

COM Commit Requests 4p7_1, 4p7_2, and 4p7_3

Rich Mellitz, Samtec

Adam Gregory, Samtec

COM Ad Hoc December 10, 2024

lusted_3dj_04_2411

Background

- ❑ One commit request was deferred in the lusted_3dj_04_2411 presentation at the November IEEE802.3dj Task Force meeting
 - See table below

Commit Request #	Submitter	Description	Proposed Disposition
4p6_4	Hossein Shakiba	<ul style="list-style-type: none">• MLSE Sequence Truncation Feature	Revised
4p6_5	Hossein Shakiba	<ul style="list-style-type: none">• Quantization Noise Feature	Defer
4p6_6	Rich Mellitz	<ul style="list-style-type: none">• Align to D1.2 Annex 178A MLSD	Accept

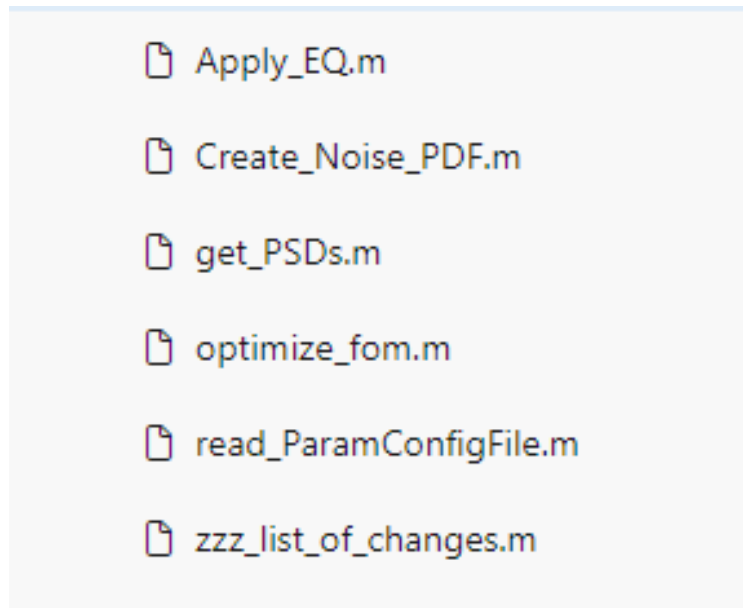
- ❑ Commit request 4p6_5 for including quantization noise in the COM computation is considered exploratory
 - It could used as data to support a respective comment against a draft
- ❑ ICN computation in Matlab script not aligned with clause 92

Commit Request 4p7_1

Align to Deferred Commit Request 4p6_5

- ❑ Most of the computation embodied in a function call
 - Makes easier to update code syntax if accepted and implemented in a draft
 - New function; **adjust_Rx_noise_for_quantization**
 - Similar to slide 7 in shakiba_3dj_COM_01_241029

❑ Functions modified.



Commit Request 4p7_1: New Keywords

REF: SHAKIBA_3DJ_COM_01_241029

❑ ENOB

- Default is 0
- If 0 or missing quantization is not used for MMSE
- If not 0, then it may be interpreted as an effective number of bits
 - Does not need to be an integer

❑ adc_clip_rate

- Default is $2 \cdot \text{DER0}$ if not specified
- Only used if ENOB is not 0

Commit Request 4p7_2

Align ICN with Clause 92

- ❑ Equation 92-44 and 92-45 has respective terms (A_{nt}^2/f_b) and (A_{ft}^2/f_b)
- ❑ The COM code divides by f_2 instead of f_b
- ❑ This request corrects this
- ❑ ICN is not a part of the COM computation
- ❑ Consider
 - ICN has been reported in prior presentations
 - ICN may be utilized for .3dj mated test fixture specification
- ❑ This is a simple change does not affect the COM computation

Commit Request 4p7_3 small update to COM 4.8beta_1

- ❑ fixed typo's and updated from shakiba_3dj_COM_02_241001 to shakiba_3dj_COM_01_241029

