RX FFE Implementation Algorithm for COM 4.1

Richard Mellitz, Samtec
Kent Lusted, Intel

August 31, 2023

Purpose

- Explain the COM 4.1 Rx FFE determination algorithm

Review from COM 4.1 Update

mellitz_3dj_01a_elec_230817

Rx_{FFE} is within the Full Grid Optimization Loop and includes T_s sweep

Added COM parameter	*Example value*	*Default*	*information*	*notes*
sample_adjustment	[-32 32]	0	Min max sample offset range from ts anchor	Integer related to M
ts_anchor	1	0	Ts anchor for sample adjustment (0,1)	See next slides

Full grid loop hierarchy

1. CTF
   - for each G_{DC} and G_{DC2}
2. Tx_{ffe}
   - for each C(n)
3. For RxFFE only (new for COM 4.1)
   Determine T_s (like a CDR)
   - Initial T_s anchor uses Mueller-Muller (MM), PR peak, or max dv/UI and then continue for each oversample step in "sample_adjustment"
   - If sample_adjustment=0 then only MM is used for T_s and thus Rx_{ffe}
   - Sample adjustment
   - Find Rx_{FFE} taps $C_{rx}(n)$ and apply
4. For Rx_{FFE} with sample_adjustment=0 or no Rx_{FFE}
   - T_s is determined from Mueller-Muller (MM), equation 93A–25
5. Compute FOM for steps [1 2 3 4]
6. Determine variable equalizer settings for best FOM

Rx FFE taps setting are determined in the inner most loop

IEEE P802.3dj 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Ethernet Task Force

Inner Most Loop: Rx Feed Forward Equalizer (Rx_{FFE}) Determination Algorithm

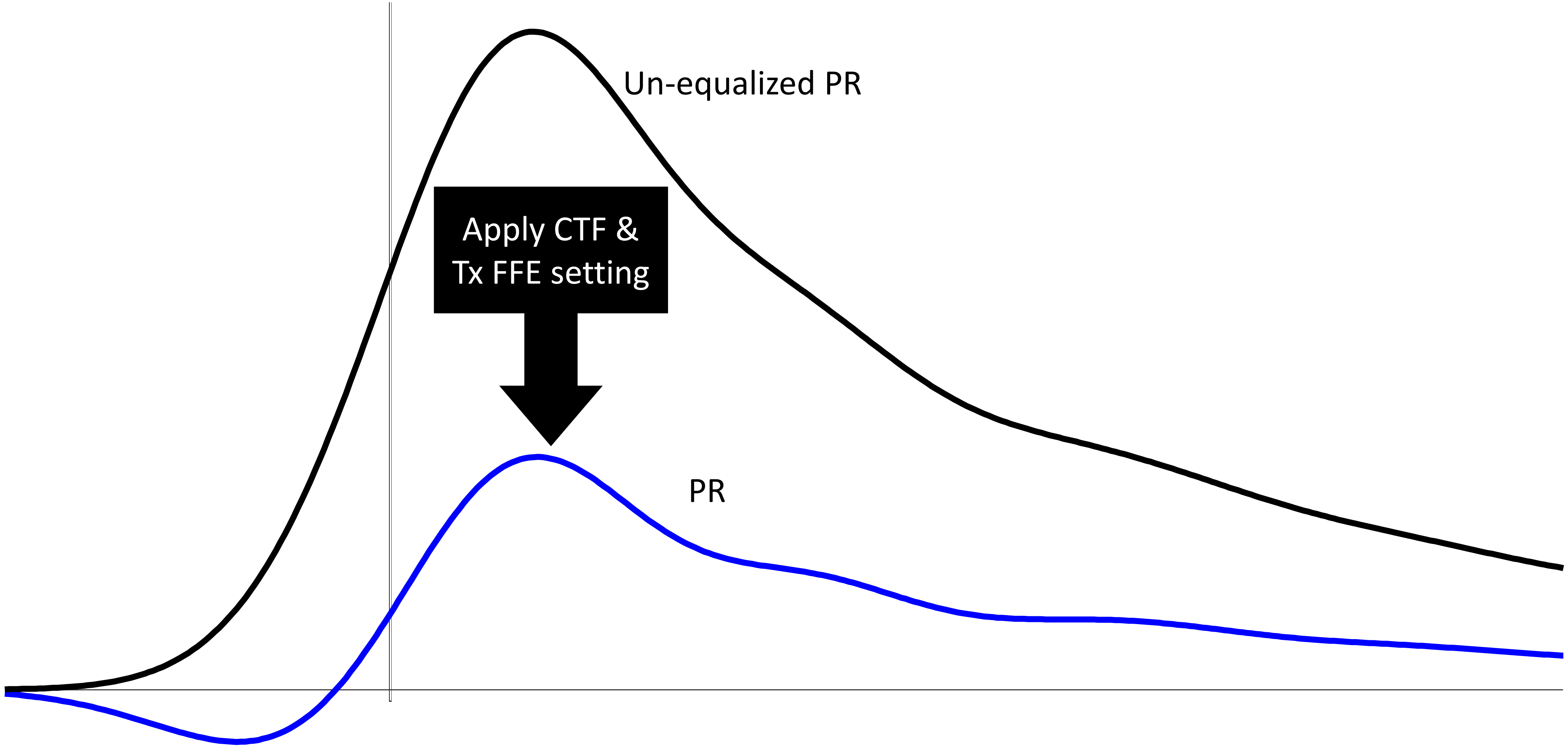
- Rx_{FFE} settings are computed for each Tx, CTF, and sample point setting
  - These are called variable equalizer parameters (93A.1.6)
- A figure of merit (FOM) is used to select the best combination of settings
  - The FOM is a signal to noise ratio (IEEE802.3 Eq 93A-3)
  - $FOM = 10\ log10\left(\frac{A_s^2}{\sigma_{TX}^2+\sigma_{ISI}^2+\sigma_J^2+\sigma_{XT}^2+\sigma_N^2}\right)$
  - The last step of the inner most loop is to determine FOM

