Quantization Noise Feature in COM Matlab Code

COM Commit Request Number 4p6_5

Hossein Shakiba
Huawei Technologies Canada
October 2024

Introduction

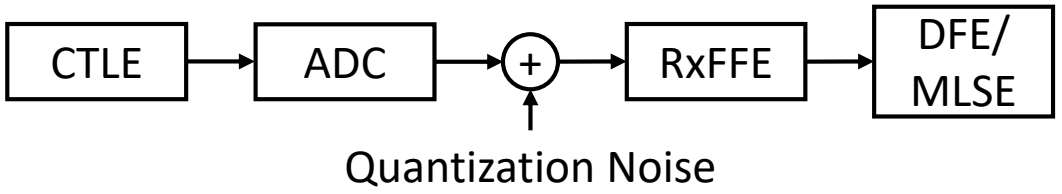
- In contribution *shakiba_3dj_02_2405.pdf* effects of including ADC quantization noise in COM channel compliance verifications was analyzed and its significance was demonstrated
- Similarly, contribution *healey_3dj_01b_2405.pdf* also considered adding a new eta_1 noise term between CTLE and RxFFE to represent quantization noise
- This contribution also suggested another option that scales the existing eta_0 noise term
- Straw poll #1 in the May interim meeting did not support addition of a new noise term

> Straw Poll #1
>
> I support adding a new noise term (such as 'eta_1' in healey_3dj_01a_2405, slide 6) to the COM reference receiver.
>
> Results (all) Y: 13, , N: 37 , A: 31

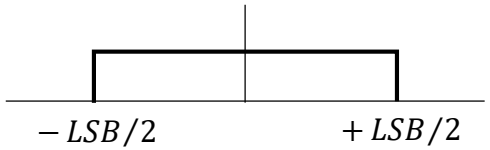
- Despite that, there has been continuous interest and request to include this capability to the COM Matlab code for the purposes of investigations and explorations
- With the capability added, parameter ENOB, which represents ADC number of bits, can be set to a non-positive number (defaulted to 0) to disable the feature for standard COM channel compliance purposes

Background

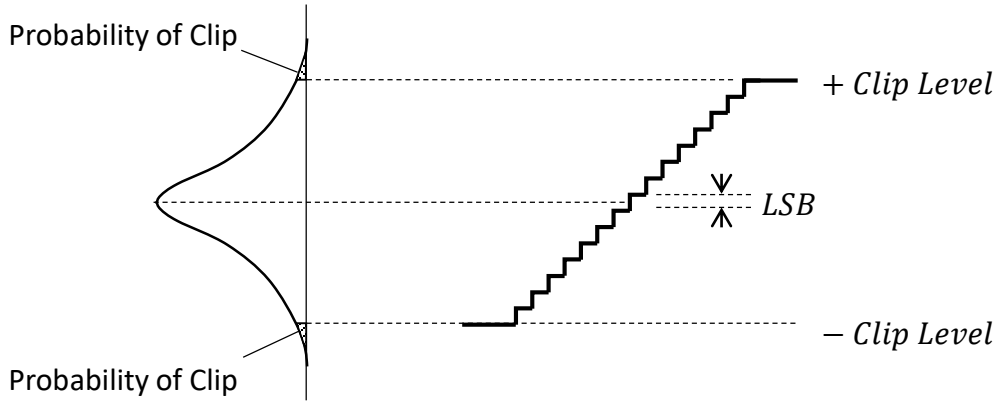
- Quantization noise is a new noise term added between CTLE and RxFFE



- It is modeled by a uniform distribution over –LSB/2 to +LSB/2



$$\sigma_{qn} = \frac{LSB}{\sqrt{12}}$$



- ADC clip level is chosen so that clipping frequency is low
  - User defined and defaulted to the target error rate (parameter adc_clip_rate)
- For details see contribution *shakiba_3dj_02_2405.pdf*

Changes to the Code (1 of 8)

1) Insert 5 lines:

