

COM: how open-source ports match the Reference Code

The IEEE 802.3 COM Reference Code, a Python port and an Octave port over the same 208 cases

Todd Bermensolo, Independent - author and presenter

Hansel D'Silva, Amphenol Corporation - co-author, Reference Code runs, Adaptive Local Search

COM Octave support: Rich Mellitz, Adam Gregory | Review and direction: Howard Heck

IEEE 802.3 COM ad hoc

29 September 2026

Not an official IEEE document. The Reference Code release used here is `com_ieee8023_4p15p0_adaptive_local_search.m`. All code BSD-3-Clause. The ports are not IEEE 802.3 collateral.

The Reference Code

One standard, and everything in this deck is measured against it

term	what it means
Reference Code	the COM MATLAB code reviewed by the ad hoc and released on the IEEE COM Git site
release used here	4.15 with the adaptive local search: the build the 208-case reference results were produced with. 4.16 adopted that search into the mainline, and on this corpus the two are bit-identical
reference results	the COM, FOM, sampling phase and selected equalizer that build reports for each of the 208 cases
what "agrees" means	a port agrees when it reports the Reference Code numbers for the same inputs. The Reference Code is not judged against the ports
what the ports are	an independent re-implementation (Python) and a compatibility build (Octave) of that same release. Checks on it, not replacements for it, and not IEEE collateral

If the Reference Code changes, the reference results change with it and every port re-correlates. That is the loop slides 14 to 17 propose to make routine.

Objective and method

One definition of a case, three implementations, measured the same way

Question: do independent open-source implementations of COM reproduce the Reference Code's result, and what does it take to run them?

Method: the existing 208-case comparison of the Python port against the Reference Code was extended with an Octave arm. All three arms are driven from ONE manifest, so channel resolution, aggressor ordering, package configuration and the crosstalk mechanism are identical by construction rather than by re-derivation.

Agreement is reported on COM, and separately on the equalizer the search selected. Two implementations can agree on COM while having chosen different equalizers, so the EQ check is reported on its own.

Adaptive local search is active in all eight configurations (Local Search = 2, Non-zero Local Search Method = 1) and therefore in all three arms. This is an adaptive-vs-adaptive comparison.

The case set

26 public channels x 4 package configurations x 2 crosstalk conditions

	value
channels	26 IEEE 802.3dj public-area CR and KR
package configurations	4 (Test 1 and 2 PKG A, Test 3 and 4 PKG B)
crosstalk conditions	2 (with, without)
total cases	208
cases with an Octave result	208
COM range covered	-4.82 to 6.68 dB

The channels are from the IEEE 802.3dj public area, so the comparison is reproducible by anyone with the same configuration workbooks. Without crosstalk means the same THRU channel with no aggressor files and both counts zero, which is how the reference results were produced.

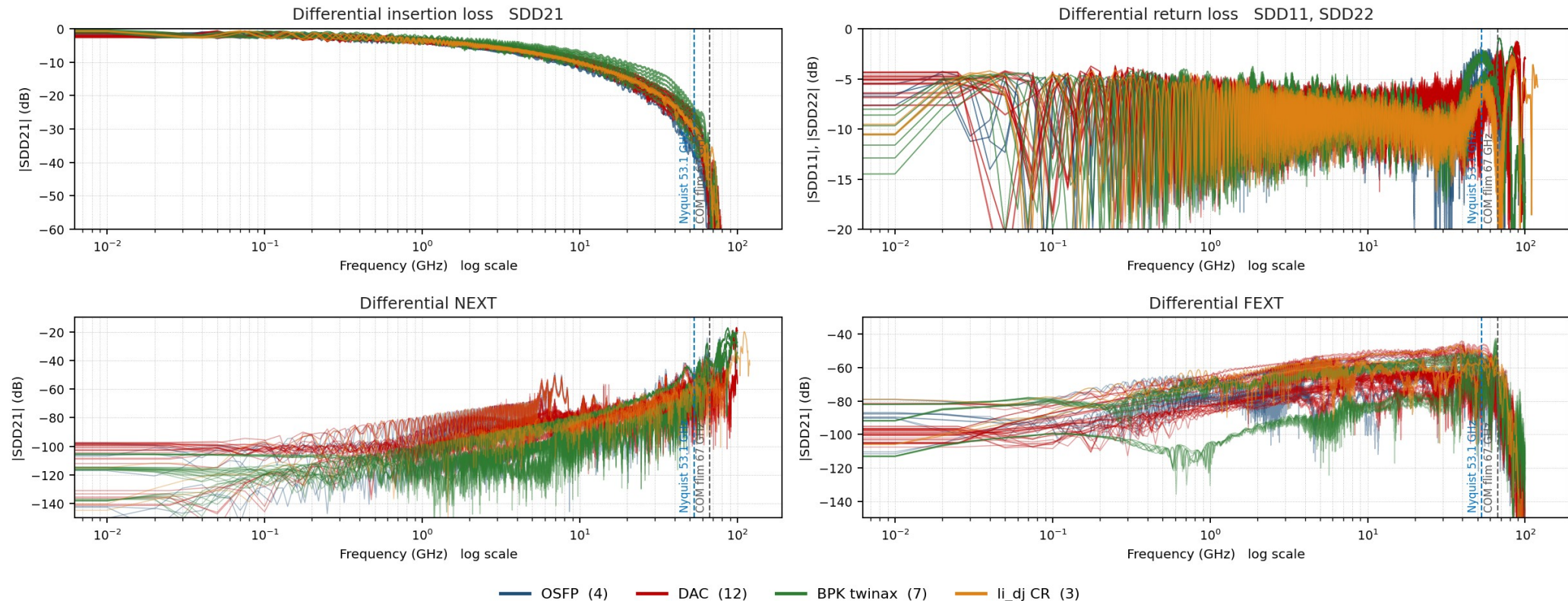
Channels are evaluated against the class A and class B package models as part of KR. The configuration workbooks are in backup, slides 22 and 23.

This set is proposed as the standing benchmark. Slide 14.

The channels behind those 208 cases

Frequency-domain behaviour of all 26 victim channels and their crosstalk aggressors, vendor channels only, no package networks

Channels in the correlation set 26 thru, 138 crosstalk aggressors, to 120 GHz



Log frequency axis over the full range the S-parameter files carry. COM truncates at flim = 67 GHz (dashed); Nyquist for 106.25 GBd PAM4 is 53.1 GHz. Four channel families: OSFP cable assemblies, DACs, backplane twinax, and CR designs.

