IEEE 802.3da – Transmitter model

Piergiorgio Beruto

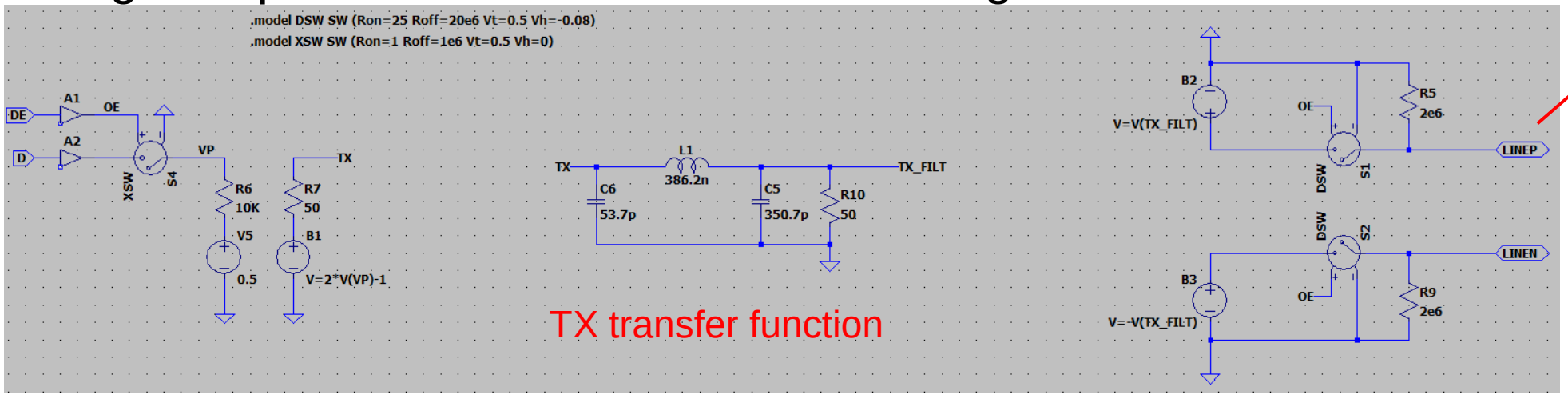
Outline

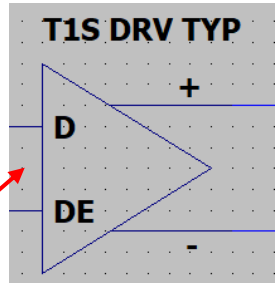
- Follow-up on Mixing Segment definition and Consensus Model
  - https://www.ieee802.org/3/da/public/040622/diminico_SPMD_01_0422.pdf (Chris DiMinico et al.)
  - https://www.ieee802.org/3/da/public/030922/Paul_01_da_03092022.pdf (Michael Paul)
- The PSD mask definition in Clause 147.5.4.4 allows a wide range of transmitters that produce a variety of eye diagrams
  - The channel models presented so far implicitly assume a “typical” transmitter PSD
  - What about the “worst” case?
- This presentation shows transmitter models that (almost) match the PSD mask limits and droop specifications
  - Proposal for adopting the TX model into the consensus model
    - https://github.com/SPE-MD/SPMD-Simulations