- Saves pulse response and its sampled values and sampling times at the CTLE output

```
841 -       chdata(i).eq_imp_response=eq_ir;
842 -       eq_pulse=filter(ones(1, param.samples_per_ui), 1, chdata(i).eq_imp_response);
843
844 -       if isequal(chdata(i).type, 'FEXT') || isequal(chdata(i).type, 'THRU')
845 -           eq_pulse = FFE( fom_result.txffe ,fom_result.cur-1 , param.samples_per_ui,  eq_pulse );
846 -       end
```

```
841 -       chdata(i).eq_imp_response=eq_ir;
842 -       eq_pulse=filter(ones(1, param.samples_per_ui), 1, chdata(i).eq_imp_response);
843         % Next 4 lines save pulse response and its sampled values and times at the CTLE output, needed for quantization noise
844 -       chdata(i).ctle_pulse = eq_pulse;
845 -       [~, sample_start] = min(abs(chdata(i).t-fom_result.sampled_best_sbr_precursors_t(1)));
846 -       chdata(i).ctle_pulse_s = eq_pulse(sample_start:param.samples_per_ui:end);
847 -       chdata(i).t_s = chdata(i).t(sample_start:param.samples_per_ui:end);
848
849 -       if isequal(chdata(i).type, 'FEXT') || isequal(chdata(i).type, 'THRU')
850 -           eq_pulse = FFE( fom_result.txffe ,fom_result.cur-1 , param.samples_per_ui,  eq_pulse );
851 -       end
```

Changes to the Code (2 of 8)

2) Insert 28 lines:

- If ENOB is positive, calculates ADC clipping level and LSB and quantization noise PDF at the injection point and after RxFFE and adds it to the overall noise PDF

```
1483    % Equation 93A-45
1484 -  combined_interference_and_noise_pdf = conv_fct(NS.isi_and_xtalk_pdf, NS.noise_pdf);
1485 -  PDF=combined_interference_and_noise_pdf;
1486
1487    % Equation 93A-37
```

```
1488    % Equation 93A-45
1489 -  combined_interference_and_noise_pdf = conv_fct(NS.isi_and_xtalk_pdf, NS.noise_pdf);
1490    % The following if statement calculates quantization noise PDF and adds i to the total noise if ENOB is positive
1491 -  if param.ENOB > 0
1492 -      ctle_signal_pdf = get_pdf_from_sampled_signal(chdata(1).ctle_pulse_s, param.levels, param.delta_y);
1493 -      ctle_signal_noise_pdf = conv_fct(ctle_signal_pdf, combined_interference_and_noise_pdf);
1494 -      ctle_signal_noise_cdf = cumsum(ctle_signal_noise_pdf.y);
1495 -      adc_clip = -CDF_inv_ev(param.adc_clip_rate, ctle_signal_noise_pdf, ctle_signal_noise_cdf);
1496 -      ctle_pulse_sigma = sqrt(sum((ctle_signal_pdf.x.^2).*ctle_signal_pdf.y));
1497 -      adc_lsb = 2*adc_clip/(2^param.ENOB-1);
1498 -      NS.sigma_Q = adc_lsb/sqrt(12);
1499 -      NS.sigma_before_clip = ctle_pulse_sigma;
1500 -      NS.peak_clip = adc_clip;
1501 -      NS.p2ptosigma_clip = 2*adc_clip/ctle_pulse_sigma;
1502 -      quantization_noise_pdf = combined_interference_and_noise_pdf; % This is to copy the fields of the structure
1503 -      [~, adc_ind_right] = min(abs(quantization_noise_pdf.x-adc_lsb/2));
1504 -      [~, adc_ind_left] = min(abs(quantization_noise_pdf.x+adc_lsb/2));
1505 -      quantization_noise_pdf.y = zeros(size(quantization_noise_pdf.x));
1506 -      quantization_noise_pdf.y(adc_ind_left:adc_ind_right) = 1/(adc_ind_right-adc_ind_left+1);
1507        % Calculate quantization noise PDF after RxFFE
1508 -      h_rxffe = fom_result.RxFFE(find(fom_result.RxFFE ~= 0));
1509 -      for irxffe = 1:length(h_rxffe)
1510 -          if irxffe ~= param.ffe_pre_tap_len
1511 -              quantization_noise_pdf_scale = scalePDF(quantization_noise_pdf, abs(h_rxffe(irxffe)));
1512 -              quantization_noise_pdf = conv_fct(quantization_noise_pdf, quantization_noise_pdf_scale);
1513 -          end
1514 -      end
1515 -      combined_interference_and_noise_pdf = conv_fct(combined_interference_and_noise_pdf, quantization_noise_pdf);
1516 -      NS.quantization_noise_pdf = quantization_noise_pdf;
1517 -  end
1518 -  PDF=combined_interference_and_noise_pdf;
```