Beta test code

COM 4.8beta1

- Commit Request 4p7_1
- Commit Request 4p7_2

COM 4.8beta2

- Commit Request 4p7_3

Added Function: adjust_Rx_noise_for_quantization

```
adjust_Rx_noise_for_quantization.m +32
/src/com_fcts/adjust_Rx_noise_for_quantization.m

1 + function [chdata,NS,combined_interference_and_noise_pdf] = adjust_Rx_noise_for_quantization(combined_interference_and_noise_pdf,NS, chdata,fom_result,param,OP)
2 + % expected input
3 + % chdata(1).eq_pulse_response --- includes packages, Hf, and Hr, Ht, and Hffe(from tx)
4 + % align the first sample point to to ts in chdata(1).eq_pulse_response
5 + % align to terminology used in MMSE and optimize_FOM
6 +
7 + %
8 + sig_after_ctle_pdf = get_pdf_from_sampled_signal(chdata(1).pulse_sampled_w_tx_ffe_ctle, param.levels, param.delta_y);
9 + sig_after_ctle_cdf = cumsum(sig_after_ctle_pdf.y);
10 + adc_clip = -CDF_inv_ev(2*param.specBER, sig_after_ctle_pdf, sig_after_ctle_cdf);
11 + ctle_pulse_sigma = sqrt(sum((sig_after_ctle_pdf.x.^2).*sig_after_ctle_pdf.y));
12 + adc_lsb = 2*adc_clip/(2^param.ENOB-1);
13 + NS.signa_Q = adc_lsb/sqrt(12);
14 + NS.signa_before_clip = ctle_pulse_sigma;
15 + NS.peak_clip = adc_clip;
16 + NS.p2ptosigna_clip = 2*adc_clip/ctle_pulse_sigma;
17 + quantizaion_noise_pdf = combined_interference_and_noise_pdf; % This is to copy the fields of the structure
18 + [~,adc_ind_right] = min(abs(quantizaion_noise_pdf.x-adc_lsb/2));
19 + [~,adc_ind_left] = min(abs(quantizaion_noise_pdf.x+adc_lsb/2));
20 + quantizaion_noise_pdf.y = zeros(size(quantizaion_noise_pdf.x));
21 + quantizaion_noise_pdf.y(adc_ind_left:adc_ind_right) = 1/(adc_ind_right-adc_ind_left+1);
22 + % Calculate quantization noise PDF after RxFFE
23 + h_rxffe = fom_result.RxFFE(find(fom_result.RxFFE ~= 0));
24 + for irxffe = 1:length(h_rxffe)
25 +     if irxffe ~= param.ffe_pre_tap_len
26 +         quantizaion_noise_pdf_scale = scalePDF(quantizaion_noise_pdf, abs(h_rxffe(irxffe)));
27 +         quantizaion_noise_pdf = conv_fct(quantizaion_noise_pdf, quantizaion_noise_pdf_scale);
28 +     end
29 + end
30 + combined_interference_and_noise_pdf = conv_fct(combined_interference_and_noise_pdf, quantizaion_noise_pdf);
31 + NS.quantizaion_noise_pdf = quantizaion_noise_pdf ;
32 +
```