Rx Feed Forward Equalizer (FFE) Determination

- ❑ The Rx FFE is implemented in COM 4.1 is a least means squares (LMS) method
- ❑ The mechanics of the LMS method uses a desired response or forcing vector (FV) derived from the pulse response based on the following
  - $FV = PR \otimes C$
    Where C is a vector of Rx FFE tap coefficients and PR is the pulse response.
  - C is solved for in each loop
  - Vector forcing is adjusted for DFE
- The LMS method can comprehend noise for determination of C but further discussion is needed
  - Consider the noise power is related to $\sigma_{TX}^2 + \sigma_{ISI}^2 + \sigma_J^2 + \sigma_{XT}^2 + \sigma_N^2$ …

The inner loop starts with a pulse response (PR)

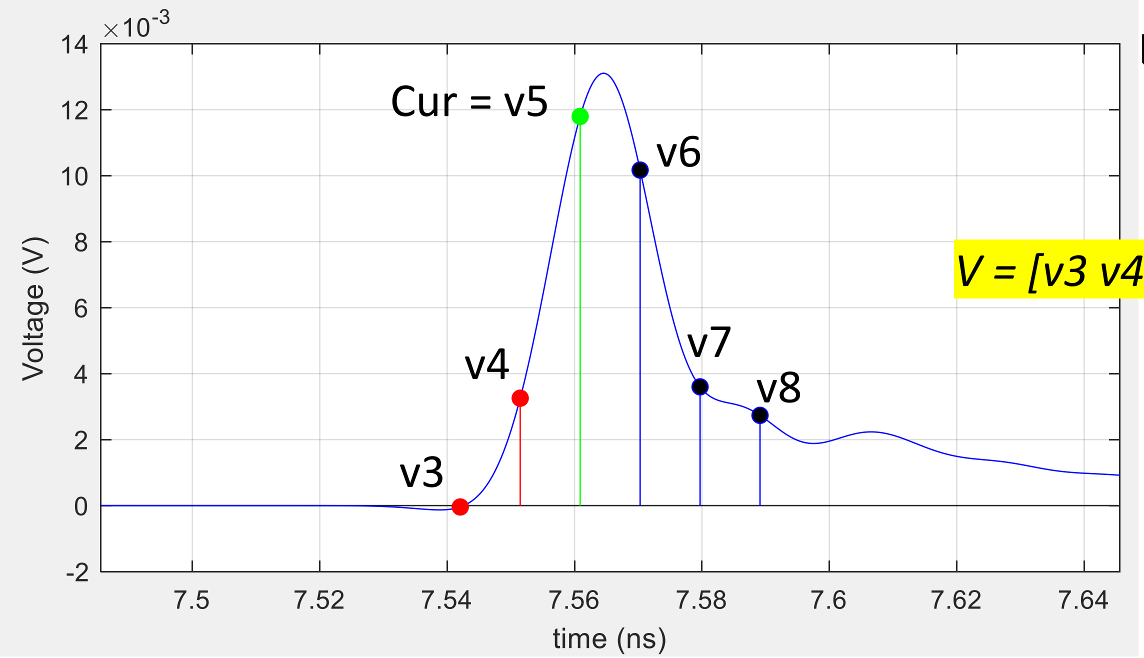
FOR EACH TX_{FFE} AND CTF SETTING



Vector Forcing Algorithm Description

Example: cmx=2 and cpx=3

- The algothim starts with a PR
- The result is a set of Rx_{FFE} tap coefficients, C
- V is the sub sampled PR

