLSE COM calculation reduces to:

$$DER_{MLSE} \approx 2\sum_{j=1}^{\infty}\left(\frac{3}{4}\right)^{j}\left(1 - CDF_{noise}\left(A_s\sqrt{1+(j-1)(1-\alpha)^2+\alpha^2}\right)\right)$$

$$\boxed{\Delta COM \approx 20\log_{10}\left(\frac{1}{A_s}CDF_{noise}^{-1}\left(1-\frac{2}{3}DER_{MLSE}\right)\right)}$$

Requires a minor tweak and in fact simplifies the implemented calculation.

Presented Updates (2nd, Recommended)

- For some computation overhead, adopt the approach presented in shakiba_3dj_elec_01_230420.pdf to more accurately calculate PDF of the MLSE sequence noise

- The MLSE COM calculation becomes:

$$DER_{MLSE} \approx 2\sum_{j=1}^{\infty}\left(\frac{3}{4}\right)^{j}\left(1 - CDF_{noise,jEE}\left(A_s\left(1+(j-1)(1-\alpha)^2+\alpha^2\right)\right)\right)$$

where $CDF_{noise,jEE}$ is calculated from $PDF_{noise,jEE}$:

$$PDF_{noise,jEE}(x) = PDF_{noise}(x) * \mathrm{conv}_{i=2}^{j} PDF_{noise}\left((1-\alpha)x\right) * PDF_{noise}(\alpha x)$$

$$\boxed{\Delta COM \approx 20\log_{10}\left(\frac{1}{A_s}CDF_{noise}^{-1}\left(1 - \frac{2}{3}DER_{MLSE}\right)\right)}$$

Requires j convolutions for each term of the summation. The extra run time is reasonable and still negligible.

Presented Updates (3rd, Recommended)

- For more computation overhead, adopt the approach presented in shakiba_3dj_elec_01_230420.pdf to include the effect of noise coloring
- The MLSE COM calculation becomes:

$$DER_{MLSE} \approx 2\sum_{j=1}^{\infty}\left(\frac{3}{4}\right)^{j}\left(1 - CDF_{noise,jEE}\left(A_s \frac{\left(\mathrm{trace}\left(\rho_{noise,jEE}\right)\right)^{\frac{3}{2}}}{\sqrt{\sum_{vertical}\sum_{horizental}\left(\rho_{noise,jEE}\right)}}\right)\right)$$

where $CDF_{noise,jEE}$ is calculated from $PDF_{noise,jEE}$:

$$PDF_{noise,jEE}(x) = PDF_{noise}(x) * \mathrm{conv}_{i=2}^{j} PDF_{noise}\left((1-\alpha)x\right) * PDF_{noise}(\alpha x)$$

and $\rho_{noise,jEE}$ is calculated from the correlation coefficients of the noise:

$$\rho_{noise,jEE} = \begin{bmatrix} 1 & -(1-\alpha)\rho_1 & +(1-\alpha)\rho_2 & \cdots & (-1)^{j+1}\alpha\rho_j \\ -(1-\alpha)\rho_{-1} & (1-\alpha)^2 & -(1-\alpha)^2\rho_1 & \cdots & (-1)^{j}\alpha(1-\alpha)\rho_{j-1} \\ +(1-\alpha)\rho_{-2} & -(1-\alpha)^2\rho_{-1} & (1-\alpha)^2 & \cdots & (-1)^{j-1}\alpha(1-\alpha)\rho_{j-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-1)^{j+1}\alpha\rho_{-j} & (-1)^{j}\alpha(1-\alpha)\rho_{-(j-1)} & (-1)^{j-1}\alpha(1-\alpha)\rho_{-(j-2)} & \cdots & \alpha^2 \end{bmatrix}$$

Correlation coefficients in RED are due to noise coloring.

Presented Updates (3rd, Recommended)

and the correlation coefficients are obtained from inverse Fourier transform of the overall noise PSD (colored):

$$R_{NN}(\tau) = F^{-1}\{PSD_{noise}\}$$

which is obtained as power sum of the individual noise PSDs, each calculated based on their corresponding shaping filters (next slides)

$$PSD_{noise} = PSD_N + PSD_{TX} + PSD_{XT} + PSD_{ISI} + PSD_J$$

and finally:

$$\boxed{\Delta COM \approx 20\log_{10}\left(\frac{1}{A_s} CDF_{noise}^{-1}\left(1 - \frac{2}{3} DER_{MLSE}\right)\right)}$$

Requires j convolutions and calculation of the correlation matrix for each term of the summation. The extra run time is still reasonable and negligible.

- If this method is adopted, it is also recommended that COM Matlab function be modified to import any of the individual input noise PSDs that is already colored

Noise Shaping Filters – eta0

- Appendix 93A:

$$\sigma_N^2 = \eta_0 \int_0^{\infty} |H_r(f) H_{ctf}(f)|^2 df \qquad (93A\text{–}35)$$

$$H_r(f) = \frac{1}{1 - 3.414214(f/f_r)^2 + (f/f_r)^4 + j2.613126(f/f_r - (f/f_r)^3)} \qquad (93A\text{–}20)$$

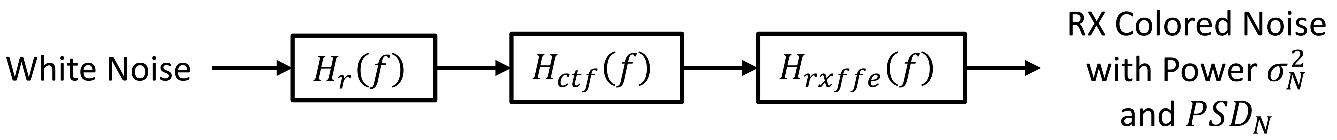
$$H_{ctf}(f) = \frac{\left(10^{\frac{g_{DC}}{20}} + j\frac{f}{f_z}\right)\left(10^{\frac{g_{DC2}}{20}} + j\frac{f}{f_{LF}}\right)}{\left(1 + j\frac{f}{f_{p1}}\right)\left(1 + j\frac{f}{f_{p2}}\right)\left(1 + j\frac{f}{f_{LF}}\right)} \qquad (93A\text{–}22)$$

- Now RX FFE should be added, similar to (93A-21) for TX:

$$H_{ffe}(f) = \sum_{i=-1}^{1} c(i)\exp(-j2\pi(i+1)(f/f_b)) \qquad (93A\text{–}21)$$

Need a similar expression for RX FFE

- eta0 noise coloring filter:



Noise Shaping Filters – TX SNR

- Appendix 93A:

$$\sigma_{TX}^2 = [h^{(0)}(t_s)]^2 10^{-SNR_{TX}/10} \quad \text{(93A–30)}$$

$$h^{(k)}(t) = \int_{-\infty}^{\infty} X(f)H^{(k)}(f)\exp(j2\pi ft)df \quad \text{(93A–24)}$$

$$X(f) = A_t T_b \mathrm{sinc}(fT_b) \quad \text{(93A–23)}$$

$$H^{(k)}(f) = H_{ffe}(f)H_t(f)H_{21}^{(k)}(f)H_r(f)H_{ctf}(f) \quad \text{(93A–19)}$$

$$H_{ffe}(f) = \sum_{i=-1}^{1} c(i)\exp(-j2\pi(i+1)(f/f_b)) \quad \text{(93A–21)}$$

$$H_t(f) = \exp(-2(\pi f T_r/1.6832)^2) \quad \text{(93A–46)}$$

$$H_r(f) = \frac{1}{1-3.414214(f/f_r)^2+(f/f_r)^4+j2.613126(f/f_r-(f/f_r)^3)} \quad \text{(93A–20)}$$

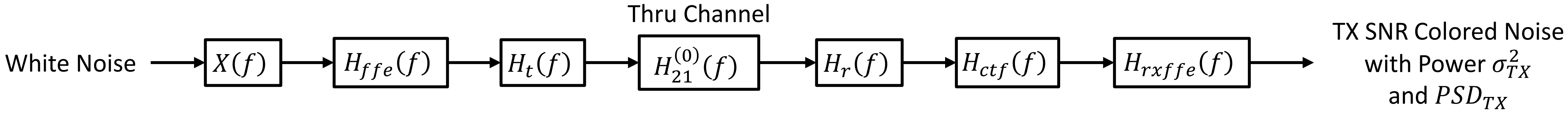
$$H_{ctf}(f) = \frac{\left(10^{\frac{g_{DC}}{20}}+j\frac{f}{f_z}\right)\left(10^{\frac{g_{DC2}}{20}}+j\frac{f}{f_{LF}}\right)}{\left(1+j\frac{f}{f_{p1}}\right)\left(1+j\frac{f}{f_{p2}}\right)\left(1+j\frac{f}{f_{LF}}\right)} \quad \text{(93A–22)}$$

- Now RX FFE should be added, similar to (93A-21) for TX:

$$H_{ffe}(f) = \sum_{i=-1}^{1} c(i)\exp(-j2\pi(i+1)(f/f_b)) \quad \text{(93A–21)}$$

Need a similar expression for RX FFE

- TX noise coloring filter:



Noise Shaping Filters – Xtalk

- Appendix 93A:

$$\sigma_{XT}^2 = \sum_{k=1}^{K-1} [\sigma_i^{(k)}]^2 \qquad (93A\text{–}34)$$

$$[\sigma_m^{(k)}]^2 = \sigma_X^2 \sum_n [h^{(k)}((m/M+n)T_b)]^2 \qquad (93A\text{–}33)$$

$$\sigma_X^2 = \frac{L^2-1}{3(L-1)^2} \qquad (93A\text{–}29)$$

$$h^{(k)}(t) = \int_{-\infty}^{\infty} X(f)H^{(k)}(f)\exp(j2\pi ft)df \qquad (93A\text{–}24)$$

$$X(f) = A_t T_b \operatorname{sinc}(fT_b) \qquad (93A\text{–}23)$$

$$H^{(k)}(f) = H_{ffe}(f)H_t(f)H_{21}^{(k)}(f)H_r(f)H_{ctf}(f) \qquad (93A\text{–}19)$$

$$H_{ffe}(f) = \sum_{i=-1}^{1} c(i)\exp(-j2\pi(i+1)(f/f_b)) \qquad (93A\text{–}21)$$

$$H_t(f) = \exp(-2(\pi f T_r/1.6832)^2) \qquad (93A\text{–}46)$$

$$H_r(f) = \frac{1}{1-3.414214(f/f_r)^2+(f/f_r)^4+j2.613126(f/f_r-(f/f_r)^3)} \qquad (93A\text{–}20)$$

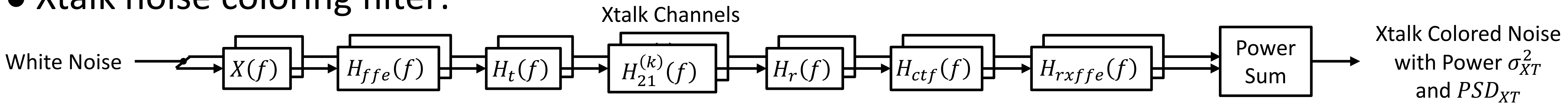
$$H_{ctf}(f) = \frac{\left(10^{\frac{g_{DC}}{20}}+j\frac{f}{f_z}\right)\left(10^{\frac{g_{DC2}}{20}}+j\frac{f}{f_{LF}}\right)}{\left(1+j\frac{f}{f_{p1}}\right)\left(1+j\frac{f}{f_{p2}}\right)\left(1+j\frac{f}{f_{LF}}\right)} \qquad (93A\text{–}22)$$

- Now RX FFE should be added, similar to (93A-21) for TX:

$$H_{ffe}(f) = \sum_{i=-1}^{1} c(i)\exp(-j2\pi(i+1)(f/f_b)) \qquad (93A\text{–}21)$$

Need a similar expression for RX FFE

- Xtalk noise coloring filter:



Noise Shaping Filters – ISI and Jitter

- Appendix 93A:

$$\sigma_{ISI}^2 = \sigma_X^2 \sum_n h_{ISI}^2(n) \quad (93A\text{–}31)$$

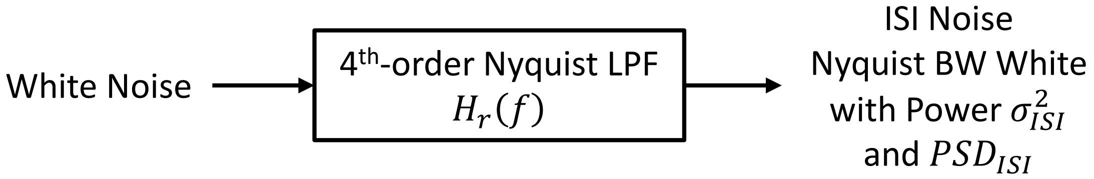
$$\sigma_J^2 = (A_{DD}^2 + \sigma_{RJ}^2)\sigma_X^2 \sum_n h_J^2(n) \quad (93A\text{–}32)$$

- ISI and jitter noises are assumed to remain white, but within the Nyquist bandwidth

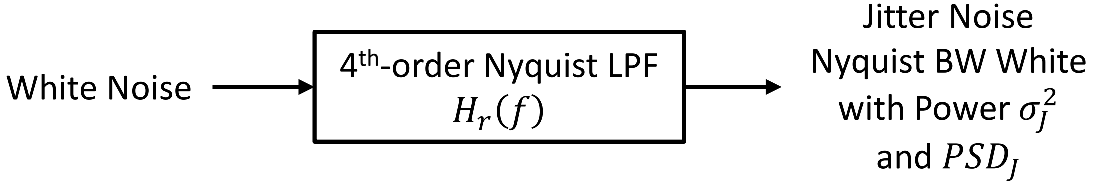
$$H_r(f) = \frac{1}{1 - 3.414214(f/f_r)^2 + (f/f_r)^4 + j2.613126(f/f_r - (f/f_r)^3)} \quad (93A\text{–}20)$$