Version provenance

What each arm actually ran

	Reference Code (MATLAB 4.15, adaptive LS)	Python port	Octave port
source	com_ieee8023_4p15p0_adaptive_local_search.m	emulates 4p15p0	Octave_compat branch, src/
runtime	MATLAB, machine unknown	Python 3.14, NumPy 2.4	GNU Octave 11.3.0
adaptive local search	active	active	active
config input	xlsx	xlsx	.mat (xlsread unavailable)
overrides applied	none	none	conv_fct_noPDFx (restore p.x) read_Nport_touchstone (filter NaN)

The 4.15-to-4.16 version question was settled by measurement, not assumed: the Python port was run over all 208 cases under both release emulations and the results are bit-identical, with the same COM, the same sampling phase and the same Tx FFE winner on every case. On this corpus no code path that differs between the releases moves a number.

The Octave arm as run for this comparison is the compatibility branch. Since then it has been rebuilt from the Reference Code release file itself: slide 21.

Comparison metrics

COM, figure of merit, sampling phase and the selected equalizer, against the Reference Code

metric	Python port	Octave port
COM tolerance class: EXACT	208 / 208	208 / 208
COM, max abs deviation (dB)	3.3e-14	5.1e-14
COM, median abs deviation (dB)	7.1e-15	9.8e-15
FOM, max abs deviation (dB)	3.4e-11	3.5e-11
FOM, median abs deviation (dB)	5.4e-12	5.6e-12
sampling phase (itick) exact	208 / 208	208 / 208
selected equalizer matches	208 / 208	208 / 208
EQ_DIFF flagged	0	0
pass/fail at 3 dB agrees	208 / 208	208 / 208

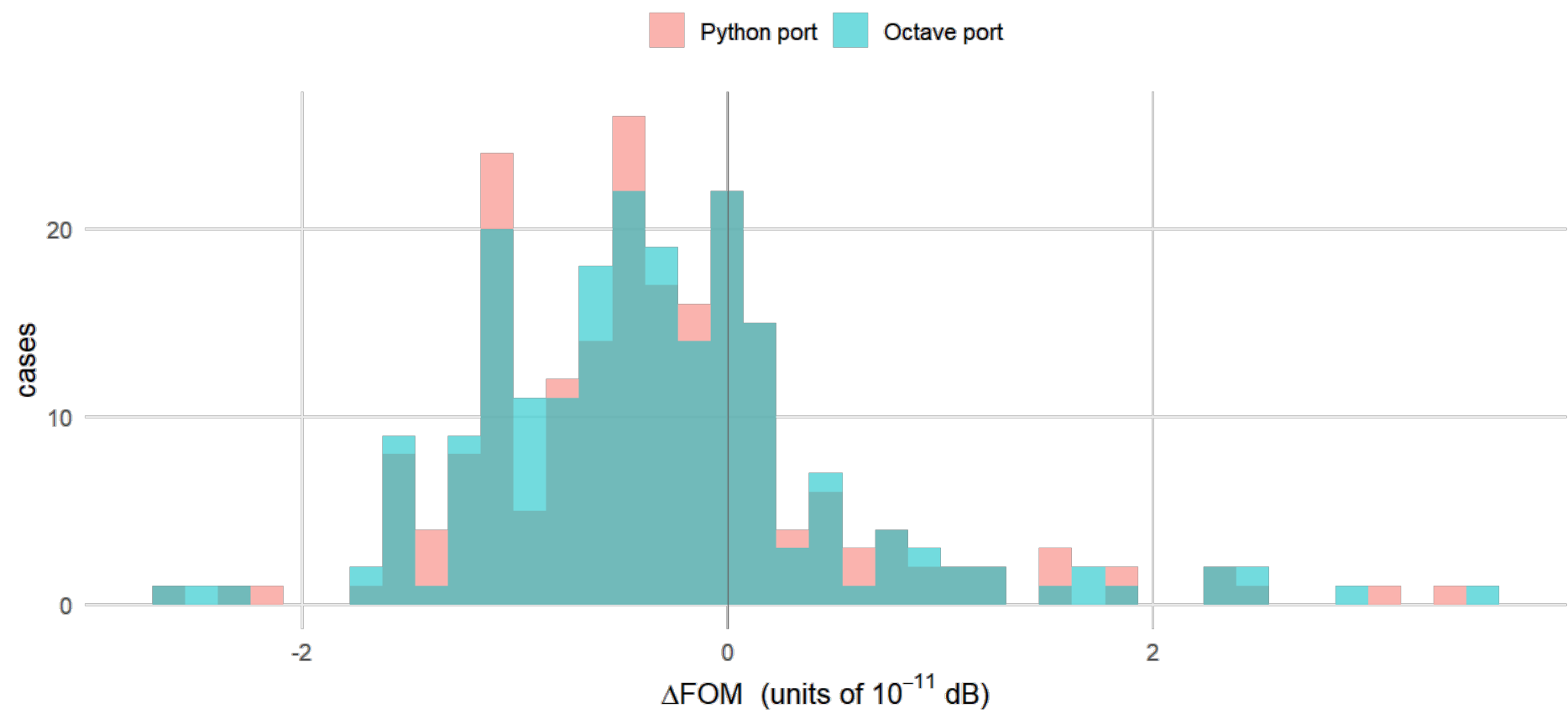
EXACT means within 1e-6 dB. The observed agreement is 1e-14 on COM and 1e-11 on FOM, which is double-precision arithmetic noise rather than agreement to a tolerance. It holds across a COM range of -4.82 to 6.68 dB, and 123 of the 208 reference cases fail the 3 dB threshold, so this is not agreement concentrated where it is easy. The equalizer check compares Tx FFE taps, CTLE poles and zero, CTLE DC gain, DFE taps and floating-tap locations per case.

Figure of merit: deviation distribution

FOM is what the equalizer search maximises, so it is the more sensitive test

Figure of merit: deviation from the MATLAB reference

208 cases $\max |dFOM| = 3.5e-11$ dB across both ports
FOM is the quantity the equalizer search maximises, so it is the more sensitive test.