Changes to the Code (3 of 8)

3) Insert 7 lines

- If ENOB is positive, registers quantization noise data

```
2852    % r259 putting COM, VEO and loss last in report
2853    %         output_args.VEO_normalized = (A_s-A_ni)/A_s;
2854 -  output_args.VEC_dB = COM_SNR_Struct.VEC_dB;
2855 -  output_args.VEO_mV = COM_SNR_Struct.VEO_mV;
```

```
2885    % r259 putting COM, VEO and loss last in report
2886    %         output_args.VEO_normalized = (A_s-A_ni)/A_s;
2887    % The following if statement registers quantization noise data if ENOB is positive
2888 -  if param.ENOB > 0
2889 -      output_args.sgm_Q = Noise_Struct.sigma_Q;
2890 -      output_args.sigma_before_clip = Noise_Struct.sigma_before_clip;
2891 -      output_args.peak_clip = Noise_Struct.peak_clip;
2892 -      output_args.p2ptosigma_clip = Noise_Struct.p2ptosigma_clip;
2893 -  end
2894 -  output_args.VEC_dB = COM_SNR_Struct.VEC_dB;
2895 -  output_args.VEO_mV = COM_SNR_Struct.VEO_mV;
```

Changes to the Code (4 & 5 of 8)

4 & 5) Insert 19 lines, and insert 1 line and change 1 line

- If ENOB is positive, calculates quantization noise PSD, otherwise sets it to 0
- Adds quantization noise PSD to the overall PSD

```
5308        %  result.S_jn
5309 -      result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn;
5310 -      result.S_n_rms = sqrt(sum(result.S_n)* delta_f);
5311
5312        %%
5313        %% Hisi to be included in MLSE rho eq 178a-28
```

```
5348        %  result.S_jn
5349        % The following if statement calculates or zeroes quantization noise PSD based on ENOB
5350 -      if param.ENOB > 0
5351 -          if OP.INCLUDE_CTLE == 1
5352 -              eq_ir = TD_CTLE(chdata(1).uneq_imp_response, param.fb, param.CTLE_fz(1), param.CTLE_fp1(1), param.CTLE_fp2(1), G_DC, param.samples_per_ui);
5353 -              eq_ir = TD_CTLE(eq_ir, param.fb, param.f_HP(1), param.f_HP(1), 100e100 , G_DC2, param.samples_per_ui);
5354 -          else
5355 -              eq_ir = chdata(1).uneq_imp_response;
5356 -          end
5357 -          ctle_pulse = filter(ones(1, param.samples_per_ui), 1, eq_ir);
5358 -          ind_max = find(ctle_pulse == max(ctle_pulse));
5359 -          adc_clip = sum(abs([ctle_pulse(ind_max-param.samples_per_ui:-param.samples_per_ui:1); ctle_pulse(ind_max:param.samples_per_ui:end)]));
5360 -          adc_lsb = 2*adc_clip/(2^param.ENOB-1);
5361 -          sigma_Q = adc_lsb/sqrt(12);
5362 -          S_qn = sigma_Q^2/(length(result.S_rn)*delta_f)*ones(size(result.S_rn));
5363 -          result.S_qn = S_qn;
5364 -          result.qn_rms = sqrt(sum(result.S_qn)* delta_f);
5365 -      else
5366 -          result.S_qn = 0;
5367 -      end
5368        % The folloeing line change adds quantization noise PSD to the overall noise PSD
5369        %     result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn;
5370 -      result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn;
5371 -      result.S_n_rms = sqrt(sum(result.S_n)* delta_f);
```