Arbitrary and reasonable choice

- ISI noise filter:



- Jitter noise filter:



Parameters Needed to Be Passed to the MLSE Calculator

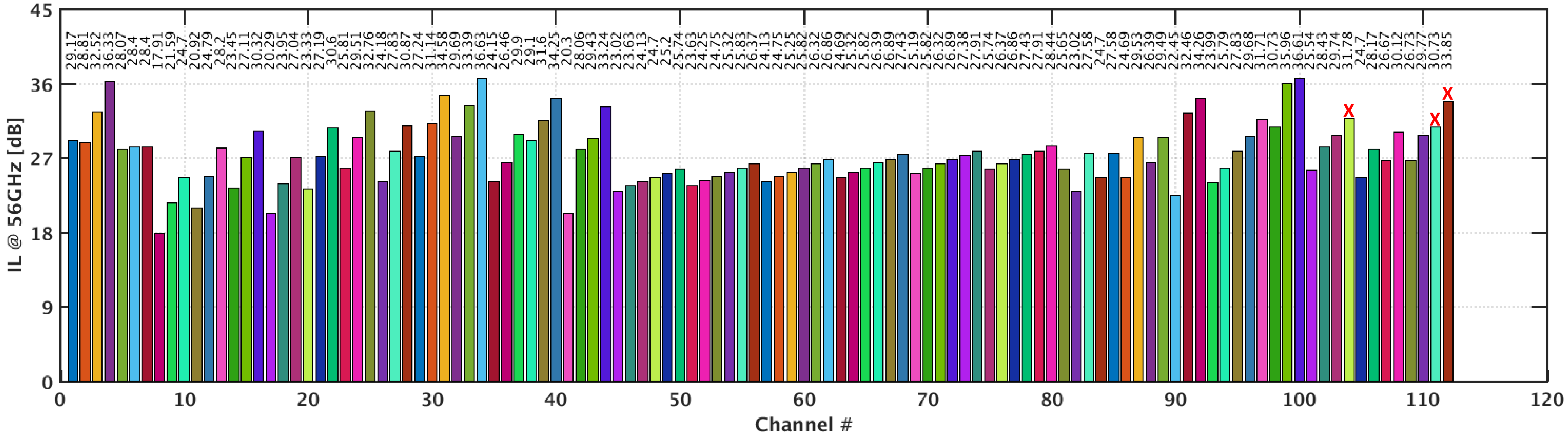
Parameter Description	Annex 93A Parameter Name	COM Matlab Parameter Name
Without including noise coloring effect		
DFE tap	b	DFE_taps
Signal amplitude	A_s	available_signal_after_eq_mV
noise PDF and CDF		PDF, CDF
Additional parameters for including noise coloring effect		
Noise sigma values	$\sigma_N, \sigma_{XT}, \sigma_{TX}, \sigma_{ISI}, \sigma_J$	sgm_N, sgm_TX, sgm_xt, sgm_isi, sgm_rjit, sgm_p_dd
Baud rate	f_b	baud_rate_GHz
Noise filter 3dB cutoff frequency	f_r	f_r
Frequency vector and step	f	faxis, max_freq_step
CTLE DC gains	g_{DC}, g_{DC2}	g_DC_HP, CTLE_DC_gaon_dB
CTLE poles and zeros	$f_{p1}, f_{p2}, f_z, f_{LF}$	CTLE_zero_poles, HP_poles_zero
RX FFE taps and number of pre-taps		RxFFE, ffe_pre_tap_len
TX FFE taps	c	TXLE_taps
TX transition times		transmitter_transition_time
Channel transfer functions	$H_{21}^{(k)}$	sdd21 and sdd21_raw
Termination resistors	R_d	R_diepad
Number of FXET and NEXT channels		num_fext, num_next
Channel s4p file names		filename
Oversampling ratio	M	samples_per_ui

Alternatively, and maybe more efficiently, calculation of noise PSDs can be done within the COM Matlab function.
In fact, the filter transfer functions may have already been calculated.

Test Channels

Channel #	Channel Source
1	https://www.ieee802.org/3/dj/public/tools/CR/lim_3dj_03_230629.zip
2	https://www.ieee802.org/3/dj/public/tools/CR/lim_3dj_04_230629.zip
3 – 7	https://www.ieee802.org/3/dj/public/tools/CR/kocsis_3dj_02_2305.zip
8 – 34	https://www.ieee802.org/3/dj/public/tools/KR/mellitz_3dj_02_elec_230504.zip
35 – 40	https://www.ieee802.org/3/dj/public/tools/CR/shanbhag_3dj_01_2305.zip
40 – 44	https://www.ieee802.org/3/dj/public/tools/KR/shanbhag_3dj_02_2305.zip
45 – 80	https://www.ieee802.org/3/dj/public/tools/KR/weaver_3dj_02_2305.zip
80 – 88	https://www.ieee802.org/3/dj/public/tools/KR/weaver_3dj_elec_01_230622.zip
89	https://www.ieee802.org/3/dj/public/tools/CR/lim_3dj_07_2309.zip
90 – 96	https://www.ieee802.org/3/dj/public/tools/KR/akinwale_3dj_01_2310.zip
97 – 100	https://www.ieee802.org/3/dj/public/tools/CR/akinwale_3dj_02_2311.zip
101 – 112	https://www.ieee802.org/3/dj/public/tools/CR/weaver_3dj_02_2311.zip

Test Channels



- IL ranges from 17.9dB to 36.6dB (average 27.5dB)
- For channels 104, 111, and 112 the COM function did not converge
  - ❖ Failed to find the floating tap locations (best_floating_tap_locations)

COM Configuration

Table 93A-1 parameters			
Parameter	Setting	Units	Information
f_b	112	GBd	
f_min	0.05	GHz	
Delta_f	0.01	GHz	
C_d	[0.4e-4 0.9e-4 1.1e-4 ; 0.4e-4 0.9e-4 1.1e-4]	nF	[TX RX]
L_s	[0.13 0.15 0.14 ; 0.13 0.15 0.14]	nH	[TX RX]
C_b	[0.3e-4 0.3e-4]	nF	[TX RX]
z_p select	[2]		[test cases to run]
z_p (TX)	[12 33; 1.8 1.8]	mm	[test cases]
z_p (NEXT)	[12 31; 1.8 1.8]	mm	[test cases]
z_p (FEXT)	[12 33; 1.8 1.8]	mm	[test cases]
z_p (RX)	[12 31; 1.8 1.8]	mm	[test cases]
C_p	[0.4e-4 0.4e-4]	nF	[TX RX]
R_0	50	Ohm	
R_d	[46.25 46.25]	Ohm	[TX RX]
A_v	0.413	V	
A_fe	0.413	V	
A_ne	0.608	V	
AC_CM_RMS	0		[test cases]
L	4		
M	32		
filter and Eq			
f_r	0.5	*fb	
c(0)	0.54		min
c(-1)	[-0.4:0.02:0]		[min:step:max]
c(-2)	[0:0.02:0.16]		[min:step:max]
c(-3)	[-0.1:0.02:0]		[min:step:max]
C(-4)	[0:0.02:0.1]		[min:step:max]
C(-5)	0		[min:step:max]
C(-6)	0		[min:step:max]
c(1)	[-0.2:0.02:0]		[min:step:max]
N_b	1	UI	
b_max(1)	0.85		As/dffe1
b_max(2..N_b)	[0.3 0.2*ones(1,22)]		As/dfe2..N_b
b_min(1)	0.3		As/dffe1
b_min(2..N_b)	[-0.3 -0.2*ones(1,22)]		As/dfe2..N_b
g_DC	[-20:1:0]	dB	[min:step:max]
f_z	44.8	GHz	
f_p1	44.8	GHz	
f_p2	112	GHz	
g_DC_HP	[-6:1:0]		[min:step:max]
f_HP_PZ	1.4	GHz	
MLSE	1		
ffe_pre_tap_len	6		
ffe_post_tap_len	24		
ffe_tap_step_size	0		
ffe_main_cursor_min	0.7		
ffe_pre_tap1_max	0.7		
ffe_post_tap1_max	0.7		
ffe_tapn_max	0.7		
ffe_backoff	0		
ffe_float	0		1:FFE float
ffe_bg	4		float FFE groups
ffe_bf	5		taps per group
ffe_Nf	60		float taps range
LF_PreTap_Ext	10		LF FFE pre-tap ext
LF_PostTap_Ext	10		LF FFE post-tap ext