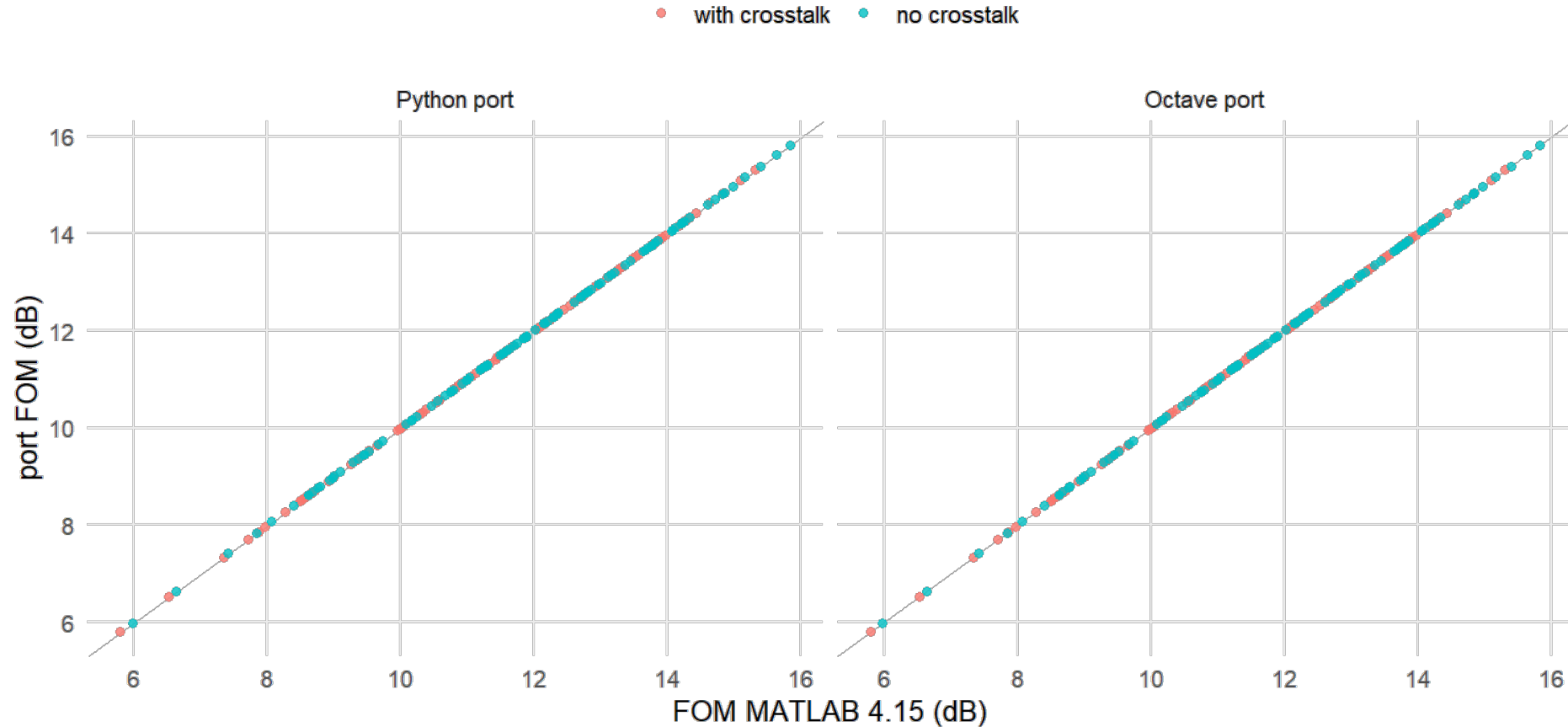


FOM correlation against the Reference Code

Each port plotted against the reference FOM

Figure of merit against the MATLAB reference

208 cases $\max |dFOM| = 3.5e-11$ dB across both ports

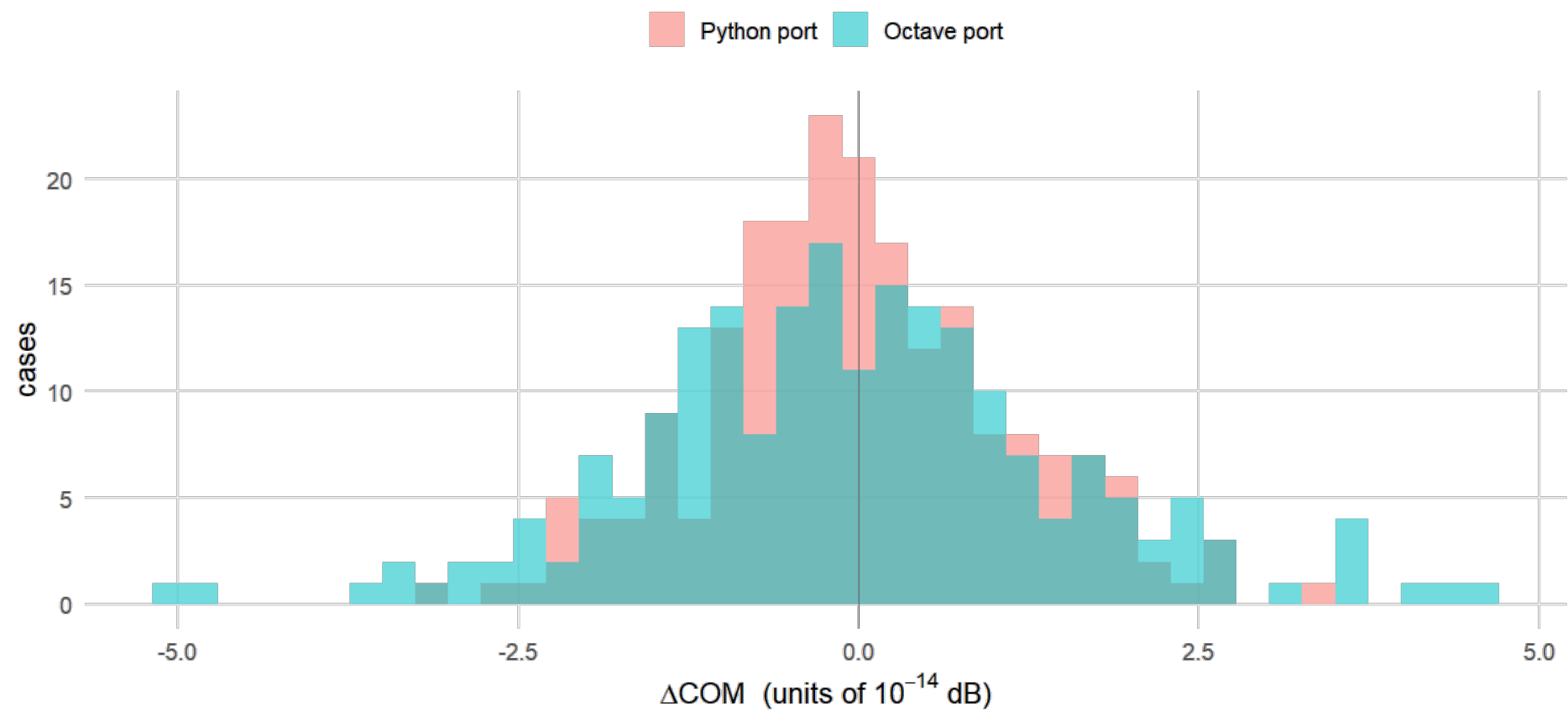


COM: deviation distribution, both ports on one axis

Nothing is hidden by a log scale or a clipped range

Deviation from the MATLAB reference, both ports on one axis

208 cases max |dCOM| = 5.1e-14 dB across both ports every case EXACT ($\leq 1\text{e-}6$ dB)
This is double-precision arithmetic noise, not agreement to a tolerance.