Changes to the Code (6 of 8)

6) Insert 1 line and change 1 line

- Adds quantization noise PSD to the overall PSD

```
7759 -                      switch upper(OP.FFE_OPT_METHOD)
7760 -                          case 'MMSE'
7761 -                              OP.WO_TXFFE=0;
7762 -                              PSD_results=get_PSDs(PSD_results,sbr,cursor_i,txffe,gdc_values(ctle_index),g_DC_low,param,chdata,OP);
7763 -                              S_n=PSD_results.S_rn+PSD_results.S_tn+PSD_results.S_xn+PSD_results.S_jn;
7764 -                              if 0 % for debug
```

```
7820 -                  switch upper(OP.FFE_OPT_METHOD)
7821 -                      case 'MMSE'
7822 -                          OP.WO_TXFFE=0;
7823 -                          PSD_results=get_PSDs(PSD_results,sbr,cursor_i,txffe,gdc_values(ctle_index),g_DC_low,param,chdata,OP);
7824                           % The following line change adds quantization noise PSD to the overall noise PSD
7825 %                           S_n=PSD_results.S_rn+PSD_results.S_tn+PSD_results.S_xn+PSD_results.S_jn;
7826 -                         S_n=PSD_results.S_rn+PSD_results.S_tn+PSD_results.S_xn+PSD_results.S_jn+PSD_results.S_qn;
7827 -                         if 0 % for debug
```

Changes to the Code (7 of 8)

7) Insert 3 lines

- Registers sampling time

```
8456      %AJG021820
8457 -    result.best_bmin=best_bmin;
8458 -    result.tail_RSS=best_tail_RSS;
8459      function param=parameter_size_adjustment(param,OP)
```

```
8519      %AJG021820
8520 -    result.best_bmin=best_bmin;
8521 -    result.tail_RSS=best_tail_RSS;
8522      % The following two lines register sampling times needed for quantization noise
8523 -    result.sampled_best_sbr_precursors_t = sampled_best_sbr_precursors_t;
8524 -    result.sampled_best_sbr_postcursors_t = sampled_best_sbr_postcursors_t;
8525      function param=parameter_size_adjustment(param,OP)
```

Changes to the Code (8 of 8)