I/O control		
DIAGNOSTICS	0	logical
DISPLAY_WINDOW	0	logical
CSV_REPORT	0	logical
RESULT_DIR	.\results\100GEL_KR_{date}\	
SAVE_FIGURES	0	logical
Port Order	[1 3 2 4]	
RUNTAG	KR_eval_	
COM_CONTRIBUTION	0	logical
Operational		
COM Pass Threshold	3	dB
ERL Pass Threshold	8	dB
DER_0	1.00E-04	
T_r	4.00E-03	ns
FORCE_TR	1	logical
Local Search	2	
BREAD_CRUMBS	1	logical
SAVE_CONFIG2MAT	1	logical
PLOT_CM	0	
TDR and ERL options		
TDR	1	logical
ERL	1	logical
ERL_ONLY	0	logical
TR_TDR	0.01	ns
N	3500	
beta_x	0	
rho_x	0.618	
fixture delay time	[0 0]	[port1 port2]
TDR_W_TXPKG	0	
N_bx	21	UI
Tukey_Window	1	logical
Noise, Jitter		
sigma_RJ	0.01	UI
A_DD	0.02	UI
eta_0	5.00E-09	V^2/GHz
SNR_TX	33	dB
R_LM	0.95	

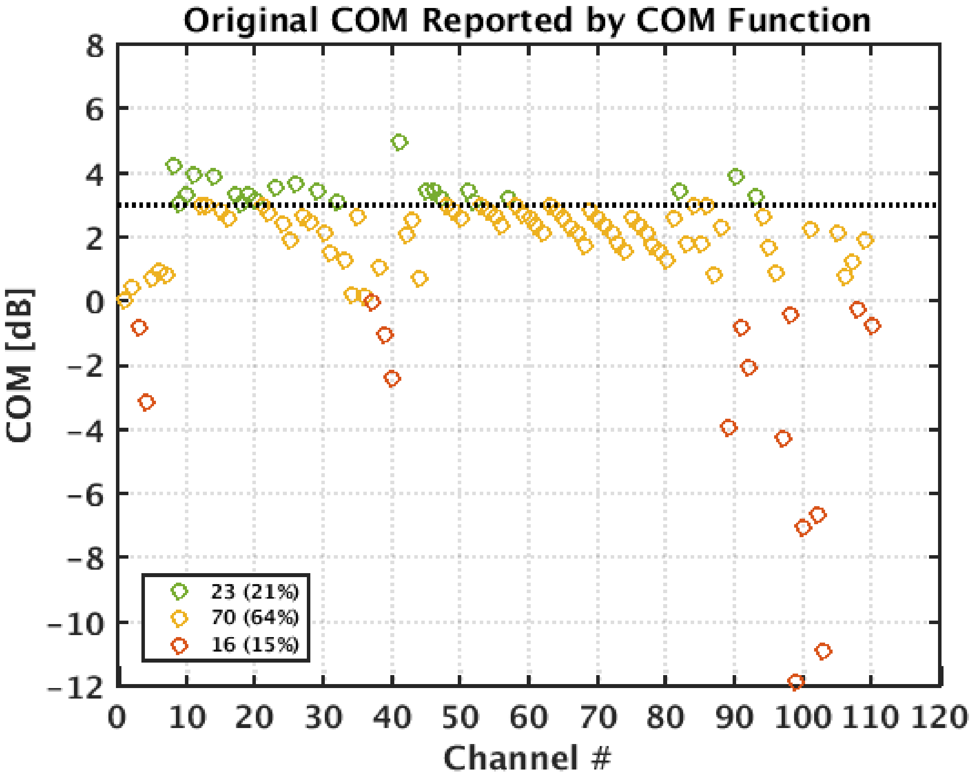
Table 93A-3 parameters		
Parameter	Setting	Units
package_tl_gamma0_a1_a2	[0.5E-3 0.89E-3 0.2E-3]	
package_tl_tau	6.141E-03	ns/mm
package_Z_c	[87.5 87.5 ; 92.5 92.5]	Ohm