Modeling the TX

Model description

- Using LT-Spice to model the transmitter stage

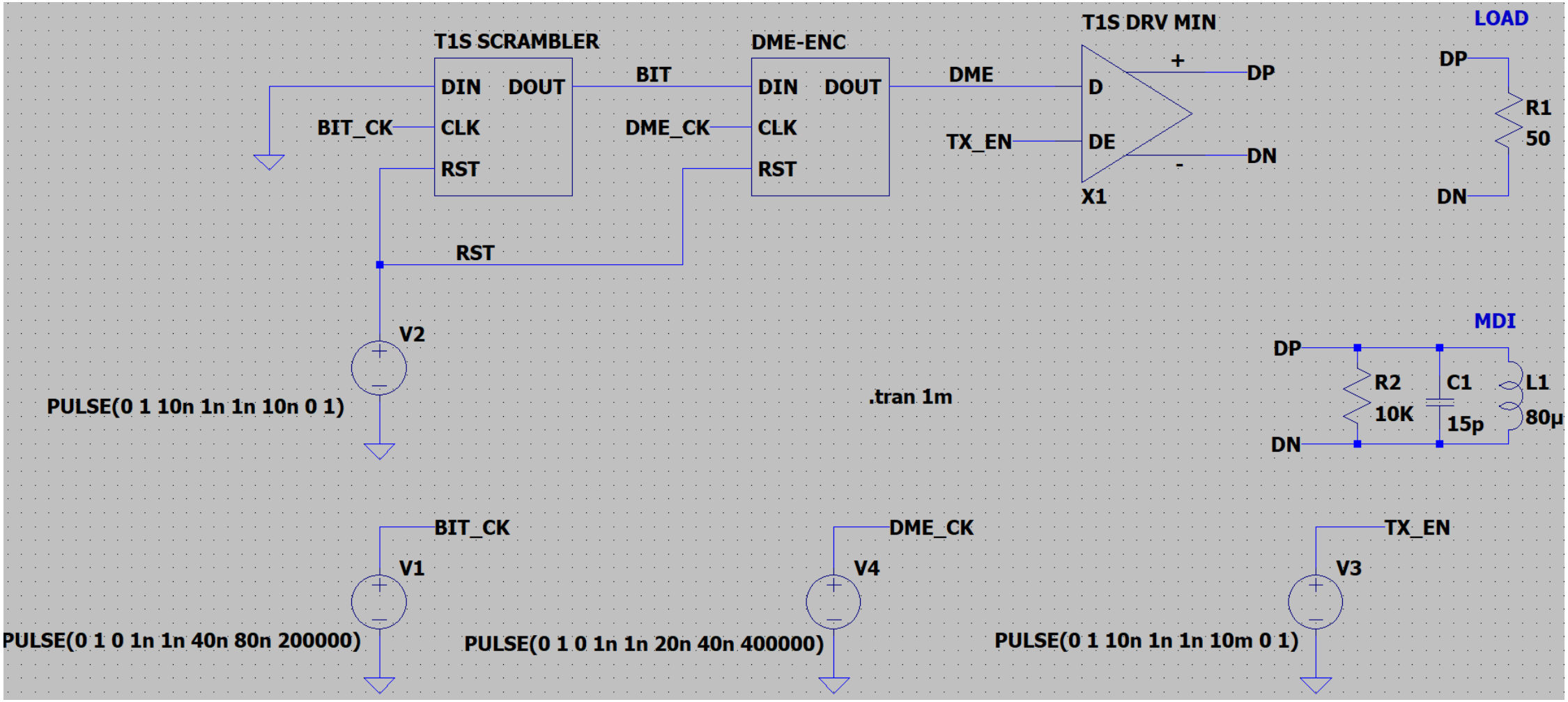




- Python script to read LT-Spice data and calculate the PSD / eye diagram
- Change the TX transfer function to get as close as possible to the defined limits
- Many transfer functions are “unreasonable” to implement, but everything that meets the PSD mask is allowed (in principle)