Create quantization noise PDF

Apply_EQ ad Create_Noise_PDF

```
----- src/com_fcts/Apply_EQ.m -----
index 89de766..7dd55db 100644
@@ -50,6 +50,14 @@ for i=1:param.number_of_s4p_files
    if isequal(chdata(i).type, 'FEXT') || isequal(chdata(i).type, 'THRU')
        eq_pulse = FFE( fom_result.txffe ,fom_result.cur-1 , param.samples_per_ui, eq_pulse );
    end
+
+ % Next 4 lines determine a pulse response required quantization,
+ % at the CTLE output with Tx_ffe
+ chdata(i).ctle_pulse = eq_pulse;
+ [~, sample_start] = min(abs(chdata(i).t-fom_result.sampled_best_sbr_precursors_t(1)));
+ chdata(i).pulse_sampled_w_tx_ffe_ctle = eq_pulse(sample_start:param.samples_per_ui:end);
+ chdata(i).t_sampled_w_tx_ffe_ctle = chdata(i).t(sample_start:param.samples_per_ui:end);
+
+ % chdata(i).ctle_imp_response
+ if OP.RxFFE
+     if isequal(upper(OP.FFE_OPT_METHOD),'MMSE')
----- src/com_fcts/Create_Noise_PDF.m -----
index 0fb3348..586e459 100644
@@ -113,8 +113,13 @@ NS.peak_interference_at_BER=abs(NS.isi_and_xtalk_pdf.x(mx));

% Equation 93A-45
combined_interference_and_noise_pdf = conv_fct(NS.isi_and_xtalk_pdf, NS.noise_pdf);
-PDF=combined_interference_and_noise_pdf;

+%%
+ % Equation 93A-37
+if param.ENOB ~=0
+    [chdata,NS,combined_interference_and_noise_pdf] = adjust_Rx_noise_for_quantization(combined_interference_and_noise_pdf);
+end
+combined_interference_and_noise_cdf=cumsum(combined_interference_and_noise_pdf.y);
+CDF=combined_interference_and_noise_cdf;
+PDF=combined_interference_and_noise_pdf;
```

Create sampled PR (required for noise quantization)

Switch added for quantization noise to be combined to the total noise PDF

get_PSDs

```
----- src/com_fcts/get_PSDs.m -----
index ca49d58..973ed5a 100644
@ -156,12 +156,32 @@ else % find noise for item that set have tx ffe for each loop
    result.S_rj_jn= result.S_rj_jn.*H_rxffe_2;
    result.S_rj_rms = sqrt(sum(result.S_rj_jn)* delta_f);
end
% result.S_jnresult.S_rn
result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn;
% result.S_qn
if(param.ENOB ~=0)
    if OP.INCLUDE_CTLE == 1
        eq_ir = TD_CTLE(chdata(1).uneq_imp_response, param.fb, param.CTLE_fz(1), param.CTLE_fpl(1),
            eq_ir = TD_CTLE(eq_ir, param.fb, param.f_HP(1), param.f_HP(1), 100e100 , G_DC2, param.sampl
    else
        eq_ir = chdata(1).uneq_imp_response;
    end
    ctle_pulse = filter(ones(1, param.samples_per_ui), 1, eq_ir);
    ind_max = find(ctle_pulse == max(ctle_pulse));
    adc_clip = sum(abs([ctle_pulse(ind_max-param.samples_per_ui:-param.samples_per_ui:1); ctle_puls
    adc_lsb = 2*adc_clip/(2^param.ENOB-1);
    sigma_Q = adc_lsb/sqrt(12);
    S_qn = sigma_Q^2/(length(result.S_rn)*delta_f)*ones(size(result.S_rn));
    result.S_qn = S_qn;
    result.qn_rms = sqrt(sum(result.S_qn)* delta_f);
else
    result.S_qn=0;
    result.S_qn_rms = 0;
    % result.S_n
end
result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn;
result.S_n_rms = sqrt(sum(result.S_n)* delta_f);

%%
%% Hisi to be included in MLSE rho eq 178a-28
%% Hisi to be included in MLSE rho eq 178a-28
if OP.COMPUTE_COM
    %% Hisi psd h include CTLE(CFT), TxFFE, and RxFFE but not sigma_X2
    % sampling_offset = mod(cursor_i-1, M)+1; % Commit request 4p4_6, healey_3dj_COM_01_240416
```

