

COM Implementation example of receiver interference tolerance specified in 178A.1.7.5 (RiTol) Commit Request 4p9_5

Richard Mellitz, Samtec

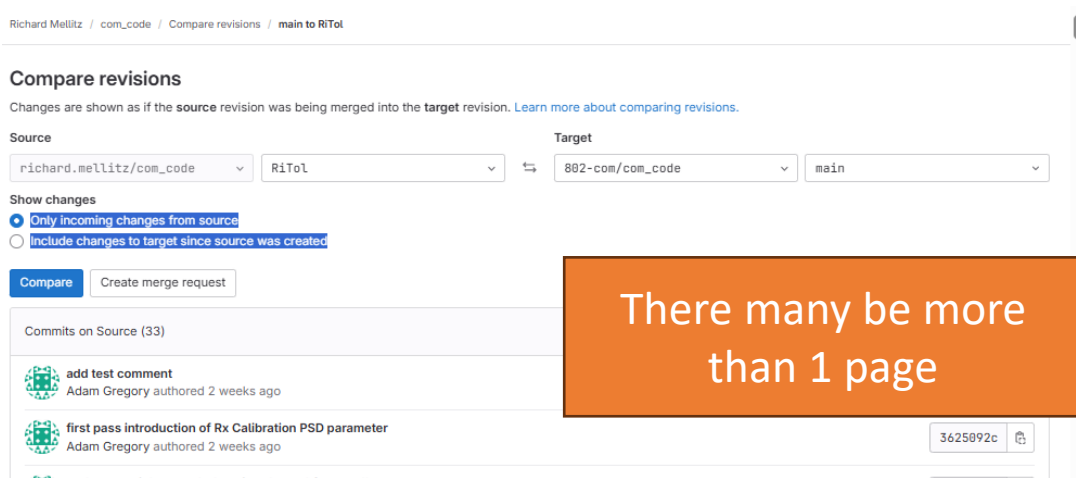
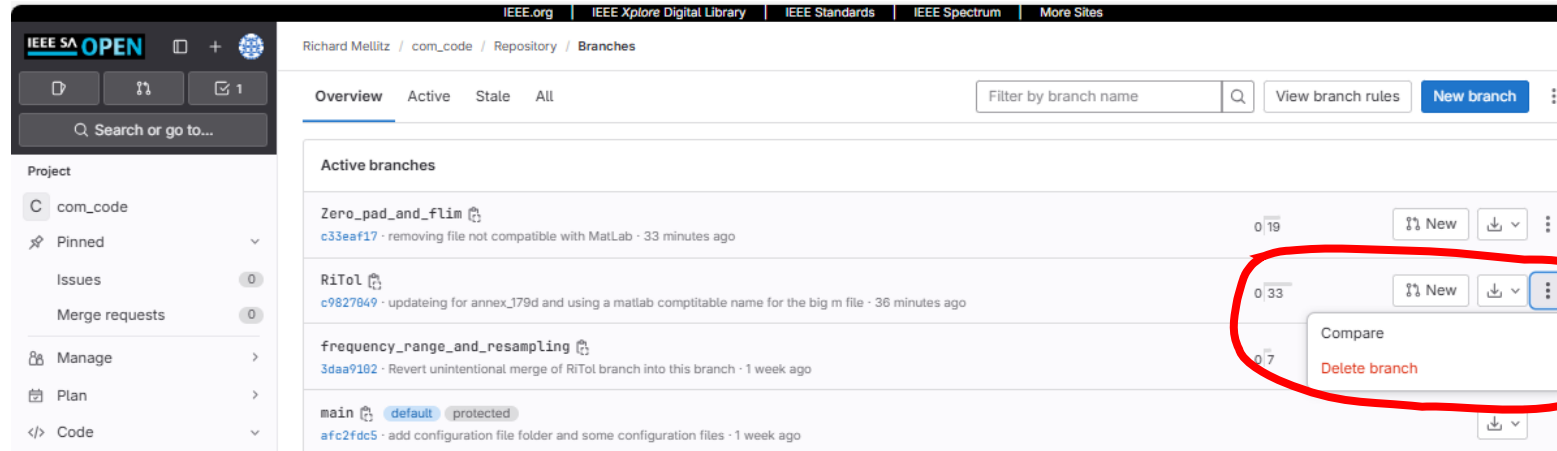
Adam Gregory, Samtec

