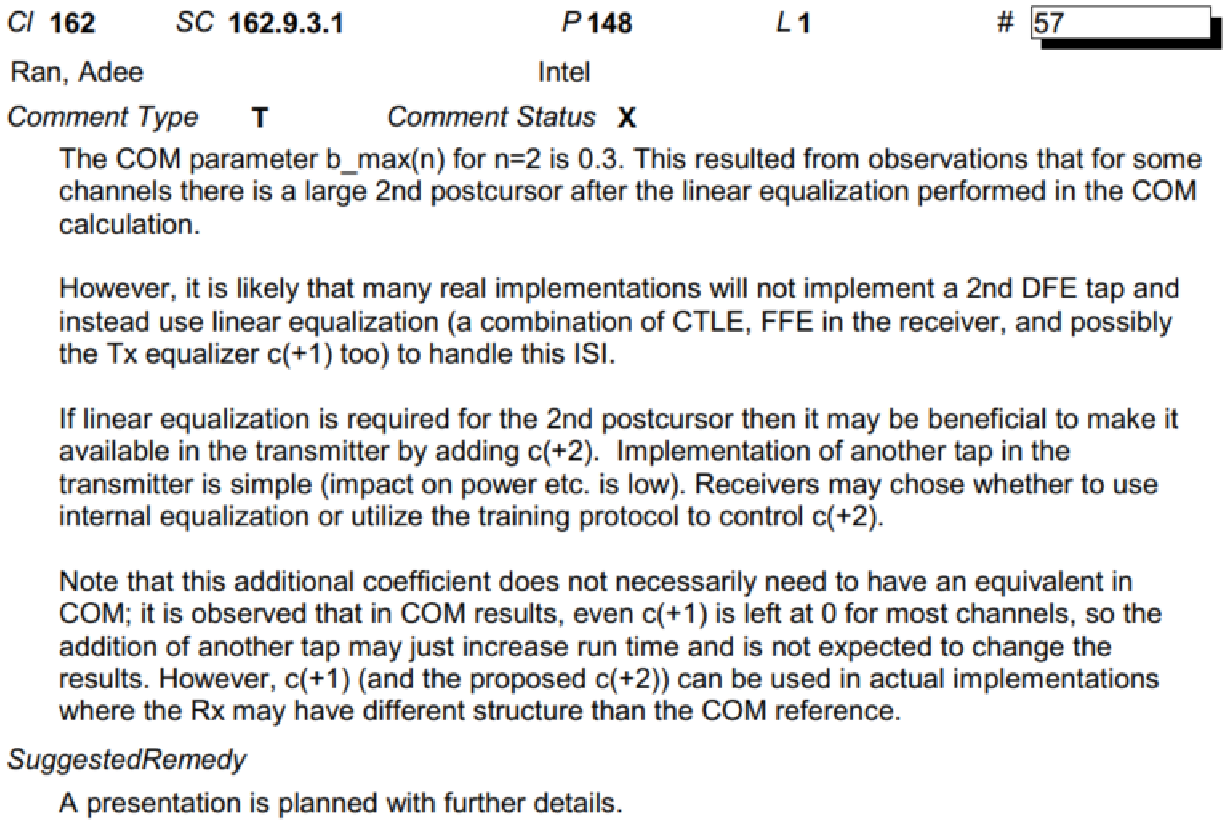
ADDING C(+2) COEFFICIENT IN TX EQUALIZER
(COMMENT #57)

Adee Ran, Intel

May 2020

Supporters

Comment



Cl **162** *SC* **162.9.3.1** *P* **148** *L* **1** # 57

Ran, Adee Intel

Comment Type **T** *Comment Status* **X**

The COM parameter b_max(n) for n=2 is 0.3. This resulted from observations that for some channels there is a large 2nd postcursor after the linear equalization performed in the COM calculation.

However, it is likely that many real implementations will not implement a 2nd DFE tap and instead use linear equalization (a combination of CTLE, FFE in the receiver, and possibly the Tx equalizer c(+1) too) to handle this ISI.

If linear equalization is required for the 2nd postcursor then it may be beneficial to make it available in the transmitter by adding c(+2). Implementation of another tap in the transmitter is simple (impact on power etc. is low). Receivers may chose whether to use internal equalization or utilize the training protocol to control c(+2).

Note that this additional coefficient does not necessarily need to have an equivalent in COM; it is observed that in COM results, even c(+1) is left at 0 for most channels, so the addition of another tap may just increase run time and is not expected to change the results. However, c(+1) (and the proposed c(+2)) can be used in actual implementations where the Rx may have different structure than the COM reference.