8) Insert 3 lines

- Reads parameters ENOB and adc_clip_rate from COM configuration file

```
9223      %% addd default to support multiple packages
9224 -    param.a_thru = xls_parameter(parameter, 'A_v', true, 0.5); % Victim differential peak source output voltage (half of peak to peak)
9225 -    param.a_fext = xls_parameter(parameter, 'A_fe', true,0.5); % FEXT aggressor differential peak source output voltage (half of peak to peak)
9226 -    param.a_next = xls_parameter(parameter, 'A_ne', true,0.5); % NEXT aggressor differential peak source output voltage (half of peak to peak)
9227 -    param.a_icn_fext = xls_parameter(parameter, 'A_ft', true, param.a_fext); % FEXT aggressor amplitude for ICN. Defaults to A_fe if not specified
9228 -    param.a_icn_next = xls_parameter(parameter, 'A_nt', true, param.a_next );% NEXT aggressor amplitude for ICN. Defaults to A_ne if not specified
9229 -    param.levels = xls_parameter(parameter, 'L'); % number of symbols levels (PAM-4 is 4, NRZ is 2)
9230 -    param.specBER = xls_parameter(parameter, 'DER_0'); % Target detector error ratio
9231 -    param.DER_CDR = xls_parameter(parameter, 'DER_CDR',true,1e-2); % min DER required for a CDR
9232 -    param.pass_threshold = xls_parameter(parameter, 'COM Pass threshold',false,0); % the pass fail threshold for COM in dB
9233 -    param.ERL_pass_threshold = xls_parameter(parameter, 'ERL Pass threshold',false,0); % the pass fail threshold for ERL in dB
9234 -    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 C2M

9289      %% addd default to support multiple packages
9290 -    param.a_thru = xls_parameter(parameter, 'A_v', true, 0.5); % Victim differential peak source output voltage (half of peak to peak)
9291 -    param.a_fext = xls_parameter(parameter, 'A_fe', true,0.5); % FEXT aggressor differential peak source output voltage (half of peak to peak)
9292 -    param.a_next = xls_parameter(parameter, 'A_ne', true,0.5); % NEXT aggressor differential peak source output voltage (half of peak to peak)
9293 -    param.a_icn_fext = xls_parameter(parameter, 'A_ft', true, param.a_fext); % FEXT aggressor amplitude for ICN. Defaults to A_fe if not specified
9294 -    param.a_icn_next = xls_parameter(parameter, 'A_nt', true, param.a_next );% NEXT aggressor amplitude for ICN. Defaults to A_ne if not specified
9295 -    param.levels = xls_parameter(parameter, 'L'); % number of symbols levels (PAM-4 is 4, NRZ is 2)
9296 -    param.specBER = xls_parameter(parameter, 'DER_0'); % Target detector error ratio
9297 -    param.DER_CDR = xls_parameter(parameter, 'DER_CDR',true,1e-2); % min DER required for a CDR
9298      % The following two lines read parameters ENOB and adc_clip_rate from the configuration spreadsheet
9299 -    param.ENOB = xls_parameter(parameter, 'ENOB', 'false', 0); % ADC Number of bits
9300 -    param.adc_clip_rate = xls_parameter(parameter, 'adc_clip_rate', 'false', 2*param.specBER); % ADC clipping probability
9301 -    param.pass_threshold = xls_parameter(parameter, 'COM Pass threshold',false,0); % the pass fail threshold for COM in dB
9302 -    param.ERL_pass_threshold = xls_parameter(parameter, 'ERL Pass threshold',false,0); % the pass fail threshold for ERL in dB
9303 -    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 C2M
```

Example Outputs

		After Change	
	Before Change	**adc_clip_rate = 2*specBER**	**adc_clip_rate = 0.2*specBER**
Without Quantization (ENOB = 0)	DER_MLSE_trunc: 2.0764e-06 Q_budget_adj: 0 COM_from_matlab: -0.7246 DER_MLSE: 2.0764e-06 DER_DFE: 1.4608e-04 COM: 1.0876 delta_com: 1.8122	DER_MLSE_trunc: 2.0764e-06 Q_budget_adj: 0 COM_from_matlab: -0.7246 DER_MLSE: 2.0764e-06 DER_DFE: 1.4608e-04 COM: 1.0876 delta_com: 1.8122	DER_MLSE_trunc: 2.0764e-06 Q_budget_adj: 0 COM_from_matlab: -0.7246 DER_MLSE: 2.0764e-06 DER_DFE: 1.4608e-04 COM: 1.0876 delta_com: 1.8122
With Quantization (ENOB = 6)	**NA**	DER_MLSE_trunc: 3.5894e-05 Q_budget_adj: 0 COM_from_matlab: -1.7698 DER_MLSE: 3.5894e-05 DER_DFE: 7.4213e-04 COM: -0.1862 delta_com: 1.5836	DER_MLSE_trunc: 4.0593e-05 Q_budget_adj: 0 COM_from_matlab: -1.8263 DER_MLSE: 4.0593e-05 DER_DFE: 8.0210e-04 COM: -0.2499 delta_com: 1.5764

- It is recommended that this change be made along with the change for MLSE truncation

Thank You ☺

Hossein Shakiba
Huawei Technologies Canada
October 2024