IEEE 802.3 Channel Operating Margin (COM) Open Source Project Ad Hoc,
New Orleans, LA May 2025 Interim

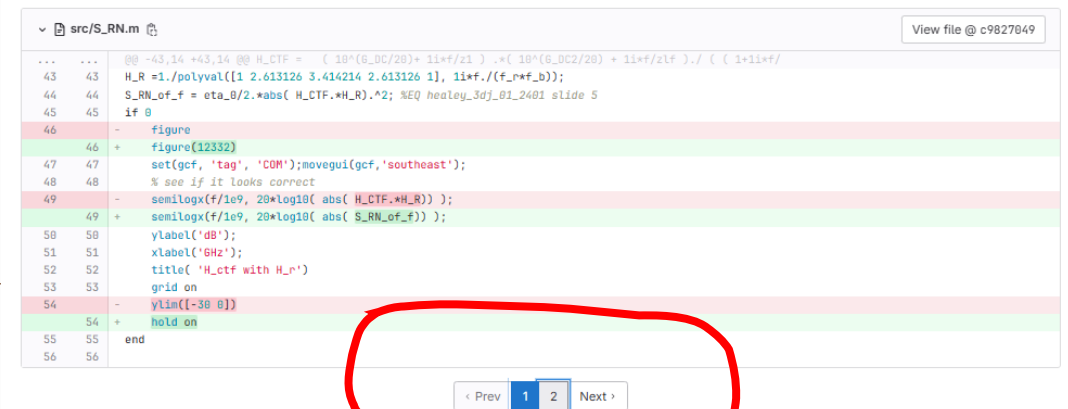
Purpose for Commit Request

- ❑ COM Implementation example of receiver interference tolerance specified in 178A.1.7.5 (RiTol)
- ❑ Limit configuration templates provided
 - Values and settings in the spread sheet may not be align with a standard or ongoing work. Please check before using.
- ❑ Branch: RiTOL
- ❑ Branch test code (will disappear after merge)
 - https://opensource.ieee.org/richard.mellitz/com_code/-/raw/RiTol/release/com_ieee8023_4p9p1_RiTOL.m?ref_type=heads&inline=false

Checking changes of a branch to the last release



There may be more than 1 page



Operation

- ❑ The keyword RX_calibration and usages with RxFFE turns on the calibration operations.
 - Prior version without RxFFE operate as in pervious versions
- ❑ Injected noise power spectral density included in to get_PDSs
- ❑ 2 new functions
 - S_in
 - Create injected noise PSD
 - N_s
 - Creates noise source
- ❑ Changes to 17 other functions to modify dialog, CLI, and looping..
- ❑ Operation is similar to prior versions.
 - i.e. looping until 3 dB COM is reached.
- ❑ The noise sigma BBN step is a starting step size as before
- ❑ The change is that crosstalk channels can be consider and entered and an added channel may be cascaded for running CR COM