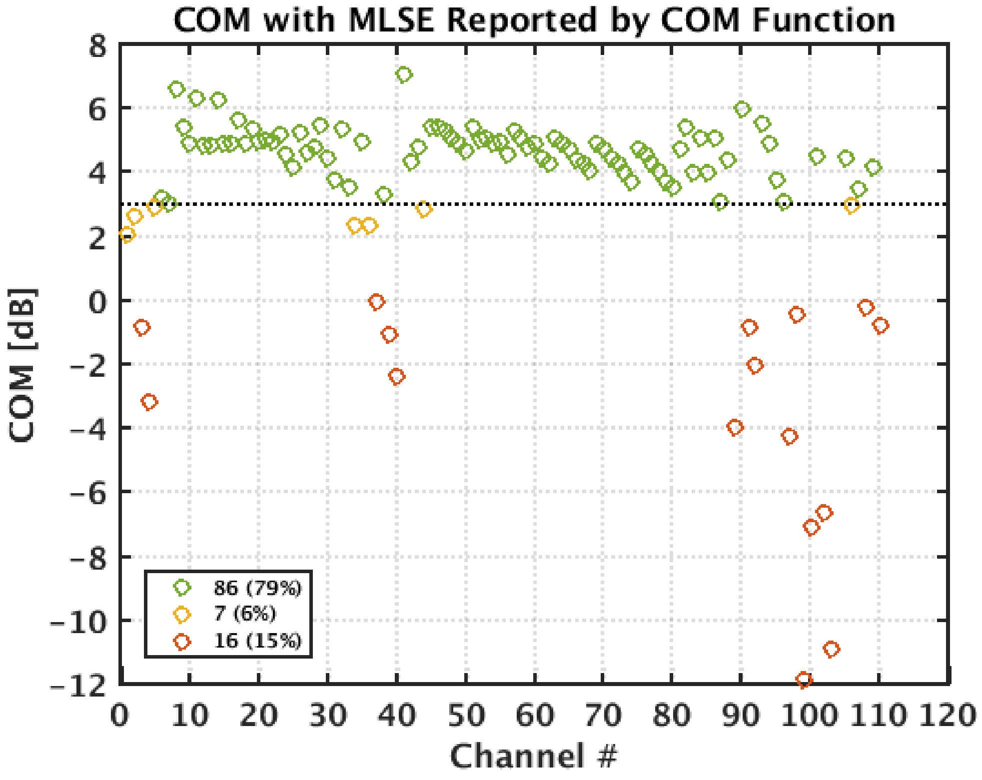
Table 92-12 parameters		
Parameter	Setting	
board_tl_gamma0_a1_a2	[0 3.8206e-04 9.5909e-05]	
board_tl_tau	5.79E-03	ns/mm
board_Z_c	100	Ohm
z_bp (TX)	110.3	mm
z_bp (NEXT)	110.3	mm
z_bp (FEXT)	110.3	mm
z_bp (RX)	110.3	mm
C_0	[0.29E-4]	nF
C_1	[0.19E-4]	nF
Include PCB	0	logical
Floating Tap Control		
N_bg	4	0 1 2 or 3 groups
N_bf	5	taps per group
N_f	60	UI span for floating taps
bmaxg	0.05	max DFE value for floating taps
B_float_RSS_MAX	0.02	rss tail tap limit
N_tail_start	25	(UI) start of tail taps limit
ICN parameters (v2.73+)		
f_v	0.528	*Fb
f_f	0.528	*Fb
f_n	0.528	*Fb
f_2	80	GHz
A_ft	0.6	V
A_nt	0.6	V

Receiver testing		
RX_CALIBRATION	0.000	logical
Sigma BBN step	5.00E-03	V

- COM Version: com_ieee8023_93a_420beta3
  - ❖ COM function calculates MLSE ΔCOM (based on SER)
- \+ Post-COM MLSE Calculator
  - ❖ Calculates MLSE ΔCOM based on SER, DER, with the improved MLSE noise PDF for white and colored noise

Test Results – COM Function



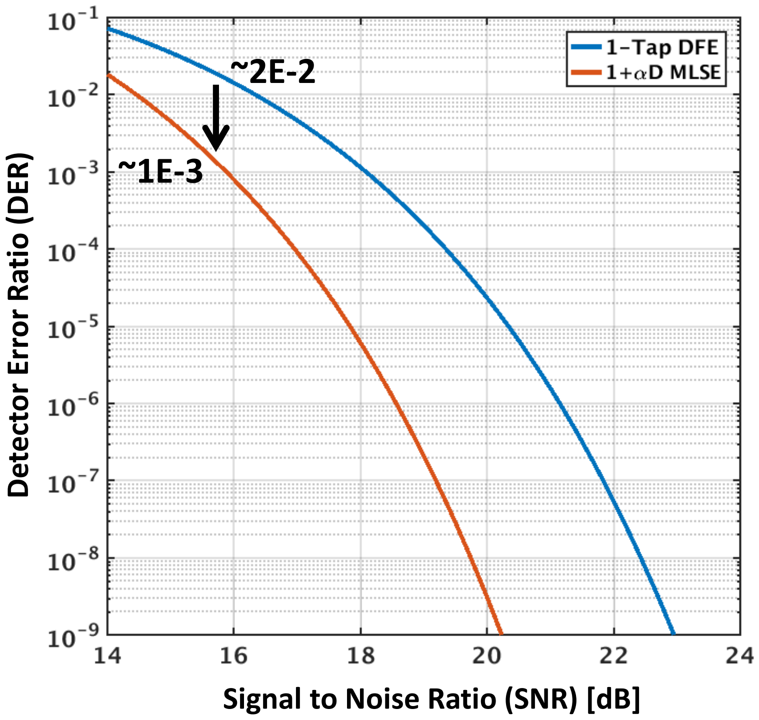


- Currently COM function ignores MLSE and reports ΔCOM = 0 if the original COM is negative
- This was a quick patch to ignore cases with low SNR, where analysis is less accurate
- 16 ignored cases for the current test channels

Cases to Ignore

- A better option is to, instead of ignoring based on COM, ignore cases with DER threshold value more than a reasonable level, say 2E-2
- DER is a more direct indication of the error performance
- DER threshold is DER at COM = 0 and is already reported by the COM function
- In either case, these are cases that most likely fail even with MLSE
- Nevertheless, it is still good to see the failing margin

Channels Ignored	Channel #
COM < 0	3, 4, 37, 39, 40, 89, 91, 92, 97, 98, 99, 100, 102, 103, 108, 110 (16 Channels)
DER Threshold > 2E-2	4, 40, 89, 97, 99, 100, 102, 103 (8 Channels)

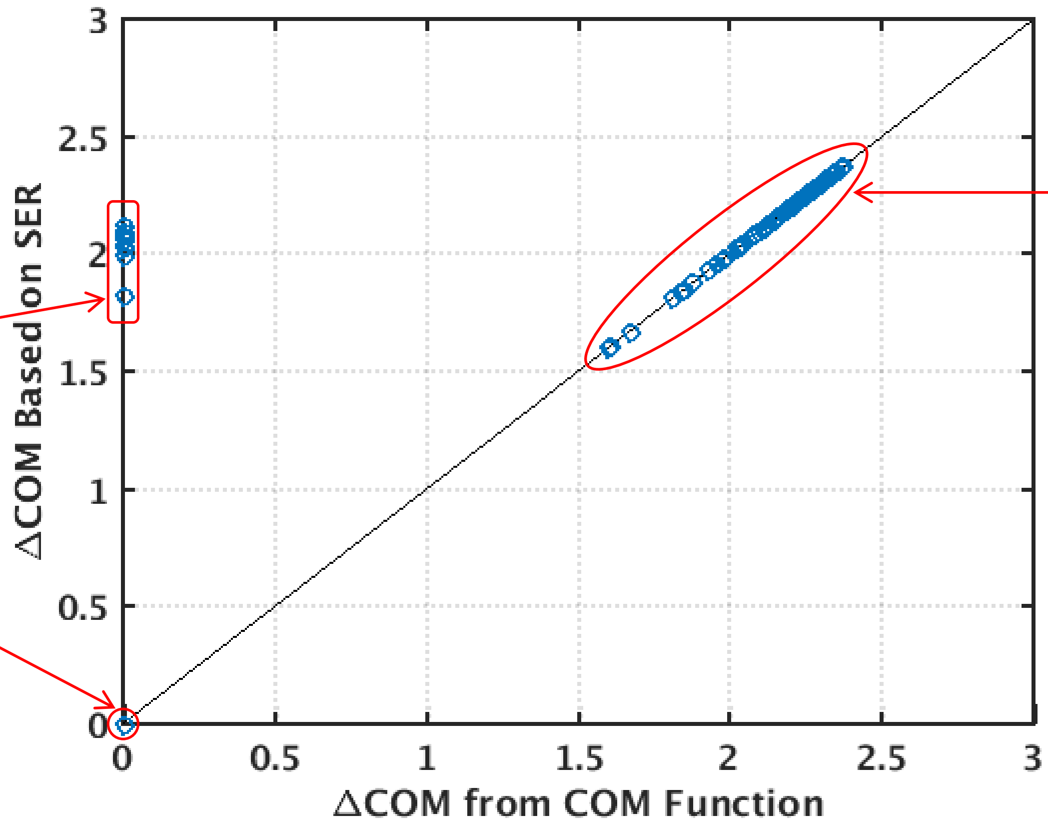


Test Results – Compare to COM Function

16 cases have COM < 0

8 of the 16 cases pass DER threshold of 2E-2 and can be added back to MLSE calculations