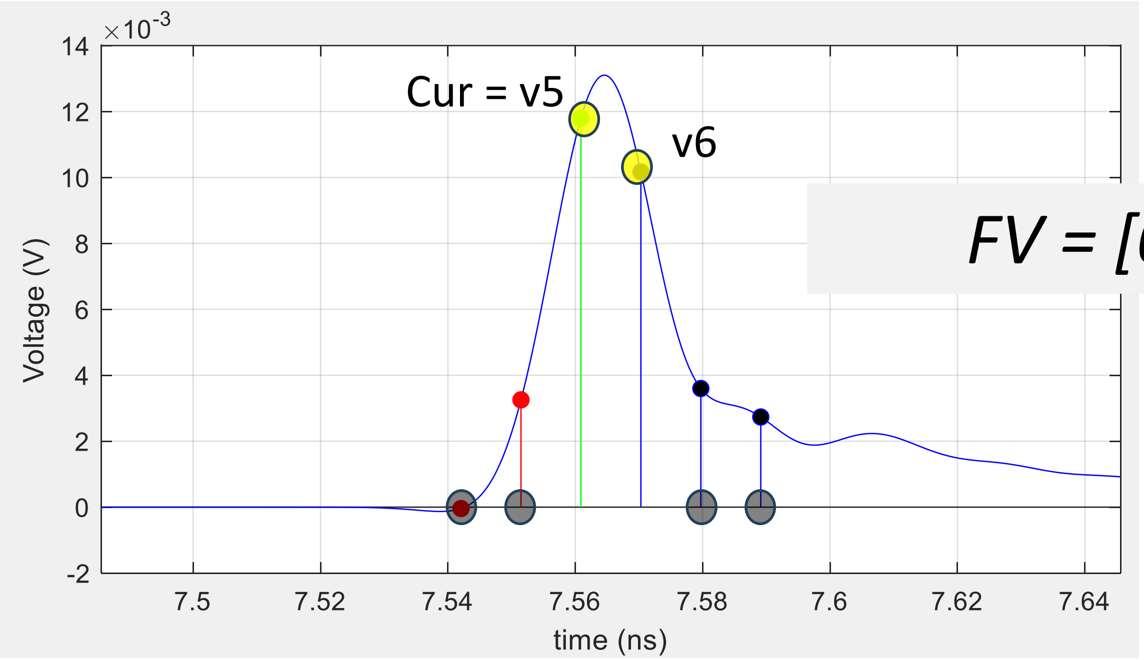


- V vector of UI spaced samples referenced to the sample point, T_s, between cmx and cpx
  - Where cmx and cpx are respectively the number of pre and post tap
- In this example the PR voltage at T_s is called “cur” or cursor

V = [v3 v4 cur v6 v7 v8]