N_s.m

- ❑ Creates noise source to be used in S_in

src/N_s.m 0 → 100644 View file @ c9827049

```
25 + % HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
26 + % SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
27 + % LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
28 + % DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
29 + % THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
30 + % (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
31 + % OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
32 + %
33 + % SPDX-License-Identifier: BSD-3-Clause
34 + %%
35 + % computes Ns(f) PSD on rx compliance injected noise base on sigma_ns
36 + % Ns_of_f the single-sided power spectral density at the noise generator output in V^2/Hz
37 + % hence the incoming f is > 0
38 + f_b=param.fb; %align variable with standard variable usage
39 + f_hp=param.f_hp; %align variable with standard variable usage
40 + inq=find( f<=f_b/2,1,'last');
41 + RIT_REF_PTR=lower(OP.RIT_REF_PTR);
42 + Ns_of_f=zeros(1,length(f));
43 + switch RIT_REF_PTR
44 +     case 'clause_178'
45 +         Ns_of_f(1:inq)=ones(1,inq)*2*sigma_ns^2/f_b;% described in 178.9.3.4.2 d1.5
46 +     case 'clause_179'
47 +         if f_hp <= 0
48 +             error('Parameter f_hp must be greater than 0 when using Clause 179');
49 +         end
50 +         beta=1-(2*f_hp/f_b)*atan(f_b/(2*f_hp));
51 +         Ns_of_f(1:inq)=(2*sigma_ns^2)/(beta*f_b)* (f(1:inq)./f_hp).^2 ./ (1+(f(1:inq)/f_hp).^2);% equation 179-23 d1.5
52 +     case 'annex_176d'
53 +         if f_hp <= 0
54 +             error('Parameter f_hp must be greater than 0 when using Clause 179');
55 +         end
56 +         beta=1-(2*f_hp/f_b)*atan(f_b/(2*f_hp));
57 +         Ns_of_f(1:inq)=(2*sigma_ns^2)/(beta*f_b)* (f(1:inq)./f_hp).^2 ./ (1+(f(1:inq)/f_hp).^2);% equation 179-23 d1.5
58 +     otherwise
59 +         error('unsuported RIT Reference Pointer "%s" ', RIT_REF_PTR);
60 + end
61 + if 0
62 +     figure(99)
63 +     set(gcf, 'tag', 'COM');movegui(gcf,'southeast');
64 +     % see if it looks correct
65 +     plot(f/1e9, Ns_of_f);
66 +     ylabel('pwr');
67 +     xlabel('GHz');
68 +     title('N_s')
69 +     grid on
70 +     hold on
71 +
72 + end
```

S_IN

- ❑ Created interference noise PSD

```
src/S_IN.m 0 → 100644 View file @ c9827049

28 + % DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
29 + % THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
30 + % (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
31 + % OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
32 + %
33 + % SPDX-License-Identifier: BSD-3-Clause
34 + %%
35 + % computes PSD on rx compliance injected noise
36 + Hn21=noise_path;% as function of f. the VTF of the passed noise channel with rx package added.
37 + p1=param.CTLE_fp1(1);
38 + z1=param.CTLE_fz(1);
39 + p2=param.CTLE_fp2(1);
40 + zlf=param.f_HP(1);
41 + plf=param.f_HP(1);
42 + f_b=param.fb;
43 + f_r=param.f_r;
44 + sigma_ns=param.sigma_ns+eps(0);
45 + H_CTF = ( 10^(G_DC/20)+ 1i*f/z1 ) .* ( 10^(G_DC2/20) + 1i*f/zlf )./ ( ( 1+1i*f/p1 ) .* ( 1+1i*f/p2 ) .* ( 1+1i*f/plf));
46 + H_R =1./polyval([1 2.613126 3.414214 2.613126 1], 1i*f./(f_r*f_b));
47 + Ns_of_f = N_s(f,param, sigma_ns,OP);
48 + S_IN_of_f = Ns_of_f./2.*abs( Hn21.*H_CTF.*H_R).^2; %
49 + if 0
50 +     delta_f=f(2)-f(1);
51 +     S_IN_of_f_rms=sqrt(sum(S_IN_of_f)* delta_f);
52 +     figure(12332)
53 +     set(gcf, 'tag', 'COM');movegui(gcf,'southeast');
54 +     % see if it looks correct
55 +     semilogx(f/1e9, 20*log10( abs( S_IN_of_f) ) );
56 +     ylabel('dB');
57 +     xlabel('GHz');
58 +     title( 'S_IN_of_f including H_ctf with H_r')
59 +     grid on
60 +     hold off
61 +
62 + end
63 +
```