8 of the 16 cases have DER threshold above 2E-2



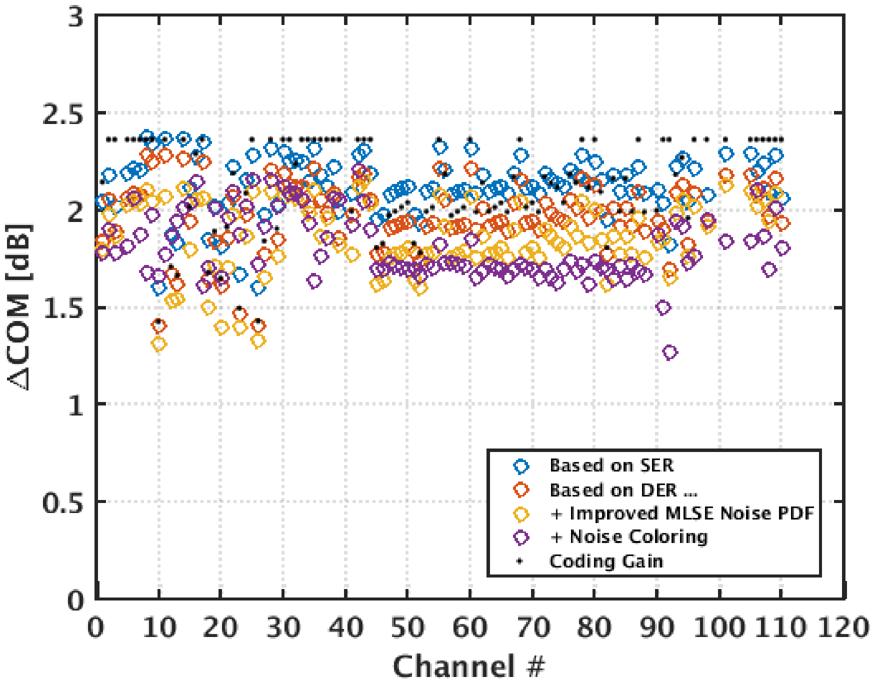
100% correlation for 93 other cases

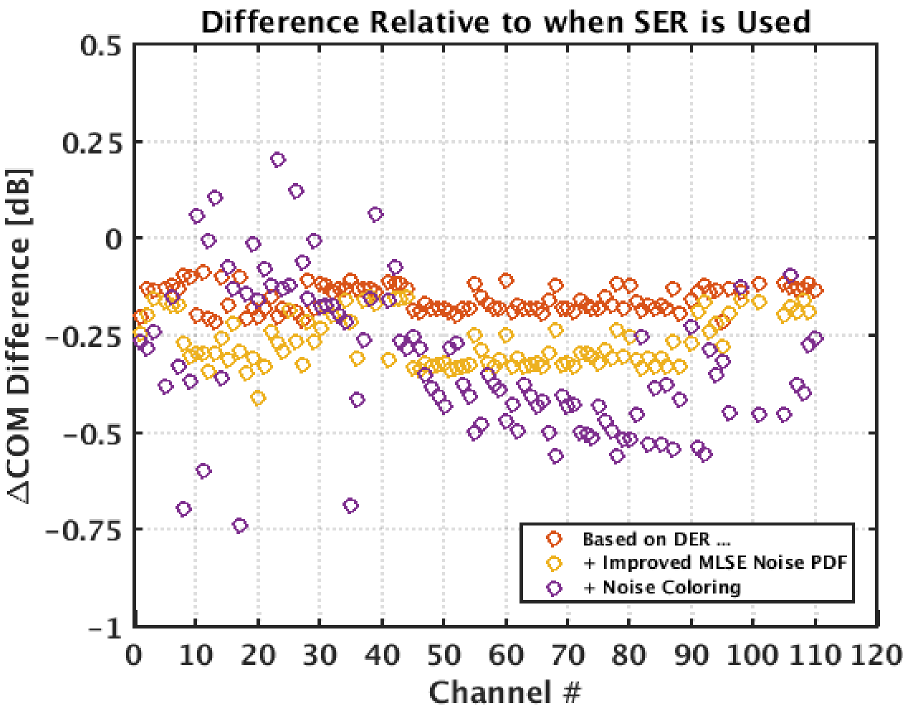
Total # of Channels	# of Channels for which COM Function Calculated COM	# of Channels for which COM Function Calculated ΔCOM	# of Channels for which COM Calculator Calculated ΔCOM
112	109	93	101

Test Results – Latest Updates

- Recent updates of the MLSE proposal on average predict a little less ΔCOM

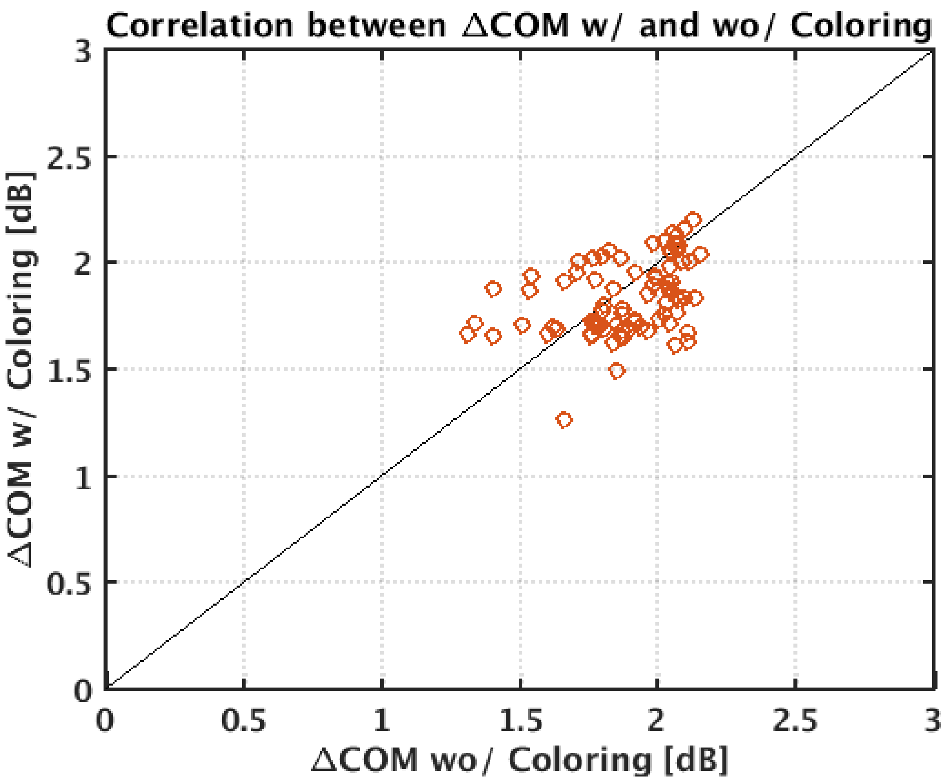
1) Change of SER to DER
2) + more comprehensive calculation method for MLSE noise PDF
3) + inclusion of noise coloring