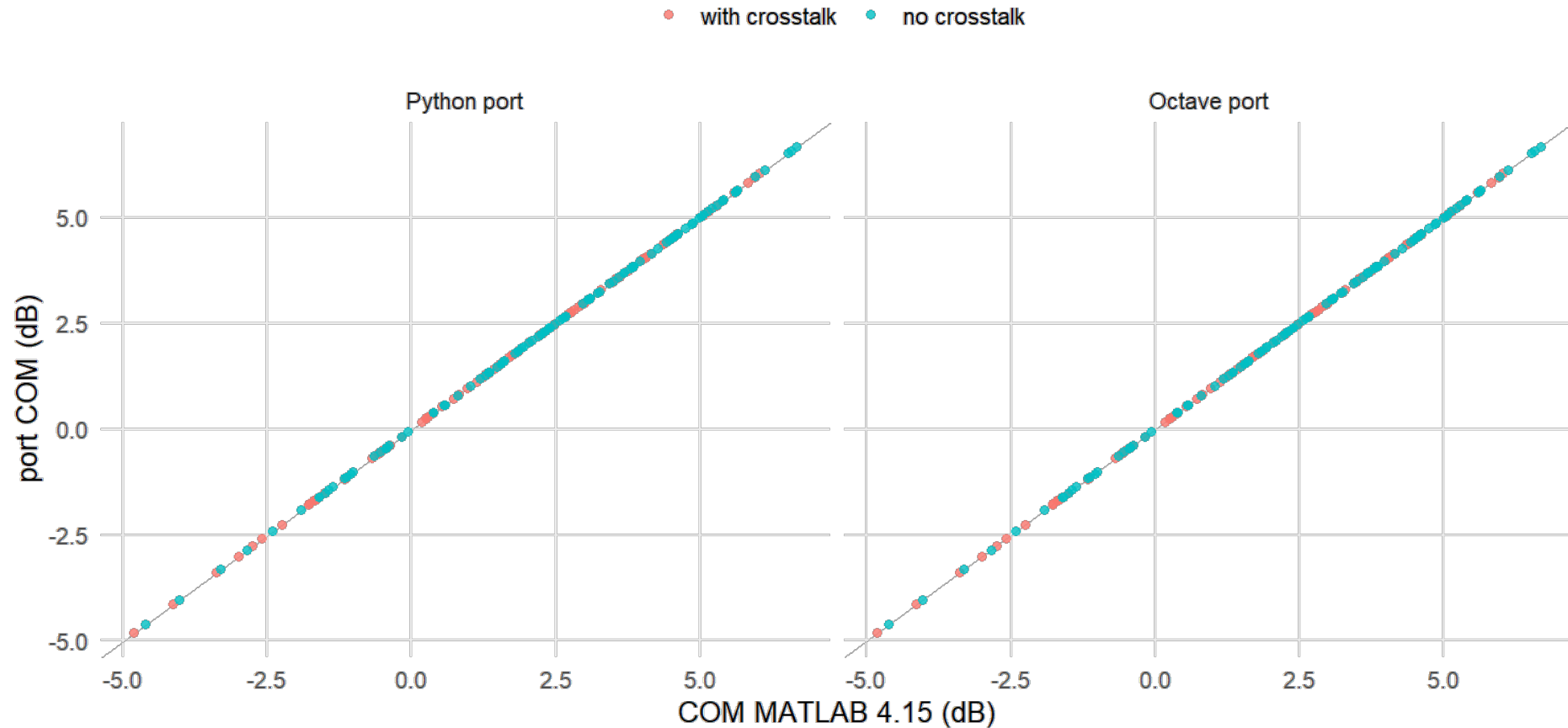


COM correlation against the Reference Code

Each port plotted against the reference COM

COM: open-source ports against the MATLAB reference

208 cases $\max |dCOM| = 5.1e-14$ dB across both ports every case EXACT ($\leq 1e-6$ dB)

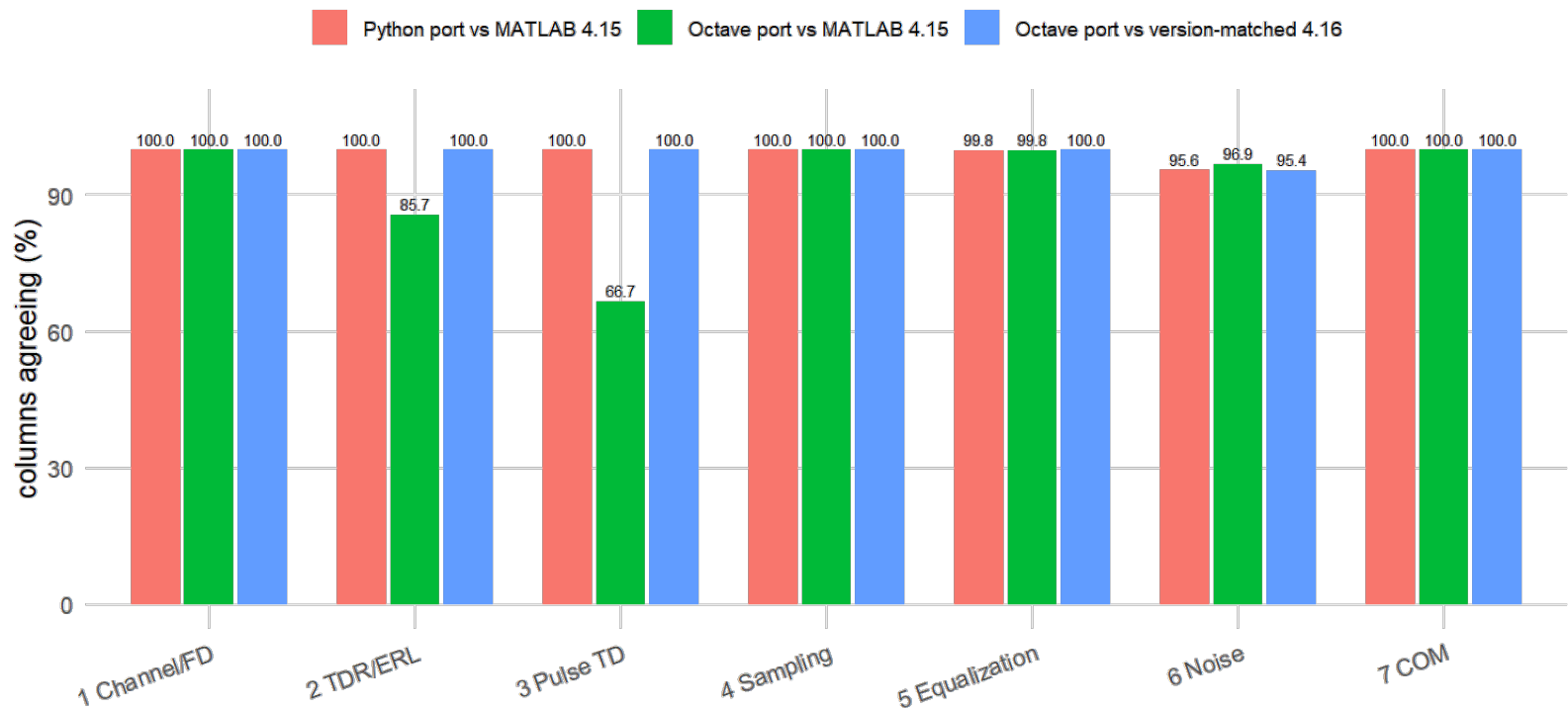


Agreement through the pipeline

Where in the COM flow each port agrees, not just at the output

Agreement through the pipeline

Share of reported output columns agreeing to better than 1e-9 relative, over all 208 cases.
The Octave arm runs COM 4.16 while the workbook reference is 4.15. Matching the reference to the version resolves three stages.



