

Updating COM 4.5 to align with 3dj D1.0 Annex 178A

Richard Mellitz, Samtec

May 21, 2024

Commit request 4p5_1 per Annex 178A

1. Implement function for MLSD described in Annex 178A
2. Remove access to untested MLSE code
3. And subsequent reporting requirements
4. Add new config parameters to support MLSE

```
MLSE_U1_c_178A.m
1 function [MLSE_results] = MLSE_U1_c_178A(param,b,A_s,A_ni,PDF,CDF,PSD_results)
2 if 1
3     num_ui=param.num_ui_RXFF_noise;
4     M=param.samples_per_ui;
5     L=param.levels;
6     sigma_X2=(L^2-1)/(3*(L-1)^2);
7     f_b=param.fb;
8 end
9 COM_from_matlab=20*log10(A_s/A_ni);
10 DER_DFE= 2*(L-1)/L*CDF_ev(A_s,PDF, CDF);
11 S_ni=PSD_results.Sn_rho;
12 R_ni=ifft(S_ni)*f_b;
13 p_scaled_by_b=scalePDF(PDF,b(1));
14 p_j=conv_fct(PDF,p_scaled_by_b);
15 p_scaled_by_1mb=scalePDF(PDF,1-b(1));
16 %
17 j=1; DER_MLSE=0; DER_MLSE_j= inf;
18 P_j.y=cumsum(p_j.y);
19 smallest_relative_change=.0001;
20 %% 178A-37
21 %
```

```
MLSE_U1_c_178A.m
20 %% 178A-37
21 rou=R_ni'/R_ni(1);
22 if DER_DFE <= param.DER_CDR
23 while j <= floor(num_ui/2) && DER_MLSE_j > DER_MLSE*smallest_relative_change
24     u_j=[ 1;(1-b(1))*ones(j-1,1); (-1)^(j+1)*b(1) ] ;% Eq slide (178A-38)
25     u_j(2:end-1)=-u_j(2:end-1);
26     % V_j=toeplitz(R_ni(1:j+1).' /R_ni(1));
27     V_j=toeplitz(rou(1:j+1));
28     P_j=cumsum(p_j.y);
29     DER_MLSE_j= ((L-1)/L)^(j-1) * ( CDF_ev( A_s *(u_j.'* u_j )^(3/2)/( u_j.*V_j*u_j)^(1/2), p_j, p_j ) ) ; % CDF_ev is (1-CDF)
30     % DER_MLSE_j= 2*(3/4)^(j) * ( CDF_ev( A_s *(u_j.'* u_j )^(3/2)/( u_j.*V_j*u_j)^(0.5), p_j, p_j ) ) ; % CDF_ev is (1-CDF)
31     DER_MLSE=DER_MLSE+DER_MLSE_j;
32     p_j=conv_fct(p_j,p_scaled_by_1mb);
33     j=j+1;
34 end
35 %% Eq (178A-36) a
36 delta_com=20*log10(1/A_s *-CDF_inv_ev ( DER_MLSE,PDF,CDF ) )- param.Q ;% shakiba_3dj_01_2405
37 % delta_com=20*log10(1/A_s *-CDF_inv_ev ( 2/3*DER_MLSE,PDF,CDF ) )- param.Q ;% (178A-36)
38 new_com=COM_from_matlab+delta_com;
39 else
40     warning('MLSE not applied because the DER is less than that required for the CDR to lock')
41     DER_MLSE=NaN;
42     new_com=COM_from_matlab;
43     delta_com=0;
44     delta_com=0;
45 end
46 %
47 %%
48 MLSE_results.COM_from_matlab=COM_from_matlab;
49 MLSE_results.DER_MLSE=DER_MLSE;
50 MLSE_results.DER_DFE=DER_DFE;
51 MLSE_results.COM=new_com;
52 MLSE_results.delta_com=delta_com;
```