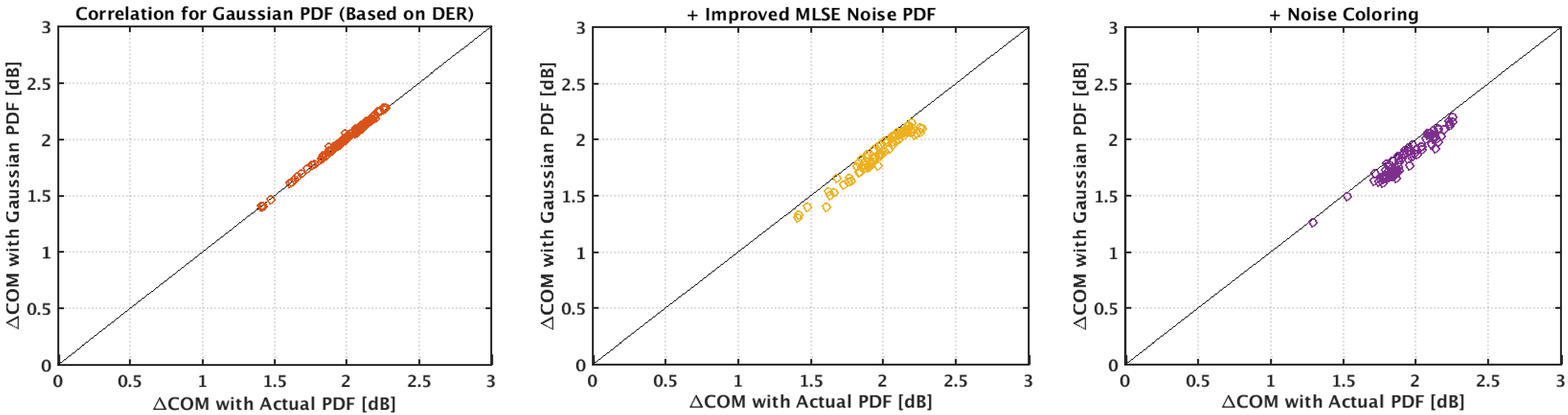
[dB]	Based on SER (as Currently Reported by the COM Function)	Based on DER (“Must” Change)	Improved MLSE Noise PDF, White (“Nice to ” Update)	Improved MLSE Noise PDF, Color (“Nice to ” Update)
ΔCOM (ave)	2.13	1.98	1.86	1.81
Difference (ave)	0	-0.15	-0.27	-0.32

Test Results – White or Colored Noise?



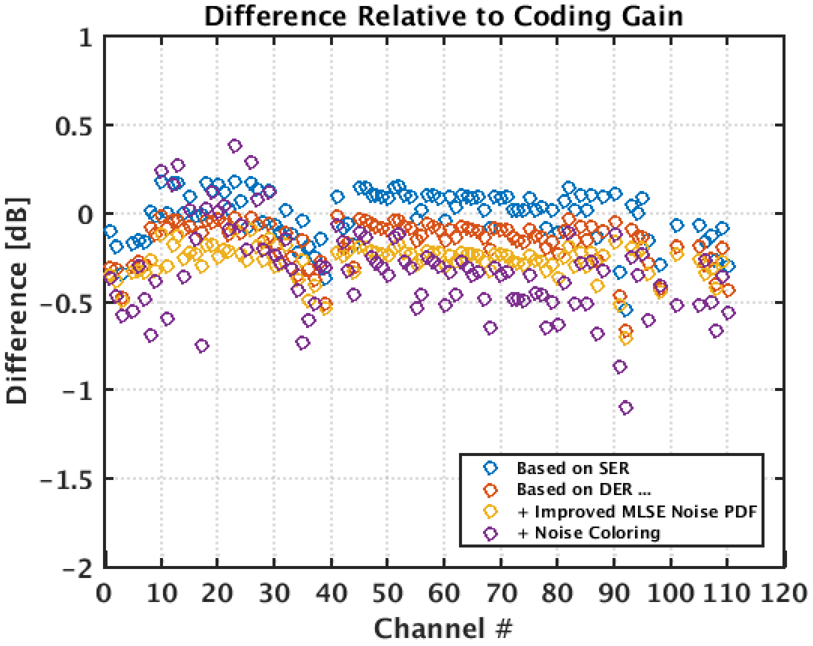
- The correlation between ΔCOM calculated with and without noise coloring is weak
- It is a good idea to include the effect of noise coloring

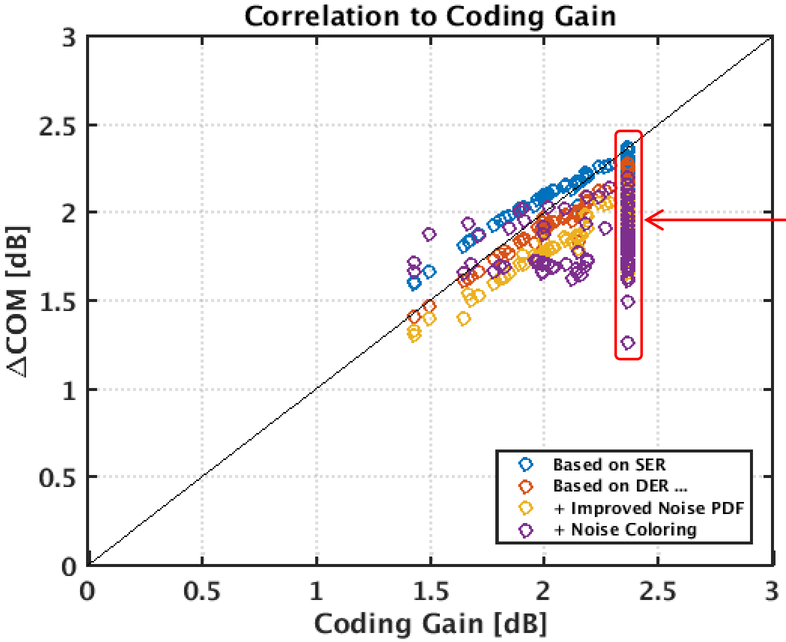
Test Results – Actual or Gaussian Noise Distribution?