SuggestedRemedy

A presentation is planned with further details.

Proposal

- Extend the transmit equalizer to a six-tap FFE, with coefficients c(*k*), where *k* = –3 to 2 (add c(+2))
- Editorial changes are in following places in the draft:
  - 162.9.3.1 Transmitter output waveform and all descendant subclauses
    - Same step size as other coefficients
    - Min: -0.1, Max: 0.1, set to 0 in all presets
  - 162.8.11 PMD control function – extend valid transmitter equalizer coefficient indices to include +2

What is the problem?

- Simulations and COM analysis with high loss channels show need for strong equalization of 2nd postcursor
  - This resulted in increase of $b_{max}(2)$ to 0.3, Motion #13 in the November 2019 meeting
  - See also heck_3ck_01_0919
  - Probably a real issue!
- Problem: real receivers don’t have equalizers identical to COM’s CTLE and DFE
  - This additional equalization requirement does not come for free

Proposed solution

- Implementing linear equalization in the Tx is known to be much cheaper than in the Rx
  - The impact of adding another tap is incremental
  - Tx adaptation protocol is already available
- Using extra help from Tx c(+2) can enable better Rx implementations
  - Less power…
  - More channels supported…
- If a receiver can do without c(+2), it can just not use it.

Why do we need a new tap now?

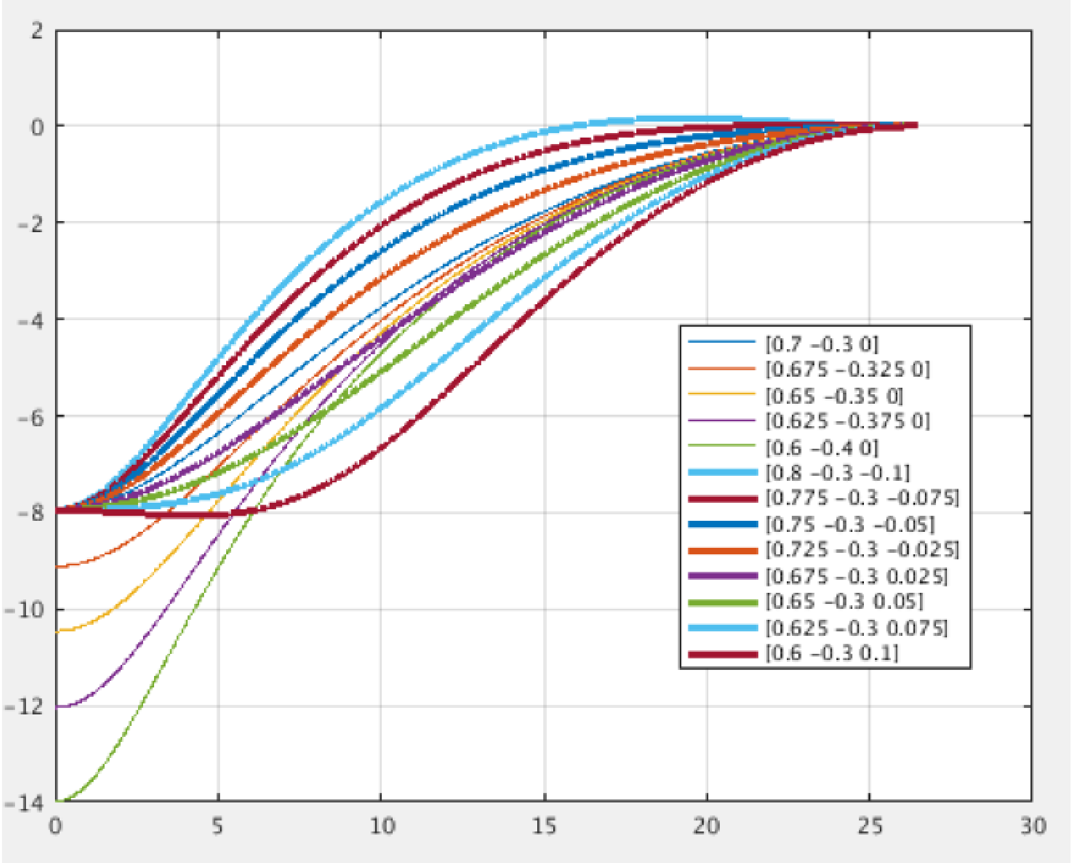
- For the channels we want to work on, IL at half of Nyquist frequency (~13 GHz) is not negligible
  - This is the Nyquist frequency at 50G – where we had CTLE tuned to that bandwidth
  - The current CTLE is tuned to a higher bandwidth
- The reference CTLE does not perfectly equalize all frequencies
  - Nor do the real CTLEs in receivers
- Many Rx implementations for 100G will use ADC and then DSP equalization
  - Increasing the digital equalization requirements can impact performance (quantization noise)
  - Equalizing early in the channel can reduce peak-to-average ratio, thus improving ADC utilization and receiver SNR
- Building good tunable CTLEs is notoriously difficult
  - Backward compatibility for lower rates makes this even harder
  - Real implementations are quite different from the COM model – the “DFE” coefficients may not look like what COM predicts (may be even higher)
  - A tunable c(+2) can mitigate many practical issues