get_PSDs

- ❑ Adding S_IN to noise PSD
- ❑ When not performing an Rx Calibration S_IN is zero.

```
src/get_PSDs.m
View file @ c9827049

... 78 78  % @ -78,9 +78,33 @ if OP.WD_TXFFE % to speed up loop find an only first time ctile is case
79 79  S_rn= [real( S_rn(1)), S_rn(2:end-1), real( S_rn(end)), conj( S_rn(end-1:-1:2))];
80 80  result.S_rn=S_rn;
81 81  result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
82 82  % compute S_in ( eq 178A-24 dB.2 )
83 83  % S_in = 0 when Rx Calibration is not performed
84 84  if OP.PSORXCAL
85 85  % interpolate to fvec
86 86  result.H_noise=H_interp(chdata(end).sdd2lp,chdata(1).faxis,fvec,f_b);
87 87  % Units are V^2/Hz
88 88  S_IN_of_f=S_IN(fvec,result.H_noise,G_DC,G_DC2,param,OP);
89 89  % Convert single-sided frequency response to conjugate-symmetric
90 90  inn_psd=[real(S_IN_of_f(1)), S_IN_of_f(2:end-1), real(S_IN_of_f(end)), conj(S_IN_of_f(end-1:-1:2))];
91 91  inn_rms = sqrt(sum(inn_psd)* delta_f);
92 92  S_in = sum(reshape(inn_psd, num_ui, M).');
93 93  S_in=S_in(1:num_ui/2+1);
94 94  S_in= [real( S_in(1)), S_in(2:end-1), real( S_in(end)), conj( S_in(end-1:-1:2))];
95 95  result.S_in=S_in;
96 96  result.S_in_rms = sqrt(sum(result.S_in)* delta_f);
97 97  else
98 98  result.S_in=0;
99 99  result.S_in_rms =0;
100 100 end
101 101 else
102 102 result.S_rn=result.S_rn.*H_rxffe_2;
103 103 result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
104 104 % compute S_rn
105 105 result.S_rn=result.S_rn.*H_rxffe_2;
106 106 result.S_rn_rms = sqrt(sum(result.S_rn)* delta_f);
107 107 % compute S_in
108 108 result.S_in=result.S_in.*H_rxffe_2;
109 109 result.S_in_rms = sqrt(sum(result.S_in)* delta_f);
110 110 end
111 111 % find noise for item that set have tx ffe for each loop
112 112 % @ -88,8 +112,13 @ else % find noise for item that set have tx ffe for each loop
113 113 % Crosstalk power spectral density
114 114 if ~OP.COMPUTE_COM % result.S_xn and result.S_xn_rms were found in optimizes_fom and passed in with the variable result
115 115 result.S_xn=0;
116 116 if length(chdata)==1
117 117 for xchan=2:length(chdata)
118 118 if OP.PSORXCAL
119 119 num_channel_files = length(chdata) - 1;
120 120 else
121 121 num_channel_files = length(chdata);
122 122 end
123 123 if num_channel_files==1
124 124 for xchan=2:num_channel_files
125 125 pulse_ctile=filter(ones(1,M),1,chdata(xchan).ctile_imp_response(:).');
126 126 pulse_ctile=[ pulse_ctile(1:floor(length(pulse_ctile)/M)*M) ];
127 127 hk(xchan).k=chdata(xchan).pulse_response_w_CFT_TXFFE_noRxFFE.';
128 128 % @ -193,7 +222,7 @ else % find noise for item that set have tx ffe for each loop
129 129 hext_txffe=filter(txffe,1,hext);
130 130 sig_after_tx_ctile_pdf = get_pdf_from_sampled_signal(hext_txffe,param.levels,OP.BinSize);
131 131 noise_after_ctile_pdf = sig_after_tx_ctile_pdf;
132 132 sigma_noise = sqrt(result.S_rn_rms^2+result.S_xn_rms^2+result.S_tn_rms^2+result.S_rj_rms^2);
133 133 + sigma_noise = sqrt(result.S_in_rms^2+result.S_rn_rms^2+result.S_xn_rms^2+result.S_tn_rms^2+result.S_rj_rms^2);
134 134 noise_after_ctile_pdf.y = 1/(sqrt(2*pi))*sigma_noise*exp(-noise_after_ctile_pdf.x.^2/(2*sigma_noise^2))*OP.BinSize;
135 135 sig_noise_after_ctile_pdf= conv_fft(sig_after_tx_ctile_pdf,noise_after_ctile_pdf);
136 136 sig_noise_after_ctile_cdf = cumsum(sig_noise_after_ctile_pdf.y);
137 137 % @ -225,7 +254,7 @ else % find noise for item that set have tx ffe for each loop
138 138 result.S_qn_rms = 0;
139 139 % result.S_n
140 140 end
141 141 result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn;
142 142 + result.S_n=result.S_rn+ result.S_tn+ result.S_xn+ result.S_jn+ result.S_qn+result.S_in;
143 143 result.S_n_rms = sqrt(sum(result.S_n)* delta_f);
144 144 %
145 145 % @ -265,7 +294,7 @ else % find noise for item that set have tx ffe for each loop
146 146 result.S_isi=sigma_X2*(abs(fft(hisi))).^2/param.fb;
147 147 result.S_isi_rms = sqrt(sum(result.S_isi)* delta_f);
148 148 %
149 149 result.S_6=result.S_tn+ result.S_rj_jn + result.S_rn; % eq 178A-30
150 150 result.S_6=result.S_tn+ result.S_rj_jn + result.S_rn+ result.S_in; % eq 178A-30
151 151 result.S_6_rms = sqrt(sum(result.S_6)* delta_f);
152 152 result.Sn_rho=result.S_isi +result.S_n; % need to include xtalk and isi
153 153 result.Sn_rho_rms = sqrt(sum(result.Sn_rho)* delta_f);
154 154 ...
```