Commit request 4p5_1 per Annex 178A, part 2

REMOVE ACCESS TO UNTESTED MLSE CODE

```
com_ieee8023_93a_460beta1.m
564     VEC_dB = -20*log10(vec_arg);
565     COM=20*log10(A_s/A_ni);
566     min_COM = min(min_COM, COM);
567     min_VEO_mV = min(min_VEO_mV,VEO_mV);
568     max_VEC_dB = max(max_VEC_dB, VEC_dB);
569
570     end
571
572     MLSE_results=struct;
573
574
575
576
577
578
579
580
581
582
583
com_ieee8023_93a_450.m
563     VEC_dB = -20*log10(vec_arg);
564     COM=20*log10(A_s/A_ni);
565     min_COM = min(min_COM, COM);
566     min_VEO_mV = min(min_VEO_mV,VEO_mV);
567     max_VEC_dB = max(max_VEC_dB, VEC_dB);
568
569     end
570
571     MLSE_results=struct;
572
573 % MLSE case added U3 option
574
575
576
577
578 warning('unsupported MLSE option')
579
580
581
582
583
```

Commit request 4p5_1 per Annex 178A, part 3

CHANGE VARIABLE NAMES IN PREPARATION FOR REPORTING

com_ieee8023_93a_460beta1.m	com_ieee8023_93a_450.m
584 COM=MLSE_results.COM;	591 COM=MLSE_results.COM_CDF;
585 VEO_mV=eye_opening*1000;	592 VEO_mV=eye_opening*1000;
586 min_COM = min(min_COM, COM);	593 min_COM = min(min_COM, COM);
587 min_VEO_mV = min(min_VEO_mV, VEO_mV);	594 min_VEO_mV = min(min_VEO_mV, VEO_mV);
588 max_VEC_db = max(max_VEC_db, VEC_db);	595 max_VEC_db = max(max_VEC_db, VEC_db);
589 COM_SNR_Struct.delta_COM=MLSE_results.delta_com;	596 output_args.delta_COM=MLSE_results.delta_com_CDF;
590 COM_SNR_Struct.DER_DFE=MLSE_results.DER_DFE;	
591 COM_SNR_Struct.DER_MLSE=MLSE_results.DER_MLSE;	
592 COM_SNR_Struct.delta_VEC=MLSE_results.delta_com-20*log10((10^(MLSE_r	597 output_args.delta_VEC=MLSE_results.delta_com_CDF-20*log10((10^(MLSE_r
593 COM_SNR_Struct.VEC_db_orig=VEC_db_orig;	
594 COM_SNR_Struct.VEC_db=VEC_db;	
595 else	598 else
596 VEO_mV = 1000*(A_s-A_ni)*2;	599 VEO_mV = 1000*(A_s-A_ni)*2;
597 vec_arg=(A_s-A_ni)/A_s;	600 vec_arg=(A_s-A_ni)/A_s;
598 if vec_arg<eps	601 if vec_arg<eps
599 vec_arg=eps;	602 vec_arg=eps;
600 end	603 end
601 VEC_db_orig = -20*log10(vec_arg);	604 VEC_db_orig = -20*log10(vec_arg);
602 VEC_db=MLSE_results.delta_com-20*log10((10^(MLSE_results.delta_com/2	605 VEC_db=MLSE_results.delta_com_CDF-20*log10((10^(MLSE_results.delta_com/2
603 COM_orig=20*log10(A_s/A_ni);	606 COM_orig=20*log10(A_s/A_ni);
604 COM=MLSE_results.COM;	607 COM=MLSE_results.COM_CDF;