Add noise quantization to the PSD computation

optimize_FOM

```
src/com_acts/optimize_fom.m
index elcff2a..e0dlcef 100644
@@ -676,15 +676,19 @@ for i=loop_count
    case 'MMSE'
        OP.WO_TXFFE=0;
        PSD_results=get_PSDs(PSD_results,sbr,cursor_i,txffe,gdc_values(ctle_index),g_DC_low,param,chdata,OP);
-       S_n=PSD_results.S_rn+PSD_results.S_tn+PSD_results.S_xn+PSD_results.S_jn;
+       S_n=PSD_results.S_rn+PSD_results.S_tn+PSD_results.S_xn+PSD_results.S_jn+ PSD_results.S_qn ;
    if 0 % for debug
+       figure(1010132)
        plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_rn*1000/100) , 'disp', 'Srn')
        hold on
-       plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_xn*1000/100) , 'disp', 'Sxn')
+       if(PSD_results.S_xn~=0)
+           plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_xn*1000/100) , 'disp', 'Sxn')
+       end
        plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_tn*1000/100) , 'disp', 'Stn')
        plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_jn*1000/100) , 'disp', 'Sjn')
        plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_n*1000/100) , 'disp', 'Sn')
-       xlim([0 0.5])
+       plot(PSD_results.fvec(1:param.num_ui_RXFF_noise)/param.fb,10*log10(PSD_results.S_qn*1000/100) , 'disp', 'Sqn')
+       xlim([0 0.5])
        % ylim([-190 -160])
        set(gcf, 'defaulttextinterpreter', 'none')
        xlabel('Normalized Frequency')

@@ -1372,3 +1376,5 @@ result.best_bmax=best_bmax;
%AJG021820
result.best_bmin=best_bmin;
result.tail_RSS=best_tail_RSS;
+result.sampled_best_sbr_precursors_t=sampled_best_sbr_precursors_t;
+result.sampled_best_sbr_postcursors_t=sampled_best_sbr_postcursors_t;
```

Add noise quantization to S_n
(total noise PSD)

and

Report out pre and post cursor
times vectors

read_ParamConfigFile and zzz_list_of_changes

```
----- src/com_fcts/read_ParamConfigFile.m -----  
index 7e15f40..f31a703 100644  
@@ -206,6 +206,8 @@ param.a_icn_next = xls_parameter(parameter, 'A_nt', true, param.a_next );% NEXT  
param.levels = xls_parameter(parameter, 'L'); % number of symbols levels (PAM-4 is 4, NRZ is 2)  
param.specBER = xls_parameter(parameter, 'DER_0'); % Target detector error ratio  
param.DER_CDR = xls_parameter(parameter, 'DER_CDR',true,1e-2); % min DER required for a CDR  
+param.ENOB = xls_parameter(parameter, 'ENOB',true,0); % adc number of bits if 0 do not apply quantization  
+param.adc_clip_rate= xls_parameter(parameter, 'adc_clip_rate',true,2*param.specBER); % adc clipping probability  
param.pass_threshold = xls_parameter(parameter, 'COM Pass threshold',false,0); % the pass fail threshold for COM in dB  
param.ERL_pass_threshold = xls_parameter(parameter, 'ERL Pass threshold',false,0); % the pass fail threshold for ERL in dB  
param.VEC_pass_threshold = xls_parameter(parameter, 'VEC Pass threshold',false,0);% the pass fail threshold for VEC in dB only used when PMD_type is
```

Add 2 new keywords

```
----- src/com_fcts/zzz_list_of_changes.m -----  
index 659ba21..e86765f 100644  
@@ -223,4 +223,5 @@ function zzz_list_of_changes  
% r470 beta2 added MLSD proposal from healey_3dj_01_2409  
% r470 remOved support for MLSE_Q  
% r480 betal added new keyword ENOB for quantization. if 0 or missing ignored  
-% r480 betal added function adjust_Rx_noise_for_quantization  
\ No newline at end of file  
+% r480 betal added function adjust_Rx_noise_for_quantization  
+% r480 betal corrected ICN  
\ No newline at end of file
```