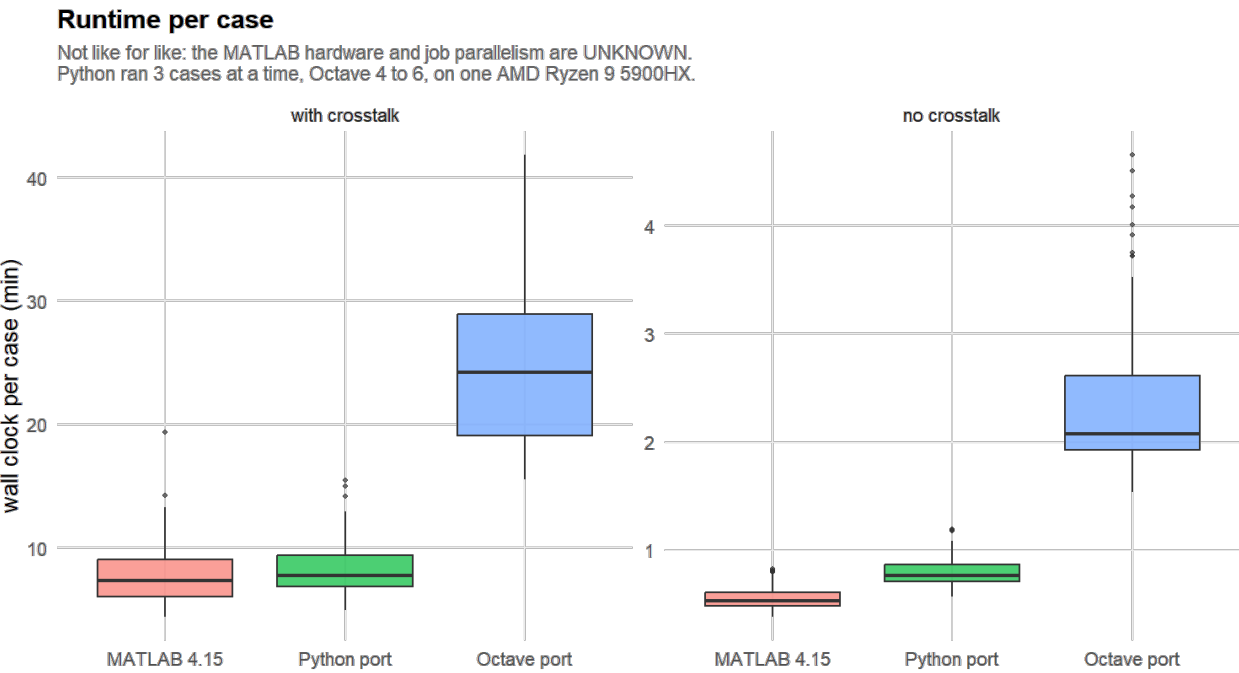
Diagnosis, and the recommendation. The two short bars are ONE column each, steady_state_voltage_mV and peak_uneq_pulse_mV, differing by exactly the drive amplitude A_v. They are not a port defect: COM 4.16 extends the USE_channel_amplitude block to scale the pulse and step responses and not just the impulse, correcting an inconsistency in 4.15. The Octave arm runs 4.16 while the reference results are 4.15. Comparing against a version-matched 4.16 reference takes those stages, and Equalization, to 100 percent. RECOMMENDATION: version-match the reference before comparing intermediate columns, or the method reports a real version change as a port defect.

Runtime

A data point, not a benchmark

condition	Reference Code	Python port	Octave port
without crosstalk, median per case	0.53 min	0.76 min	2.07 min
with crosstalk, median per case	7.38 min	7.75 min	24.21 min

The Reference Code hardware and job parallelism are UNKNOWN, so that column is context, not a comparison. Python and Octave ran on the same machine, an AMD Ryzen 9 5900HX with 8 cores and 16 threads and 15.4 GB, Python three cases at a time and Octave four to six. Self-contention costs about 1.7x, so solo per-case times would be lower. Only the Python-to-Octave comparison is like for like. Octave is loop-bound rather than matrix-bound, and light on memory at about 0.6 GB per case with crosstalk.



A standing benchmark: these 208 cases

What a benchmark case is, and why these fit

a benchmark case is	what that means
a Reference Code release	one named release file, here com_ieee8023_4p15p0_adaptive_local_search.m
a configuration workbook	one named workbook per case. The eight used here carry a BSD-3-Clause License Notice sheet of their own, so they could travel with the benchmark
a channel set	THRU plus aggressors, each named by its path in the IEEE 802.3dj public tools area. Channels are not redistributed, they are re-downloaded
the reference results	COM, FOM, sampling phase, Tx FFE taps, CTLE, DFE taps, from that release on those inputs

Why these 208:

- Public channels, four package configurations, with and without crosstalk, adaptive local search active in every arm. Already run three ways, with COM from -4.82 to 6.68 dB and 123 of 208 cases failing the 3 dB threshold.
- Reproducible by anyone holding the workbooks and the public channels. No private input is needed to run it.
- Large enough that a port cannot pass by accident, small enough to run overnight on one desktop machine.

Run privately here today. Which reference results travel with it, and how, is the choice put to COM leadership on slide 17.

The correlation standard, as a precedent

What "correlates" should mean each time we do this again

quantity	expectation	why
COM	EXACT, within 1e-6 dB, every case	both ports sit at 1e-14; 1e-6 leaves room for a different BLAS without hiding a real defect
sampling phase	identical, every case	a wrong phase can still give a close COM
selected equalizer	identical taps, poles, gains and floating-tap locations	two implementations can agree on COM having chosen different equalizers
pass / fail	agrees, every case	the quantity the standard actually uses
FOM	reported, expected within 1e-9 dB	the search maximises it, so it is the more sensitive test
anything outside EXACT	attributed, not tolerated	each deviation assigned a cause: version, search, shim, config, numerical library, run failure. Unattributed must be zero

Tolerance classes as used throughout this deck: EXACT $\leq 1\text{e-}6$ dB, TIGHT ≤ 0.01 , LOOSE ≤ 0.05 , MISMATCH above. Every class is reported and none is hidden by a log axis or a clipped range. The report format is the two tables on slides 7 and 13 plus the attribution table, produced the same way every time so that releases can be set side by side.