com_ieee8023_

- ❑ 39 changed files
 - [Review in GIT](#)
- ❑ Code complete and initially test
- ❑ Configuration sheet need contributions.
- ❑ An example provided
- ❑ New keywords and interactive dialog and command line syntax

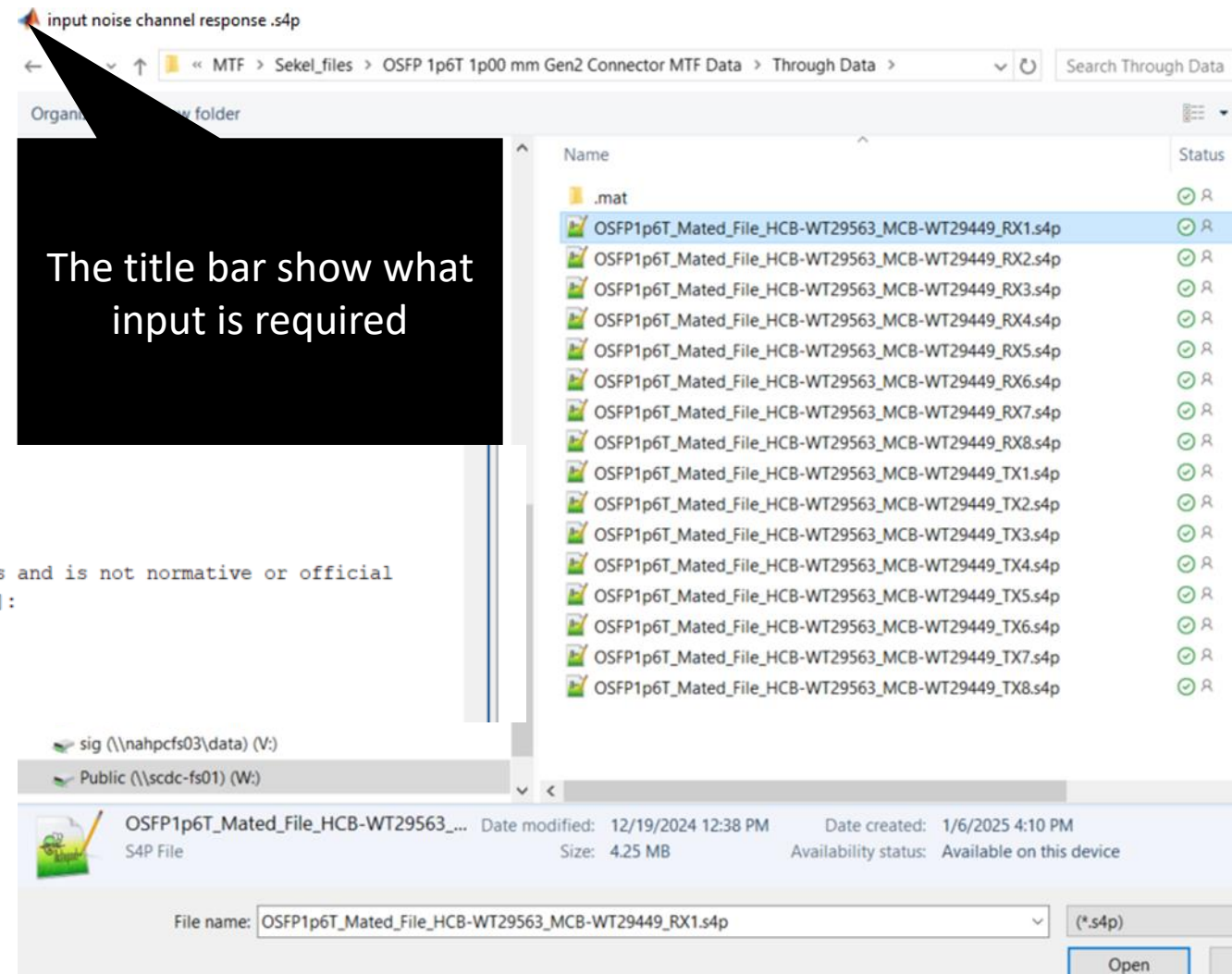
CONTRIBUTING.md	+1 -1
config_templates/C2M/200G/README.md	+0 -0
config_templates/C2M/200G/config_com_3dj_200...	+0 -0
config_templates/CR/200G/CA/README.md	+0 -0
config_templates/CR/200G/CR_Host/README.md	+0 -0
config_templates/CR/200G/CR_Module/README.md	+0 -0
config_templates/KR/200G/README.md	+0 -0
config_templates/KR/200G/config_com_dj_200G...	+0 -0
config_templates/KR/200G/config_com_dj_200G...	+0 -0
config_templates/README.md	+0 -0
config_templates/RiToL_Calitbration/200G/178/REA...	+2 -0
config_templates/RiToL_Calitbration/200G/178/con...	+0 -0
config_templates/RiToL_Calitbration/200G/178/con...	+0 -0