Some in code documenting

FD_Processing (ICN fix)

```
----- src/com_fcts/FD_Processing.m -----
index 1f5eec0..ce4aa9a 100644
@@ -262,15 +262,15 @@ for i=1:param.number_of_s4p_files
 else % NEXT or FEXT
   if isequal(chdata(i).type, 'FEXT')
     MDFEXT=sqrt(abs(chdata(i).sdd21f).^2+MDFEXT.^2); % power sum xtk
-     MDFEXT_ICN=sqrt(2*chdata(i).delta_f/param.f2*sum( chdata(i).Aicn^2*PWF(index_f1:index_f2).*abs(MDFEXT(index_f1:index_f2)).^2)); %eq 46
+     MDFEXT_ICN=sqrt(2*chdata(i).delta_f/param.fb*sum( chdata(i).Aicn^2*PWF(index_f1:index_f2).*abs(MDFEXT(index_f1:index_f2)).^2)); %eq 46 corrected for fb
     output_args.MDFEXT_ICN_92_47_mV=MDFEXT_ICN*1000;
   elseif isequal(chdata(i).type, 'NEXT')
     MDNEXT=sqrt(abs(chdata(i).sdd21f).^2+MDNEXT.^2); % power sum xtk
-     MDNEXT_ICN=sqrt(2*chdata(i).delta_f/param.f2*sum( chdata(i).Aicn^2*PWF(index_f1:index_f2).*abs(MDNEXT(index_f1:index_f2)).^2)); %eq 47
+     MDNEXT_ICN=sqrt(2*chdata(i).delta_f/param.fb*sum( chdata(i).Aicn^2*PWF(index_f1:index_f2).*abs(MDNEXT(index_f1:index_f2)).^2)); %eq 47 corrected for fb
     output_args.MDNEXT_ICN_92_46_mV=MDNEXT_ICN*1000;
   end
   PSXT=sqrt((abs(chdata(i).sdd21f)*chdata(i).Aicn).^2+PSXT.^2); % power sum xtk
-   ICN=sqrt(2*chdata(i).delta_f/param.f2*sum( PWF(index_f1:index_f2).*abs(PSXT(index_f1:index_f2)).^2));
+   ICN=sqrt(2*chdata(i).delta_f/param.fb*sum( PWF(index_f1:index_f2).*abs(PSXT(index_f1:index_f2)).^2)); % corrected for fb
   output_args.ICN_mV=ICN*1000;
   ICN_test=norm([MDFEXT_ICN MDNEXT_ICN]);
   if OP.PLOT_CM && OP.DISPLAY_WINDOW
```

Replace f2 for fb

COM 4.8beta1 to 4.8beta2 change
fixed typo's and updated from
shakiba_3dj_COM_02_241001 to
shakiba_3dj_COM_01_241029