Offered for the ad hoc to adopt or amend.

Maintenance and distribution

Where the port lives, who maintains it, and how it stays in sync

question	answer
Where does the Python port live?	SiCoPR, a Git repository, BSD-3-Clause, the same licence as the Reference Code. Public ahead of this meeting
What ships with it?	the engine, the reference MATLAB files it emulates, the Octave-capable release files and their generator, the workbook converter, the comparison driver, 890 per-function tests. No IEEE channel data
Who maintains it?	Todd Bermensolo. The commitment is deliberately narrow: consistency and correlation with the Reference Code. IEEE maintains one code base; this is an extra implementation outside it, not a second standard
How does it stay in sync?	per release: diff it, map the diff onto the functions needing re-check, rerun the per-function tests, rerun the benchmark, publish the correlation tables. The 4.15 to 4.16 step was done this way
Where do the Octave files live?	in the same repository, generated from the release file (slide 21). Whether an Octave-capable release also belongs on the IEEE site is for COM leadership, given that one code base is maintained there. Open item

What it is not: a replacement for the Reference Code, IEEE collateral, or a place to change COM. A change to COM goes to the ad hoc and the Reference Code first. The port re-correlates after.

Maintenance and distribution: Benchmark

The cases move with the code, and two choices for COM leadership

Pinned to a Reference Code release. When the release changes, rerun the existing cases on it FIRST, so the delta between releases is measured rather than assumed; 4.15 to 4.16 was done this way, bit-identical COM on all 208 cases (slide 12).

When a feature is added, 400G among them, add cases that exercise it, tagged with the release that introduced them. Old cases stay; the set only grows. The dj workbooks are close to locked with D3.3 approaching, and the pending keywords are runtime-related at under 0.05 dB: exactly what the rerun-first rule is for.

Choice 1: which reference results travel with the benchmark?

	option 1: MATLAB reference results	option 2: COM Octave results
the column is	the Reference Code's own numbers on the 208 cases, the run this deck is measured against	the same 208 cases through the Octave-capable release file. This deck shows it matches the Reference Code to 1e-14 dB on every case
who can rerun it	anyone with a MATLAB licence	anyone with GNU Octave. No licence
what it needs	release of the values by the co-author who produced them	about two days of machine time here

Choice 2: where the benchmark lives, and who owns the quality check

	option 1: the IEEE COM Git site	option 2: with the COM port, in the SiCoPR repository
the set sits	beside the Reference Code, on the site that releases it	beside the port that is measured against it
who owns the quality check	IEEE COM signs up for benchmark metric management	the COM port maintainer, as part of every release check

The set is the same in either place: the eight licensed workbooks, a manifest naming each channel by its IEEE public-area path, the reference results as one CSV, and the driver that reruns and compares.

How the Python port was built, and how it is verified

One function at a time, each checked against the Reference Code executed, not read

step	what it means
1. Split the reference	the Reference Code release file is decomposed into its 156 functions
2. Translate one function	each becomes com_functions/fn/<name>/py_impl.py, naming the MATLAB lines it mirrors
3. Test it against the reference, executed	the Reference Code function is executed under GNU Octave on the same inputs and its outputs are pinned as the expected values: 146 of 146 value-bearing functions
4. Assemble	the functions are concatenated into one file, sicopr.py; no MATLAB is needed to run it
5. Check every stage, then the result	every field of every intermediate result against the Reference Code under Octave at 10 stage boundaries (28 cases); every reported output against the MATLAB reference results (208 cases)

Running the reference is what finds the differences a line-by-line reading passes: std normalised by N rather than N-1, rounding half to even, max propagating NaN. The 208 cases were re-run after this verification: the sampling point is identical on 208 of 208, COM within 4.62e-14 dB.

156

functions translated

146 / 146

checked against the Reference Code, executed

1,760

per-function tests all passing

28 x 10

cases x stages, every field checked

599

engine-wide audit checks, 0 failing

What it took, part 1: the release under Octave

Three blockers in the Reference Code release, all already solved on the compatibility branch

blocker	why it fails under Octave
verLessThan('matlab','7.4.1'), line 92	Octave resolves the first argument against installed Octave packages and errors. Unguarded, so it fires on every run.
touchstone reader, textscan	The reader finds frequency lines by counting NaN per row of a 9-wide textscan, which needs MATLAB line-record semantics. Octave streams numbers continuously.
MMSE, ifft	Octave ifft always returns complex; MATLAB returns real for a conjugate-symmetric input. The residue reaches the Toeplitz solve and the EQ search finds no solution at all.

A fourth, environmental: on Windows a result path over 260 characters stops the run after COM is computed. Keep RESULT_DIR short, or prefix it with \\?\ (Adam Gregory); exist() then needs the prefix too.

All three are handled on the Octave_compat branch in src/, behind OP.OCTAVE ('Octave' as the first argument). The branch's release/ files are byte-identical to the mainline releases and carry none of it.

What it took, part 2: three defects in the compatibility branch

Each was found by comparison, not by reading code. The third was found a fortnight later, on channels the first corpus did not contain

change	without it	with it
MLSE_U1_c_178A / conv_fct_noPDFx restore the p.x line conv_fct has	CDF_ev indexes a stale axis from the second loop pass. DER_MLSE 9.4e-14 against 4.6e-08. MLSE gain 4.028 dB against 1.371. COM reads 6.153 dB where the reference is 3.496.	every field matches at 1e-14. COM +2.4e-14
read_Nport_touchstone add raw = raw(~isnan(raw));	Octave textscan emits a NaN per blank line, so files with blank-line separated blocks lose their frequency axis. 12 of 164 s4p files, all on 3 channels. 24 of 208 cases crash in Tukey_Window.	24 of 24 run, all EXACT
read_Nport_touchstone read the file whole and parse it without textscan	Octave textscan can stop part way through a touchstone file, silently: 2246 of 8001 points on the branch reader. 102 of 1650 files in the new corpus cut short; 36 of them, on 10 channels, below the 67 GHz that COM uses. None in the 208-case set.	20 of 20 affected cases run, EXACT at 1.3e-14

The reader defects corrupt channel data silently. On the new corpus a short THRU passed every check: nine cases reported COM of -12 to -23.5 dB where the truth is +3.2 to +5.8 dB. The Python port, reading the same files, exposed it.

The 208-case results are unaffected: all 164 of its files read the same number of points through both readers. The first two were reported to the branch author on 2026-09-03; the third is tracked by the port maintainer.