config_templates/RiToL_Calitbration/200G/178/tem...	+0 -0
config_templates/RiToL_Calitbration/200G/179/HH/...	+2 -0
config_templates/RiToL_Calitbration/200G/179/HL/...	+2 -0
config_templates/RiToL_Calitbration/200G/179/HN/...	+2 -0
config_templates/RiToL_Calitbration/README.md	+3 -0
release/com_ieee8023_490beta1.m	+0 -14482
release/com_ieee8023_4p9p0_beta_SNDR_Re...	+0 -10691

release/com_ieee8023_4p9p0_beta_SNR_MD...	+0 -10663
src/+Management/make_release.m	+2 -1
src/COM_FD_to_TD.m	+10 -1
src/Create_Noise_PDF.m	+6 -1
src/FD_Processing.m	+5 -0
src/H_interp.m	+51 -0
src/N_s.m	+69 -0
src/Output_Arg_Fill.m	+1 -1
src/S_IN.m	+63 -0
src/S_RN.m	+3 -3
src/add_brd.m	+1 -1
src/com_ieee8023.m	+65 -16
src/get_PSDs.m	+36 -7
src/get_s4p_files.m	+22 -0
src/get_xtlk_noise.m	+2 -0
src/make_full_pkg.m	+4 -0
src/read_ParamConfigFile.m	+27 -1
src/s21_pkg.m	+8 -1
src/save_cmd_line.m	+4 -8