adjust_Rx_noise_for_quantization (*incrémental changes*)

```
adjust_Rx_noise_for_quantization.m -4+6
/src/com_fcts/adjust_Rx_noise_for_quantization.m
View

-----
10  adc_clip = -CDF_inv_ev(2*param.specBER, sig_after_ctle_pdf, sig_after_ctle_cd
11  ctile_pulse_sigma = sqrt(sum((sig_after_ctle_pdf.x.^2).*sig_after_ctle_pdf.y))
12  adc_lsb = 2*adc_clip/(2^param.ENOB-1);
- NS.sigma_Q = adc_lsb/sqrt(12);
+ NS.sigma_Q = adc_lsb/sqrt(12);
13  NS.sigma_before_clip = ctile_pulse_sigma;
14  NS.peak_clip = adc_clip;
15  NS.p2ptosigma_clip = 2*adc_clip/ctile_pulse_sigma;
16  NS.p2ptosigma_clip = 2*adc_clip/ctile_pulse_sigma;
17  quantizaion_noise_pdf = combined_interference_and_noise_pdf; % This is to cop
- [~,adc_ind_right] = min(abs(quantizaion_noise_pdf.x-adc_lsb/2));
- [~,adc_ind_left] = min(abs(quantizaion_noise_pdf.x+adc_lsb/2));
18  + % [~,adc_ind_right] = min(abs(quantizaion_noise_pdf.x-adc_lsb/2));
19  + % [~,adc_ind_left] = min(abs(quantizaion_noise_pdf.x+adc_lsb/2));
20  + adc_ind_right= find ( min(abs(quantizaion_noise_pdf.x-adc_lsb/2)) == abs(
21  + adc_ind_left= find ( min(abs(quantizaion_noise_pdf.x+adc_lsb/2)) == abs(
22  quantizaion_noise_pdf.y = zeros(size(quantizaion_noise_pdf.x));
23  quantizaion_noise_pdf.y(adc_ind_left:adc_ind_right) = 1/(adc_ind_right-adc_in
24  % Calculate quantization noise PDF after RxFFE

-----
30  end
31  end
32  combined_interference_and_noise_pdf = conv_fct(combined_interference_and_nois
- NS.quantizaion_noise_pdf = quantizaion_noise_pdf ;
+ NS.Quantizaion_noise_pdf = quantizaion_noise_pdf ;
33
34
```

```
adjust_Rx_noise_for_quantization.m -13+14
/src/com_fcts/adjust_Rx_noise_for_quantization.m
View

19  - % [~,adc_ind_left] = min(abs(quantizaion_noise_pdf.x+adc_lsb/2));
20  - adc_ind_right= find ( min(abs(quantizaion_noise_pdf.x-adc_lsb/2)) == abs(quantizaion_noise_pdf.x-adc_lsb/2)
21  - adc_ind_left= find ( min(abs(quantizaion_noise_pdf.x+adc_lsb/2)) == abs(quantizaion_noise_pdf.x+adc_lsb/2)
22  - quantizaion_noise_pdf.y = zeros(size(quantizaion_noise_pdf.x));
23  - quantizaion_noise_pdf.y(adc_ind_left:adc_ind_right) = 1/(adc_ind_right-adc_ind_left+1);
24  + NS.p2ptosigma_clip = 2*adc_clip/ctile_pulse_sigma;
17  + NS.p2ptosigma_clip = 2*adc_clip/ctile_pulse_sigma;
18  + quantizaion_noise_pdf = combined_interference_and_noise_pdf; % This is to copy the fields of the structure
19  + [~,adc_ind_right] = min(abs(quantizaion_noise_pdf.x-adc_lsb/2));
20  + [~,adc_ind_left] = min(abs(quantizaion_noise_pdf.x+adc_lsb/2));
21  + % adc_ind_right= find ( min(abs(quantizaion_noise_pdf.x-adc_lsb/2)) == abs(quantizaion_noise_pdf.x-adc_lsb/2)
22  + % adc_ind_left= find ( min(abs(quantizaion_noise_pdf.x+adc_lsb/2)) == abs(quantizaion_noise_pdf.x+adc_lsb/2)
23  + quantizaion_noise_pdf.y = zeros(size(quantizaion_noise_pdf.x));
24  + quantizaion_noise_pdf.y(adc_ind_left:adc_ind_right) = 1/(adc_ind_right-adc_ind_left+1);
24 25  % Calculate quantization noise PDF after RxFFE
25 26  h_rxffe = fom_result.RxFFE(find(fom_result.RxFFE ~= 0));
26 27  for irxffe = 1:length(h_rxffe)
27 28  if irxffe ~= param.ffe_pre_tap_len
28  - quantizaion_noise_pdf_scale = scalePDF(quantizaion_noise_pdf, abs(h_rxffe(irxffe)));
29  - quantizaion_noise_pdf = conv_fct(quantizaion_noise_pdf, quantizaion_noise_pdf_scale);
29  + quantizaion_noise_pdf_scale = scalePDF(quantizaion_noise_pdf, abs(h_rxffe(irxffe)));
30  + quantizaion_noise_pdf = conv_fct(quantizaion_noise_pdf, quantizaion_noise_pdf_scale);
30 31  end
31 32  end
32  - combined_interference_and_noise_pdf = conv_fct(combined_interference_and_noise_pdf, quantizaion_noise_pdf);
33  - NS.Quantizaion_noise_pdf = quantizaion_noise_pdf ;
33  + combined_interference_and_noise_pdf = conv_fct(combined_interference_and_noise_pdf, quantizaion_noise_pdf);
34  + NS.quantizaion_noise_pdf = quantizaion_noise_pdf ;
34 35
```