What it took, part 3: the Octave work as a single release file

From a branch plus path overrides, to the Reference Code release plus a named patch set

	as run for this comparison	as it stands now
base	branch src/ working copy, 10 files, 321 insertions	the Reference Code release file, unmodified
Octave changes	spread through the branch, gated on OP.OCTAVE	a generator applies a patch set: 5 subfunctions replaced, 2 added, 8 line edits. Every item computes the same under MATLAB
the three defects	two override files added last on addpath; the third not yet known	the reader is in the patch set, now reading the file whole; the stale-axis defect cannot arise, since the release keeps its own conv_fct and only CDF_ev is replaced
result	208 cases, EXACT	one file per release, its header naming the release file it was generated from. Checked on 24 of the 208 cases: bit-identical to the branch route, EXACT against the Reference Code

The patch set, each item found by running the release under Octave:

verLessThan guarded (Octave resolves it against packages) | touchstone reader: whole file, parsed without textscan
MMSE: real() after the ifft; a local subfunction, so no path override reaches it | MLSE: real() on the CDF_ev arguments
CDF_ev: lookup instead of find on a growing axis, the 30-60x speed difference | Octave detected on the command line; config read and CSV write without xlsread or writecell
Optional speed items: inner loops restructured and compiled kernels, 2.5 to 3x, results bit-identical

Documented, not proposed. The list is short and it is here for the branch author to take up or leave; either way the port carries it, so an Octave run needs no MATLAB licence and no branch checkout.

Backup: COM spreadsheet, package class A

configuration 1 (config_com_dj_200G_CAKR_178_PKGA_06_2_2025_Case1.xlsx), COM_Settings sheet as used; its package block below

data rate, die load, ref impedance																			
Parameter	Setting	Units	Information	I/O control				Operational								SAVE_CONFIG2MAT	0		
f_b	106.25	GBd		DIAGNOSTICS	1	logical		ERL Pass threshold	11	dB						Receiver testing			
f_min	0.05	GHz		DISPLAY_WINDOW	0	logical		COM Pass threshold	3	db						RX_CALIBRATION	0		logical
Delta_f	0.01	GHz		CSV_REPORT	1	logical		DER_0	2.00E-04							Sigma BBN step	5.00E-03		V
C_d	[0.4e-4 0.9e-4 1.1e-4 ;0.4e-4~	nF	[TX RX]	RESULT_DIR	.\\results\\CAKR_{date}\\			T_r	0.00400	ns						ICN parameters			
L_s	[0.13 0.15 0.14; 0.13 0.15 0.14~	nH	[TX RX]	SAVE_FIGURES	0	logical		FORCE_TR	1	logical		for legacy but required				T_t	6.000		ps
C_b	[0.3e-4 0.3e-4]	nF	[TX RX]	Port Order	[1 3 2 4]			PMD_type	C2C	for MMSE use C2C only						f_v	0.371		39.42
R_0	46.25	Ohm		RUNTAG	KR_pkgA_			EW	1							T_ft	4.250		ps
PKG_NAME	PKG_LowR_CLASSA	PKG_LowR_CLASSA	TX RX	COM_CONTRIBUTION	1	logical		MLSE	1	logical						T_nt	4.250		ps
z_p select	[1]							ts_anchor	1							f_f	0.524		55.65
L	4			TDR and ERL options				sample_adjustment	[-24 24]							f_n	0.524		55.65
M	32			TDR	1	logical		Local Search	2							f_1	0.010		GHz
filter and Eq				ERL	1	logical		flim	6.70E+10	Hz						f_2	67.000		GHz
f_r	0.55	*fb		ERL_ONLY	0	ns		zero_pad	1	logical						A_ft	0.600		V
c(0)	0.54		min	TR_TDR	0.005			Filter: Rx FFE								A_nt	0.600		V
c(-1)	0		[-0.34:.0~ [min:step:max]	N	7000	logical		ffe_pre_tap_len	6	UI		d_w				Parameter	Setting		
c(-2)	0		[0.14:.02~ [min:step:max]	TDR_Butterworth	1			ffe_post_tap_len	8	UI		N_fix-d_w				board_tl_gamma0_a1_a2	[0 5.95e-4 2.6e-05]		1.4 db/in @ 53.125G
c(-3)	0		[min:step:max]	beta_x	0			ffe_pre_tap1_max	0.7	(normalized)		w_max(d_w) and -w_min(d_w)				board_tl_tau	5.790E-03		ns/mm
c(-4)	0		[min:step:max]	rho_x	0.618			ffe_post_tap1_max	0.7	(normalized)		w_max(d_w+2) and -w_min(d_w+2)				board_Z_c	92.5		Ohm
c(1)	0		[-0.2:.02~ [min:step:max]	TDR_W_TXPKG	0	UI		ffe_tapn_max	0.7	(normalized)		all other fixed w_max and w_min				z_bp (TX)	9		mm
N_b	1	UI		N_bx	16	??		num_ui_RXFF_noise	4096							z_bp (NEXT)	9		mm
b_max(1)	0.85		As/dffe1	fixture delay time	[0 0]			Floating Tap Control								z_bp (FEXT)	9		mm
b_max(2..N_b)	0		not used	Tukey_Window	1			N_bg	2	0 1 2 or 3 groups		N_wg				z_bp (FEXT)	9		mm
b_min(1)	0		As/dffe1	Z_t	46.25			N_bf	4	taps per group		N_wf				z_bp (RX)	9		mm
b_min(2..N_b)	0		not used	Noise, jitter		UI		N_f	80	UI span for floating taps		Nmax-d_w-1				C_0	[0 0]		nF
g_DC	[-20:1:0]	dB	[min:step:max]	sigma_Rj	0.01	UI		bmaxg	0.05	max FFE value for floating taps		all floating w_max and w_min				C_1	[0 0]		nF
f_z	42.50	GHz		A_AD	0.02	V^2/GHz		N_tail_start	9	(UI) start of tail taps limit		not supposed to be used but untested				Include PCB	0		logical
f_p1	42.50	GHz		eta_0	7.50E-09	dB		TS_SRCH_MODE	full-sweep										
f_p2	106.25	GHz		SNR_TX	33.5			Non-zero Local Search Method	1	0: Current local search; 1: Ada~		This only comes into play when "Local Search" is greater than zero							
g_DC_HP	[-6:1:0]		[min:step:max]	R_LM	0.95			Clip Method	slow	"slow" or "Fast": Typing mistake fixed in COM v4p16									
f_HP_PZ	1.328125	GHz		N_qb	6														
				P_qc	1.00E-07														

.START	PKG_LowR_CLASSA		
Parameter	Setting	Units	Information
package_tl_gamma0_a1_a2	[0.0005 0.00089 0.0002]		
package_tl_tau	0.006141	ns/mm	
package_Z_c	[87.5 87.5 ; 95 95 ; 100 100; ~	Ohm	
R_d	[46.25 46.25]	Ohm	[TX RX]
z_p (TX)	[12 33 33 33 ; 1.8 1.8 1.8 1~	mm	[test cases]
z_p (NEXT)	[12 33 33 33 ; 1.8 1.8 1.8 1~	mm	[test cases]
z_p (FEXT)	[12 33 33 33 ; 1.8 1.8 1.8 1~	mm	[test cases]
z_p (RX)	[12 33 33 33 ; 1.8 1.8 1.8 1~	mm	[test cases]
C_p	[0.4e-4 0.4e-4]	nF	[TX RX]
A_v	0.385	V	Vf=0.400
A_fe	0.385	V	Vf=0.399
A_ne	0.481	V	Vf=0.400
.END			

Configuration 2 differs from 1 only in COM_CONTRIBUTION (0 against 1), z_p select ([2] against [1]). The with-crosstalk workbook (_sweep_TxFFE) sweeps the Tx FFE: c(-1) [-0.34:.02:0], c(-2) [0:.02:0.14], c(1) [-0.2:.02:0]; without crosstalk those value cells are 0, and the ranges beside them are a template COM does not read.