Do we need to add c(+2) in COM?

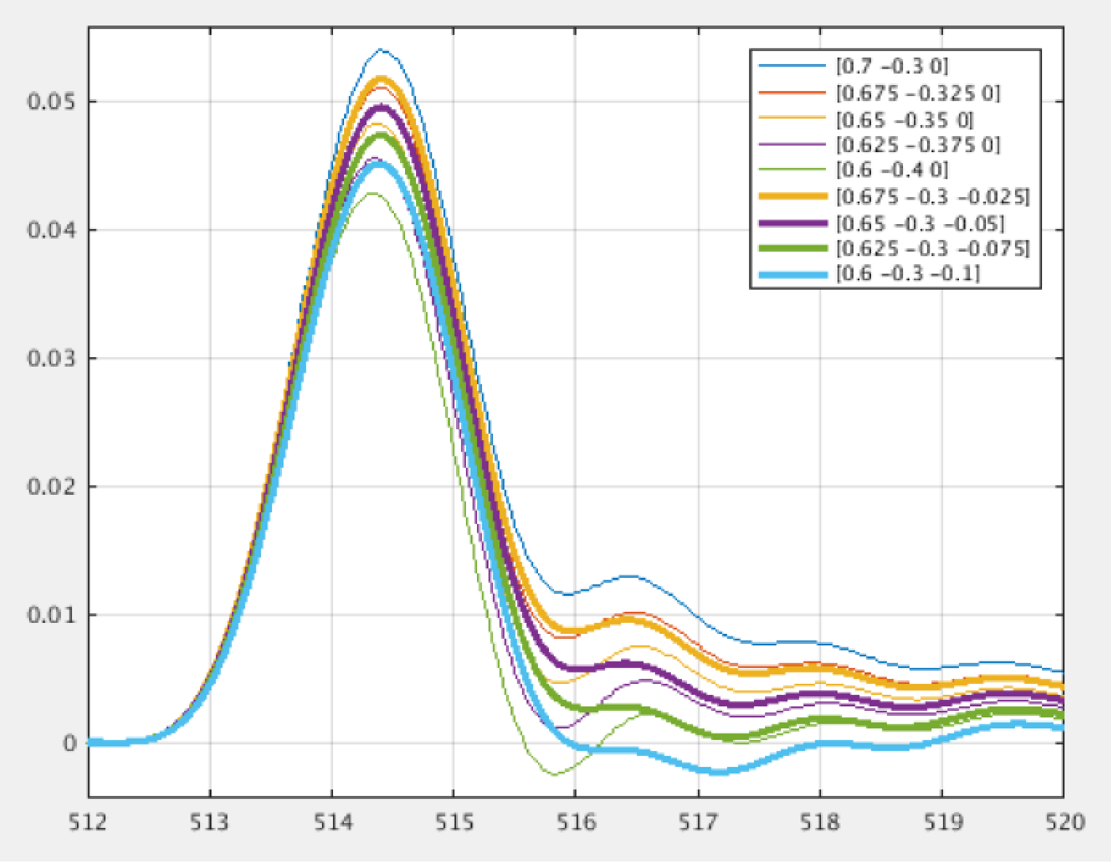
- Not necessarily.
- Even c(+1) that exists today is usually set to 0
  - But it does not mean it is useless… real receivers can benefit from it
- If we don’t reduce $b_{max}(2)$ then c(+2) will also be set to 0

BACKUP

Frequency domain effects of C(+2)



Time domain effects of C(+2)


0.05
0.04
0.03
0.02
0.01
0
512
513
514
515
516
517
518
519
520
[0.7 -0.3 0]
[0.675 -0.325 0]
[0.65 -0.35 0]
[0.625 -0.375 0]
[0.6 -0.4 0]
[0.675 -0.3 -0.025]
[0.65 -0.3 -0.05]
[0.625 -0.3 -0.075]
[0.6 -0.3 -0.1]