Output_Arg_Fill (*incrémental changes*)

Output_Arg_Fill.m +6
/src/com_fcts/Output_Arg_Fill.m

```
-----  
158  
159 % r259 putting COM, VEO and loss last in report  
160 %         output_args.VEO_normalized = (A_s-A_ni)/A_s;  
161 + if param.ENOB ~= 0  
162 +     output_args.sgm_Q =Noise_Struct.sigma_Q;  
163 +     output_args.sigma_before_clip = Noise_Struct.signa_before_clip;  
164 +     output_args.peak_clip = Noise_Struct.peak_clip;  
165 +     output_args.P2ptopsigma_clip = Noise_Struct.p2ptosigma_clip;  
166 + end  
167     output_args.VEC_dB = COM_SNR_Struct.VEC_dB;  
168     output_args.VEO_mV = COM_SNR_Struct.VEO_mV;  
169     if OP.RX_CALIBRATION ==0 && OP.EW == 1  
-----
```

Output_Arg_Fill.m -2+2
/src/com_fcts/Output_Arg_Fill.m

```
-----  
160 160 %         output_args.VEO_normalized = (A_s-A_ni)/A_s;  
161 161 if param.ENOB ~= 0  
162 162     output_args.sgm_Q =Noise_Struct.sigma_Q;  
163 -     output_args.sigma_before_clip = Noise_Struct.signa_before_clip;  
163 +     output_args.sigma_before_clip = Noise_Struct.sigma_before_clip;  
164 164     output_args.peak_clip = Noise_Struct.peak_clip;  
165 -     output_args.P2ptopsigma_clip = Noise_Struct.p2ptosigma_clip;  
165 +     output_args.P2ptopsigma_clip = Noise_Struct.p2ptosigma_clip;  
166 166 end  
167 167     output_args.VEC_dB = COM_SNR_Struct.VEC_dB;  
168 168     output_args.VEO_mV = COM_SNR_Struct.VEO_mV;  
-----
```


read_ParamConfigFile

zzz_list_of_changes

```
read_ParamConfigFile.m -1+1
/src/com_fcts/read_ParamConfigFile.m
View

-----

533         error('MLSD nnot presently no supported for VEC')
534     end
535     if OP.COM_CONTRIBUTION_CURVES ~=0
-   warning("COM_CONTRIBUTION_CURVES not functional yet with MLSE")
+   warning('COM_CONTRIBUTION_CURVES not functional yet with MLSE')
536 +   warning('COM_CONTRIBUTION_CURVES not functional yet with MLSE')
537     OP.COM_CONTRIBUTION_CURVES=0;
538     end
539 end

-----

zzz_list_of_changes.m -1+1
/src/com_fcts/zzz_list_of_changes.m
View

-----

224     % r470 remOved support for MLSE_Q
225     % r480 beta1 added new keyword ENOB for quantization. if 0 or missing ignore
226     % r480 beta1 added function adjust_Rx_noise_for_quantization
-   % r480 beta1 corrected ICN
227 +   % r480 beta2 fixed typo's and updated from shakiba_3dj_COM_02_241001 to shak
```

Thank You!