Forcing vector FV

DESIRED RESPONSE IS 0 V EVERYWHERE EXCEPT AT THE CURSOR AND POST CURSOR



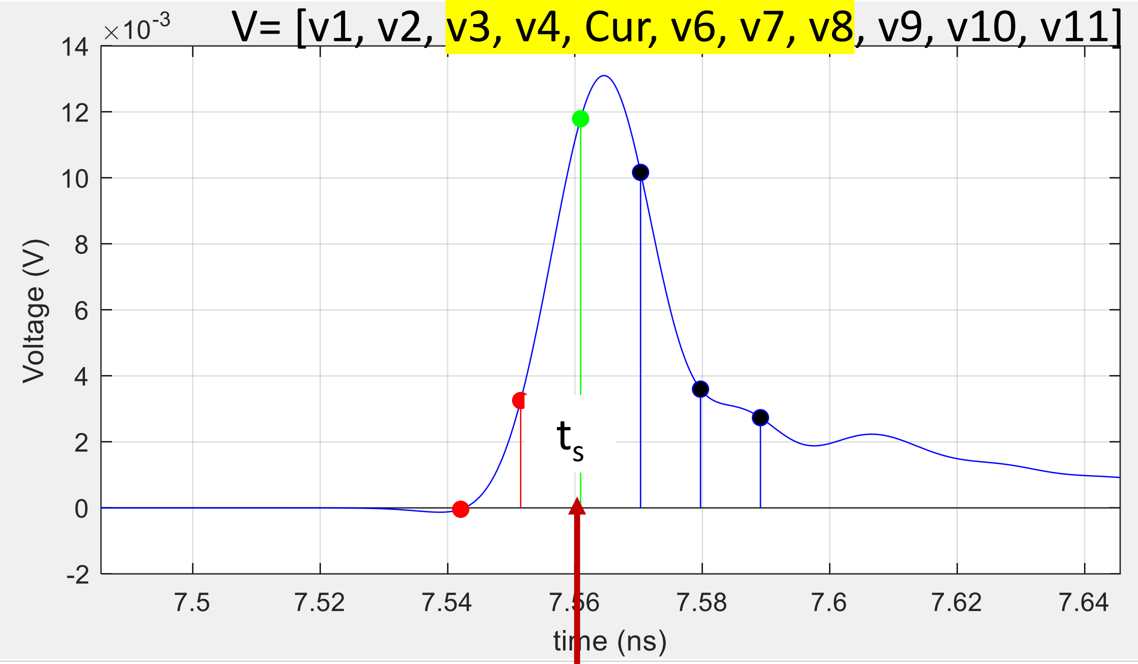
FV = [0 0 cur v6 0 0]

v6 in not allowed to be more the b_{max}(1)

Set up the matrix VV

USING THE PRIOR EXAMPLE WHERE CMX=2 AND CPX=3

- At this point we have V and FV
- $FV = PR \otimes C$ is a convolution operation
  - Convolution is a basically shifting and adding
- Rewriting in matrix form
  - $\overline{FV} = [VV]\bar{C}$
  - VV, the convolution matrix, is pre-shifted instance of V



$$\begin{pmatrix}
Cur & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 \\
v_4 & Cur & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 \\
v_3 & v_4 & Cur & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 \\
v_2 & v_3 & v_4 & Cur & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 \\
v_1 & v_2 & v_3 & v_4 & Cur & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} \\
v_{11} & v_1 & v_2 & v_3 & v_4 & Cur & v_6 & v_7 & v_8 & v_9 & v_{10} \\
v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 & Cur & v_6 & v_7 & v_8 & v_9 \\
v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 & Cur & v_6 & v_7 & v_8 \\
v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 & Cur & v_6 & v_7 \\
v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 & Cur & v_6 \\
v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_1 & v_2 & v_3 & v_4 & Cur
\end{pmatrix}$$

$$VV= \begin{pmatrix}
Cur & v_6 & v_7 & v_8 & v_9 & v_{10} \\
v_4 & Cur & v_6 & v_7 & v_8 & v_9 \\
v_3 & v_4 & Cur & v_6 & v_7 & v_8 \\
v_2 & v_3 & v_4 & Cur & v_6 & v_7 \\
v_1 & v_2 & v_3 & v_4 & Cur & v_6 \\
v_{11} & v_1 & v_2 & v_3 & v_4 & Cur
\end{pmatrix}$$

Determine Rx_{FFE} taps

SOLVE FOR C

- ❑ The solution for C from $\overline{FV} = [VV]\bar{C}$ is an LMS fit
- ❑ C becomes the Rx_{FFE} coefficients used in the inner loop
  - $\bar{C} = (([VV]^T[VV])^{-1}[VV]^T)^T FV^T$
  - C is normalized such that C(0)=1
    - $C = C/C(0)$

COM is Computed as before

AFTER ALL THE VARIABLE EQUALIZER SETTING ARE APPLIED

- Except that Rx_{FFE} has been added to the frequency domain filtering for crosstalk and through channels.
- ISI and crosstalk frequency domain responses are converted to time domain and then to the statistical domain ([1]PDF).
- Other noise PDFs from eta_0, jitter, and SNR_Tx are computed
- The COM script proceeds to combine all the noise PDFs and determines total noise PDF which is converter to a noise [2]CDF
- The total noise CDF and the available signal (A_s) are used to determine COM at DER0

[1]PDF probability density function
[2]CDF cumulate distribution function

Next steps

- Consensus discussion on merits and methods for including noise in the Rx determination step
  - Maybe VV= VV + randn(rank(VV))*noise
- Investigate/review modifications to VV to reduce the likelihood of ill condition matrix operations

Thank You!