Backup: COM spreadsheet, package class B

configuration 3 (config_com_dj_200G_CAKR_178_PKGB_06_2_2025_Case3.xlsx), COM_Settings sheet as used; its package block below

data rate, die load, ref impedance																					
Parameter	Setting	Units	Information	I/O control		Operational						SAVE_CONFIG2MAT		0							
f_b	106.25	GBd		DIAGNOSTICS	1	logical	ERL Pass threshold	11	dB				RX_CALIBRATION	0		logical					
f_min	0.05	GHz		DISPLAY_WINDOW	0	logical	COM Pass threshold	3	db				Sigma BBN step	5.00E-03		V					
Delta_f	0.01	GHz		CSV_REPORT	1	logical	DER_0	2.00E-04					ICN parameters								
C_d	[0.4e-4 0.9e-4 1.1e-4 ;0.4e-4~	nF	[TX RX]	RESULT_DIR	.\results\CAKR_{date}\		T_r	0.00400	ns				T_t	6.000		ps					
L_s	[0.13 0.15 0.14; 0.13 0.15 0.14~	nH	[TX RX]	SAVE_FIGURES	1	logical	FORCE_TR	1	logical				f_v	0.371		39.42					
C_b	[0.3e-4 0.3e-4]	nF	[TX RX]	Port Order	[1 3 2 4]		PMD_type	C2C					T_ft	4.250		ps				Table 17-	
R_0	46.25	Ohm		RUNTAG	KR_pkgB_		EW	1					T_nt	4.250		ps					
PKG_NAME	PKG_HIR_CLASSB PKG_HIR_CLASSB		TX RX	COM_CONTRIBUTION	0	logical	MLSE	1	logical				f_f	0.524		55.65					
z_p select	[3]			TDR and ERL options			ts_anchor	1					f_n	0.524		55.65					
L	4			TDR	1	logical	sample_adjustment	[-24 24]					f_1	0.010		GHz					
M	32			ERL	1	logical	Local Search	2					f_2	67.000		GHz					
filter and Eq				ERL_ONLY	0	ns	flii	6.70E+10	Hz				A_ft	0.600		V					
f_r	0.55	*fb		TR_TDR	0.005		zero_pad	1	logical				A_nt	0.600		V					
c(0)	0.54		min	N	7000	logical	Filter: Rx FFE														
c(-1)	0		[-0.34:.0~	TDR_Butterworth	1		ffe_pre_tap_len	6	UI				d_w								
c(-2)	0		[0.14:.02~	beta_x	0		ffe_post_tap_len	8	UI				N_fix-d_w								
c(-3)	0		[min:step:max]	rho_x	0.618		ffe_pre_tap1_max	0.7	(normalized)				w_max(d_w) and -w_min(d_w)								
c(-4)	0		[min:step:max]	TDR_W_TXPKG	0	UI	ffe_post_tap1_max	0.7	(normalized)				w_max(d_w+2) and -w_min(d_w+2)								
c(1)	0		[-0.2:.02~	N_bx	16	??	ffe_tapn_max	0.7	(normalized)				all other fixed w_max and w_min								
N_b	1	UI	[min:step:max]	fixture delay time	[0 0]		num_ui_RXFF_noise	4096													
b_max(1)	0.85		As/dffe1	Tukey_Window	1		Floating Tap Control														
b_max(2..N_b)	0		not used	Z_t	[46.25]		N_bg	2	0 1 2 or 3 groups				N_wg								
b_min(1)	0		As/dffe1	Noise, jitter		UI	N_bf	4	taps per group				N_wf								
b_min(2..N_b)	0		not used	sigma_RJ	0.01	UI	N_f	80	UI span for floating taps				Nmax-d_w-1								
g_DC	[-20:1:0]	dB	[min:step:max]	A_DD	0.02	V^2/GHz	bmaxg	0.05	max FFE value for floating taps				all floating w_max and w_min								
f_z	42.50	GHz		eta_0	7.50E-09	dB	N_tail_start	9	(UI) start of tail taps limit				not supposed to be used but untested								
f_p1	42.50	GHz		SNR_TX	33.5		TS_SRCH_MODE	full-sweep													
f_p2	106.25	GHz		R_LM	0.95		Non-zero Local Search Method	1	0: Current local search; 1: Ada~				This only comes into play when "Local Search" is greater than zero								
g_DC_HP	[-6:1:0]		[min:step:max]	N_qb	6		Clip Method	slow	"slow" or "Fast": Typing mistake fixed in COM v4p16												
f_HP_PZ	1.328125	GHz		P_qc	1.00E-07																

.START	PKG_HIR_CLASSB		
Parameter	Setting	Units	Information
package_tl_gamma0_a1_a2	[0.0005 0.00065 0.000293]		
package_tl_tau	0.006141	ns/mm	
package_Z_c	[87.5 87.5 ; 95 95 ; 100 100; ~	Ohm	
R_d	[46.25 46.25]	Ohm	[TX RX]
z_p (TX)	[8 24 30 45 ; 2 2 2 2 ; 1.3 ~	mm	[test cases]
z_p (NEXT)	[8 24 29 44 ; 2 2 2 2 ; 1.3 ~	mm	[test cases]
z_p (FEXT)	[8 24 30 45 ; 2 2 2 2 ; 1.3 ~	mm	[test cases]
z_p (RX)	[8 24 29 44 ; 2 2 2 2 ; 1.3 ~	mm	[test cases]
C_p	[0.4e-4 0.4e-4]	nF	[TX RX]
A_v	0.385	V	Vf=0.400
A_fe	0.385	V	Vf=0.399
A_ne	0.481	V	Vf=0.400
.END			

Configuration 4 differs from 3 only in z_p select ([4] against [3]). The with-crosstalk workbook (_sweep_TxFFE) sweeps the Tx FFE: c(-1) [-0.34:.02:0], c(-2) [0:.02:0.14], c(1) [-0.2:.02:0]; without crosstalk those value cells are 0, and the ranges beside them are a template COM does not read.