New Dialog

- ❑ Question added as to whether want to proceed with a Rx calibration.
- ❑ The file request window is the entry for the noise channel

```
>> com_ieee8023_  
This is NOT an official IEEE document.  
Revision: This is a computation example for exploring COM and ERL  
for projects like IEEE P802.3bj/b/bs/cd/ck/dj with some exploratory extensions and is not normative or official  
Enter config XLS file or return will just pop a window to ask for the XLS file]:  
How many FEXT channels are to be entered? [return means no FEXT] 1  
How many NEXT channels are to be entered? [return means no NEXT] 1  
Do you want to continue with Rx calibration (Y/N) ? [return means yes]
```



COM CLI call adds a noise path

```
com_ieee8023_('C:\Users\richardm\OneDrive - Samtec\COM\802-  
COM\Richard_Mellitz_com_code\com_code\config_templates\RiTol_Calitbration\200G\178\config_com_dj_200G_RxTol_CAKR_178_PKGB_05_23_2025.xlsx'...  
,1,1,...  
'W:\userdata\channels\IEEE802.3df\sekel_3dj_02_2503\OSFP1p6T_Mated_File_HCB-WT29563_MCB-WT29449_RX1.s4p',...  
'W:\userdata\channels\IEEE802.3df\mammenga_3dj_02_2501\NEXT\HCB_vic_TX1_HCB_aggr_RX1_NEXT.s4p', ...  
'W:\userdata\channels\IEEE802.3df\mammenga_3dj_02_2501\NEXT\HCB_vic_TX1_HCB_aggr_RX1_NEXT.s4p',...  
'W:\userdata\channels\IEEE802.3df\sekel_3dj_02_2503\OSFP1p6T_Mated_File_HCB-WT29563_MCB-WT29449_RX1.s4p');
```

↑
Noise Path

New Keywords

- ❑ RIT_REF_PTR – controls noise filter in N_in
 - [clause_179](#) (default)
 - [clause_178](#)
 - [annex_176d](#)

Output indicates what RMS voltage to set noise source too

```
WC All cases FAIL ... COM = 2.993 dB
WC All cases DER = 1.208e-04 at COM threshold
WC All cases: PASS ... ERL = 11.079 dB (12.642 dB, 11.079 dB)
  LOOP with [sigma_bn] = [0.015234] performed with COM = 2.9929
  Set Tx calibration noise (sigma_hp) rms voltage to 15.2344 mV.
  redo string is: eval(['My_var_0 = ' getappdata(0,'cmd_str')])
```

fx >>

<

Spreadsheet snippets

Receiver testing		
RX_CALIBRATION	1	logical
Sigma BBN step	0.05	V
RIT_REF_PTR	clause_178	select
f_hp	clause_178	Hz
Parameter	clause_179	
board_tl_gamma0_a1_a2	annex_176D	1.4 db/in @ 53.125G
board_tl_tau	5.790E-03	ns/mm
board_Z_c	92.5	Ohm
z_bp (TX)	0	mm
z_bp (NEXT)	60	mm
z_bp (FEXT)	0	mm
z_bp (RX)	60	mm
C_0	[0 0]	nF
C_1	[0 0]	nF
Include PCB	0	logical

Switch for Rx calibration

Starting noise step size

If Rx FFE is used there are 3 Annex 176d
Not used if not using Rx FFE

This section needs to be filled in from
the referring section

Thank You!