Commit request 4p5_1 per Annex 178A, part 3

COM_SNR_STRUCTURE TO INCLUDE MLSE RESULTS

m_ieee8023_93a_460beta1.m	com_ieee8023_93a_450.m
581 VEC_db_orig = 20*log10(vec_arg); % was negative in 400 beta1 ... Fixed 2- 582 VEC_db=MLSE_results.delta_com-20*log10((10^(MLSE_results.delta_com/20)-1 583 COM_orig=20*log10(2*A_s/A_ni); 584 COM=MLSE_results.COM; 585 VEO_mV=eye_opening*1000; 586 min_COM = min(min_COM, COM); 587 min_VEO_mV = min(min_VEO_mV, VEO_mV); 588 max_VEC_db = max(max_VEC_db, VEC_db); 589 COM_SNR_Struct.delta_COM=MLSE_results.delta_com; # 590 COM_SNR_Struct.DER_DFE=MLSE_results.DER_DFE; 591 COM_SNR_Struct.DER_MLSE=MLSE_results.DER_MLSE; 592 COM_SNR_Struct.delta_VEC=MLSE_results.delta_com-20*log10((10^(MLSE_results. 593 COM_SNR_Struct.VEC_db_orig=VEC_db_orig; 594 COM_SNR_Struct.VEC_db=VEC_db; 595 else 596 VEO_mV = 1000*(A_s-A_ni)*2; 597 vec_arg=(A_s-A_ni)/A_s; 598 if vec_arg<eps 599 vec_arg=eps; 600 end 601 VEC_db_orig = -20*log10(vec_arg); 602 VEC_db=MLSE_results.delta_com-20*log10((10^(MLSE_results.delta_com/20)-1 603 COM_orig=20*log10(A_s/A_ni); 604 COM=MLSE_results.COM; 605 min_COM = min(min_COM, COM); 606 min_VEO_mV = min(min_VEO_mV, VEO_mV); 607 max_VEC_db = max(max_VEC_db, VEC_db); 608 COM_SNR_Struct.delta_COM=MLSE_results.delta_com; # 609 COM_SNR_Struct.DER_DFE=MLSE_results.DER_DFE;	588 VEC_db_orig = 20*log10(vec_arg); % was negative in 400 beta1 ... Fixed 589 VEC_db=MLSE_results.delta_com_CDF-20*log10((10^(MLSE_results.delta_com_CDF/20)-1 590 COM_orig=20*log10(2*A_s/A_ni); 591 COM=MLSE_results.COM_CDF; 592 VEO_mV=eye_opening*1000; 593 min_COM = min(min_COM, COM); 594 min_VEO_mV = min(min_VEO_mV, VEO_mV); 595 max_VEC_db = max(max_VEC_db, VEC_db); 596 output_args.delta_COM=MLSE_results.delta_com_CDF; 597 output_args.delta_VEC=MLSE_results.delta_com_CDF-20*log10((10^(MLSE_results. 598 else 599 VEO_mV = 1000*(A_s-A_ni)*2; 600 vec_arg=(A_s-A_ni)/A_s; 601 if vec_arg<eps 602 vec_arg=eps; 603 end 604 VEC_db_orig = -20*log10(vec_arg); 605 VEC_db=MLSE_results.delta_com_CDF-20*log10((10^(MLSE_results.delta_com_CDF/20)-1 606 COM_orig=20*log10(A_s/A_ni); 607 COM=MLSE_results.COM_CDF; 608 min_COM = min(min_COM, COM); 609 min_VEO_mV = min(min_VEO_mV, VEO_mV); 610 max_VEC_db = max(max_VEC_db, VEC_db); 611 output_args.delta_COM=MLSE_results.delta_com_CDF;