- Using the actual noise PDF (as calculated by the COM function) instead of assuming it is Gaussian makes more sense as the accuracy of the MLSE calculator improves

Test Results – ΔCOM or Coding Gain?



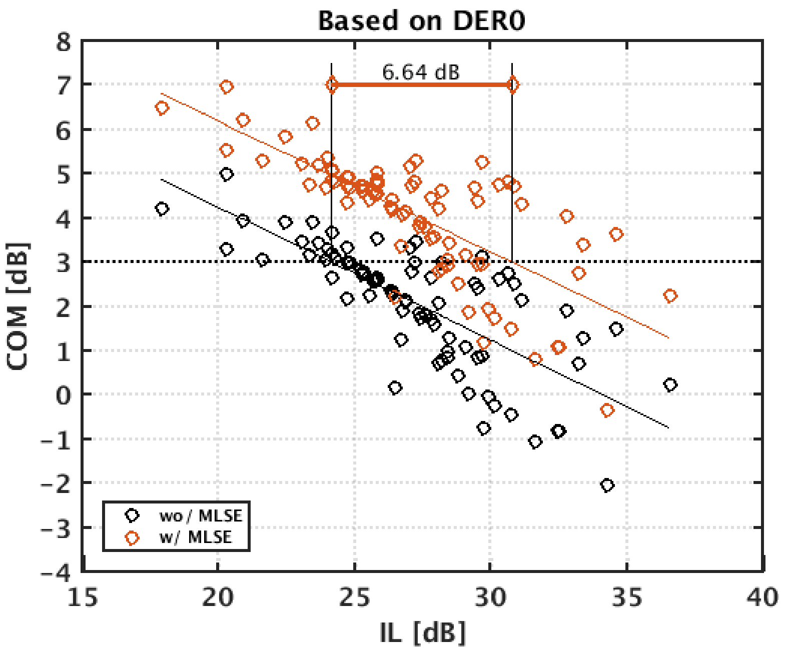


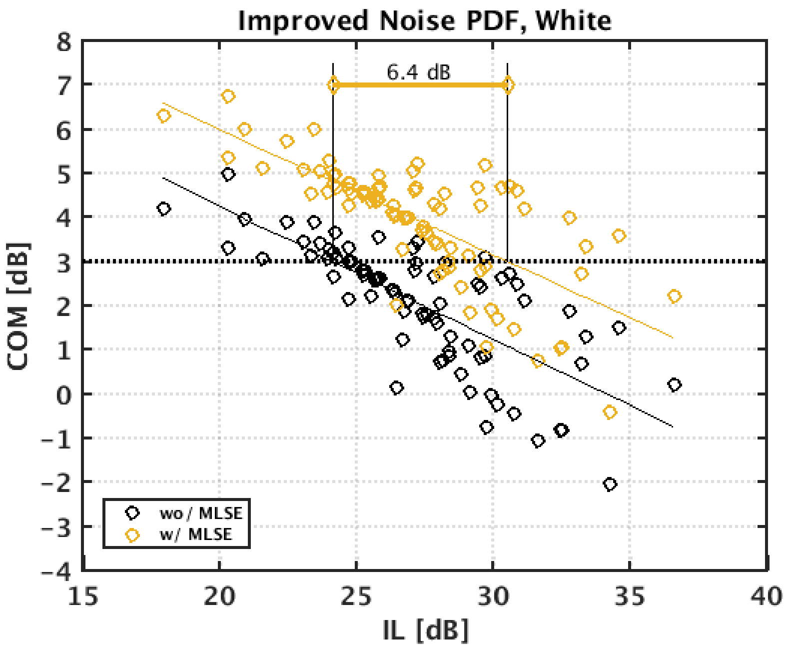
DFE tap saturates (0.85)

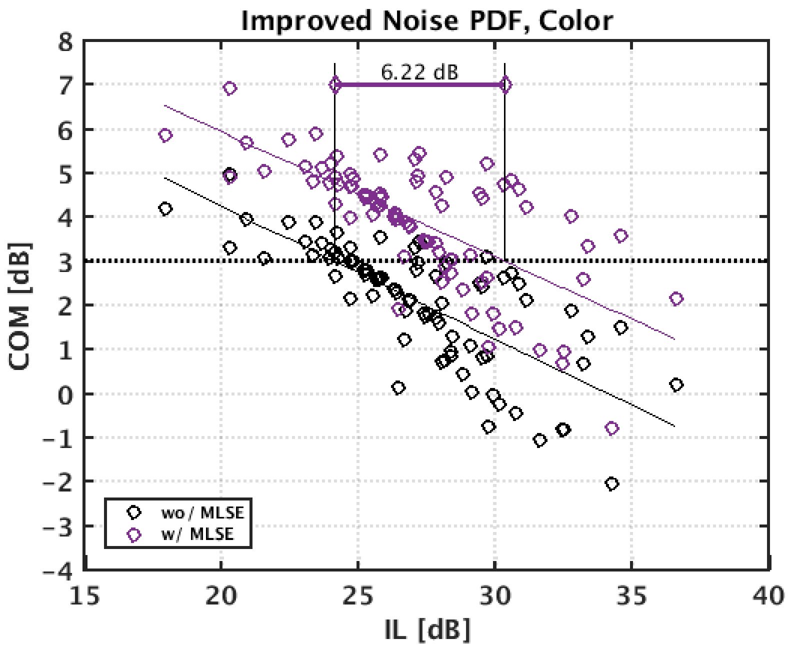
- Coding gain correlation to ΔCOM weakens as ΔCOM calculation becomes more accurate
- Coding gain is not a representative of MLSE COM advantage when DFE tap saturates

[dB]	Coding Gain ($10\log_{10}(1+\alpha^2)$)	Based on SER (as Currently Reported by the COM Function)	Based on DER (“Must” Change)	Improved MLSE Noise PDF, White (“Nice to ” Update)	Improved MLSE Noise PDF, Color (“Nice to ” Update)
ΔCOM (ave)	2.14	2.13	1.98	1.86	1.81
Difference (ave)	0	-0.01	-0.16	-0.27	-0.33

Test Results – IL Improvement







[dB]	Based on DER (“Must” Change)	Improved MLSE Noise PDF, White (“Nice to ” Update)	Improved MLSE Noise PDF, Color (“Nice to ” Update)
IL Improvement with MLSE	6.64	6.40	6.22

- Note that this is only a rough average estimate based a linear fit to the scattered data

Summary and Conclusions

- MLSE seems necessary for KR/CR and maybe for AUIs
- This presentation summarized a proposal for inclusion of MLSE in COM flow (Annex 93A and COM Matlab function)
- Other options such as ignoring MLSE, allowing a fix COM margin, or using MLSE coding gain are far less accurate and non-realistic
- Currently, this proposal is the most accurate and realistic option to include MLSE for channel compliance
- A simple calculator can post-process data generated by the COM Matlab function
- Effects of actual noise distribution and coloring were quantified and included in the proposal
- MLSE ΔCOM results for additional 112 test channel cases were presented