Commit request 4p5_1 per Annex 178A, part 3

add new items in COM_SNR_struct to output_args

com_ieee8023_93a_460beta1.m		com_ieee8023_93a_450.m
2824	if OP.MLSE	2768 if OP.MLSE
2825	output_args.COM_orig=COM_SNR_Struct.COM_orig;	2769 output_args.COM_orig=COM_SNR_Struct.COM_orig;
2826	output_args.delta_COM = COM_SNR_Struct.delta_COM;	
2827	output_args.DER_DFE= COM_SNR_Struct.DER_DFE;	
2828	output_args.DER_MLSE= COM_SNR_Struct.DER_MLSE;	
2829	if strcmpi(upper(OP.PHY), 'C2M')	
2830	output_args.VEC_dB_orig= COM_SNR_Struct.VEC_dB_orig;	2770 output_args.VEC_dB_orig=COM_SNR_Struct.VEC_dB_orig;
2831	output_args.delta_VEC = COM_SNR_Struct.delta_VEC;	
2832	output_args.VEC_dB_orig = COM_SNR_Struct.VEC_dB_orig;	
2833	output_args.VEC_dB= COM_SNR_Struct.VEC_dB;	
2834	end	
2835	end	2771 end

Commit request 4p5_1 per Annex 178A, part 3

Add 2 new configuration parameters for MLSE

- DER_CDR (default: 1e-2) maximum DER_DFE that MLSE will be evaluated
 - This replaces the criterial that COM must be greater than zero for MLSE to be evaluated
- Q (default = 0) MLSE implementation penalty in Anne 178A

com_ieee8023_93a_460beta1.m		com_ieee8023_93a_450.m	
9179	param.levels = xls_parameter(parameter, 'L'); % number of symbols levels (I)	9109	param.levels = xls_parameter(parameter, 'L'); % number of symbols levels (I)
9180	param.specBER = xls_parameter(parameter, 'DER_0'); % Target detector error	9110	param.specBER = xls_parameter(parameter, 'DER_0'); % Target detector error
-	9181 param.DER_CDR = xls_parameter(parameter, 'DER_CDR', true, 1e-2); % min DER re		
9182		9200	
9201		9201	
9202		9202	
9203		9203	
9204		9204	
9205		9205	
9206		9206	
9207		9207	
9208		9208	
9209		9209	
9210		9210	
9211		9211	
9212		9212	
9213		9213	
9214		9214	
9215		9215	
9216		9216	
9217		9217	
9218		9218	
9219		9219	
9220		9220	
9221		9221	
9222		9222	
9223		9223	
9224		9224	
9225		9225	
9226		9226	
9227		9227	
9228		9228	
9229		9229	
9230		9230	
9231		9231	
9232		9232	
9233		9233	
9234		9234	
9235		9235	
9236		9236	
9237		9237	
9238		9238	
9239		9239	
9240		9240	
9241		9241	
9242		9242	
9243		9243	
9244		9244	
9245		9245	
9246		9246	
9247		9247	
9248		9248	
9249		9249	
9250		9250	
9251		9251	
9252		9252	
9253		9253	
9254		9254	
9255		9255	
9256		9256	
9257		9257	
9258		9258	
9259		9259	
9260		9260	
9261		9261	
9262		9262	
9263		9263	
9264		9264	
9265		9265	
9266		9266	
9267		9267	
9268		9268	
9269		9269	
9270		9270	
9271		9271	
9272		9272	
9273		9273	
9274		9274	
9275		9275	
9276		9276	
9277		9277	
9278		9278	
9279		9279	
9280		9280	
9281		9281	
9282		9282	
9283		9283	
9284		9284	
9285		9285	
9286		9286	
9287		9287	
9288		9288	
9289		9289	
9290		9290	
9291		9291	
9292		9292	
9293		9293	
9294		9294	
9295		9295	
9296		9296	
9297		9297	
9298		9298	
9299		9299	
9300		9300	
9301		9301	
9302		9302	
9303		9303	
9304		9304	
9305		9305	
9306		9306	
9307		9307	
9308		9308	
9309		9309	
9310		9310	
9311		9311	
9312		9312	
9313		9313	
9314		9314	
9315		9315	
9316		9316	
9317		9317	
9318		9318	
9319		9319	
9320		9320	
9321		9321	
9322		9322	
9323		9323	
9324		9324	
9325		9325	
9326		9326	
9327		9327	
9328		9328	
9329		9329	
9330		9330	
9331		9331	
9332		9332	
9333		9333	
9334		9334	
9335		9335	
9336		9336	
9337		9337	
9338		9338	
9339		9339	
9340		9340	
9341		9341	
9342		9342	
9343		9343	
9344		9344	
9345		9345	
9346		9346	
9347		9347	
9348		9348	
9349		9349	
9350		9350	
9351		9351	
9352		9352	
9353		9353	
9354		9354	
9355		9355	
9356		9356	
9357		9357	
9358		9358	
9359		9359	
9360		9360	
9361		9361	
9362		9362	
9363		9363	
9364		9364	
9365		9365	
9366		9366	
9367		9367	
9368		9368	
9369		9369	
9370		9370	
9371		9371	
9372		9372	
9373		9373	
9374		9374	
9375		9375	
9376		9376	
9377		9377	
9378		9378	
9379		9379	
9380		9380	
9381		9381	
9382		9382	
9383		9383	
9384		9384	
9385		9385	
9386		9386	
9387		9387	
9388		9388	
9389		9389	
9390		9390	
9391		9391	
9392		9392	
9393		9393	
9394		9394	
9395		9395	
9396		9396	
9397		9397	
9398		9398	
9399		9399	
9400		9400	
9401		9401	
9402		9402	
9403		9403	
9404		9404	
9405		9405	
9406		9406	
9407		9407	
9408		9408	
9409		9409	
9410		9410	
9411		9411	
9412		9412	
9413		9413	
9414		9414	
9415		9415	
9416		9416	
9417		9417	
9418		9418	
9419		9419	
9420		9420	
9421		9421	
9422		9422	
9423		9423	
9424		9424	
9425		9425	
9426		9426	
9427		9427	
9428		9428	
9429		9429	
9430		9430	
9431		9431	
9432		9432	
9433		9433	
9434		9434	
9435		9435	
9436		9436	
9437		9437	
9438		9438	
9439		9439	
9440		9440	
9441		9441	
9442		9442	
9443		9443	
9444		9444	
9445		9445	
9446		9446	
9447		9447	
9448		9448	
9449		9449	
9450		9450	
9451		9451	
9452		9452	
9453		9453	
9454		9454	
9455		9455	
9456		9456	
9457		9457	
9458		9458	
9459		9459	
9460		9460	
9461		9461	
9462		9462	
9463		9463	
9464		9464	
9465		9465	
9466		9466	
9467		9467	
9468		9468	
9469		9469	
9470		9470	
9471		9471	
9472		9472	
9473		9473	
9474		9474	
9475		9475	
9476		9476	
9477		9477	
9478		9478	
9479		9479	
9480		9480	
9481		9481	
9482		9482	
9483		9483	
9484		9484	
9485		9485	
9486		9486	
9487		9487	
9488		9488	
9489		9489	
9490		9490	
9491		9491	
9492		9492	
9493		9493	
9494		9494	
9495		9495	
9496		9496	
9497		9497	
9498		9498	
9499		9499	
9500		9500	
9501		9501	
9502		9502	
9503		9503	
9504		9504	
9505		9505	
9506		9506	
9507		9507	
9508		9508	
9509		9509	
9510		9510	
9511		9511	
9512		9512	
9513		9513	
9514		9514	
9515		9515	
9516		9516	
9517		9517	
9518		9518	
9519		9519	
9520		9520	
9521		9521	
9522		9522	
9523		9523	
9524		9524	
9525		9525	
9526		9526	
9527		9527	
9528		9528	
9529		9529	
9530		9530	
9531		9531	
9532		9532	
9533		9533	
9534		9534	
9535		9535	
9536		9536	
9537		9537	
9538		9538	
9539		9539	
9540		9540	
9541		9541	
9542		9542	
9543		9543	
9544		9544	
9545		9545	
9546		9546	
9547		9547	
9548		9548	
9549		9549	
9550		9550	
9551		9551	
9552		9552	

Thank You!