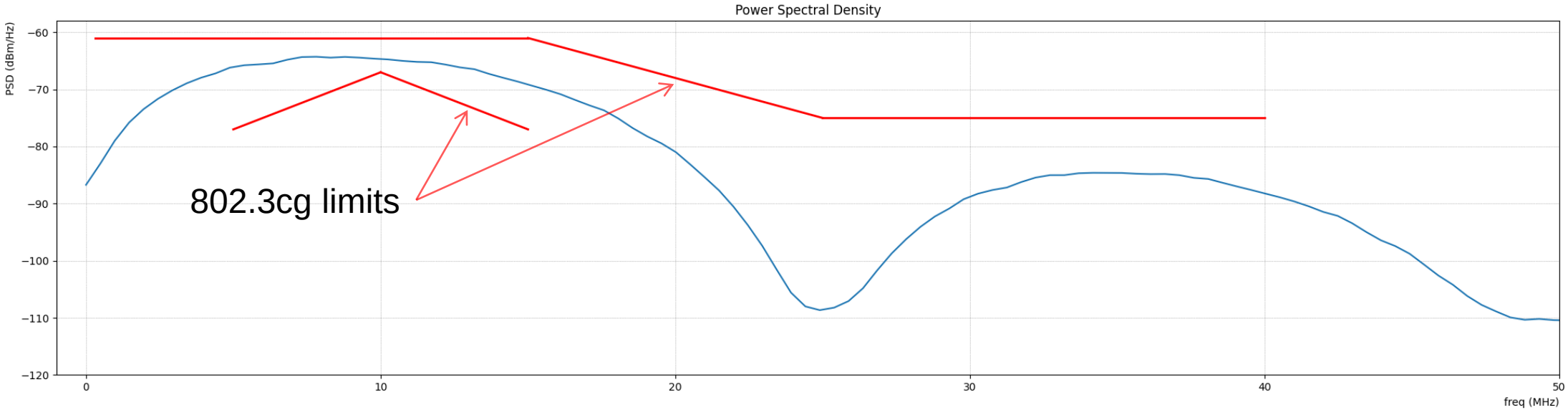
```python
def run(name, skipsim):
    if not skipsim:
        # run LTSPICE simulation
        run_ltsim(name)

    # get the differential voltage
    vdiff = get_vdiff(name, fs, noise=-25.0)

    # estimate the PSD using Welch's method
    # f   = frequencies [Hz]
    # psd = spectral density [V^2/Hz]
    f, psd = sig.welch(vdiff, fs, nperseg=1024, scaling='density')

    # convert frequencies to MHz
    f /= 1e6

    # convert PSD to dBm/Hz
    # note: V^2 / R * 1000 = power in mW
    psd = 10 * np.log10(psd * 1000.0 / Z)

    # do the plot! ------------------------------------------------------- #

    plt.figure()
    grid = (2, 2)
    wnd = (
        plt.subplot2grid(grid, (0, 0), colspan=2, rowspan=1),
        plt.subplot2grid(grid, (1, 0), colspan=1, rowspan=1),
        plt.subplot2grid(grid, (1, 1), colspan=1, rowspan=1)
    )

    # plot the PSD data -------------------------------------------------- #

    wnd[0].set_title('Power Spectral Density')
    wnd[0].plot(f, psd)
```

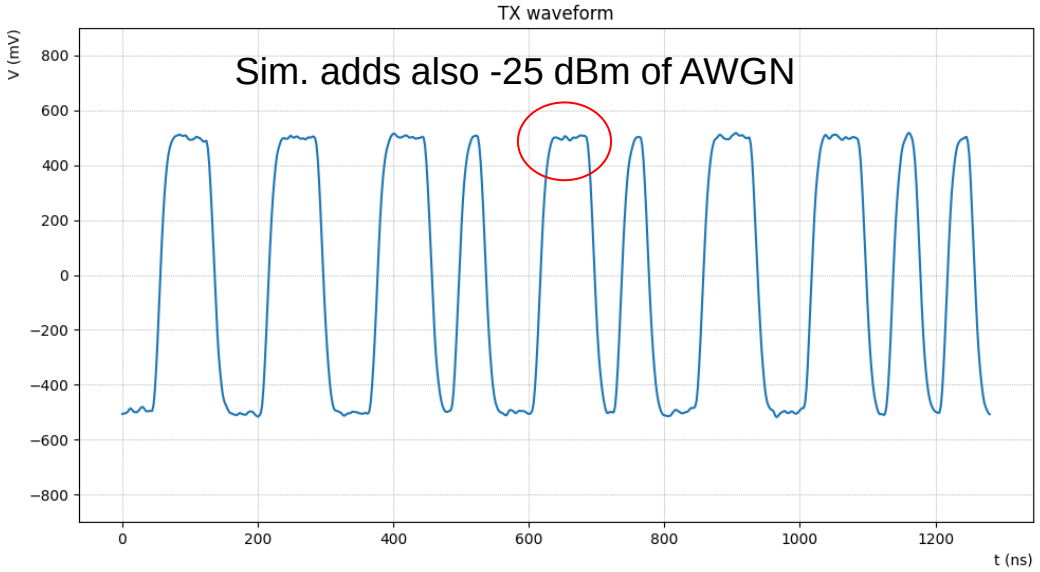
Testbench

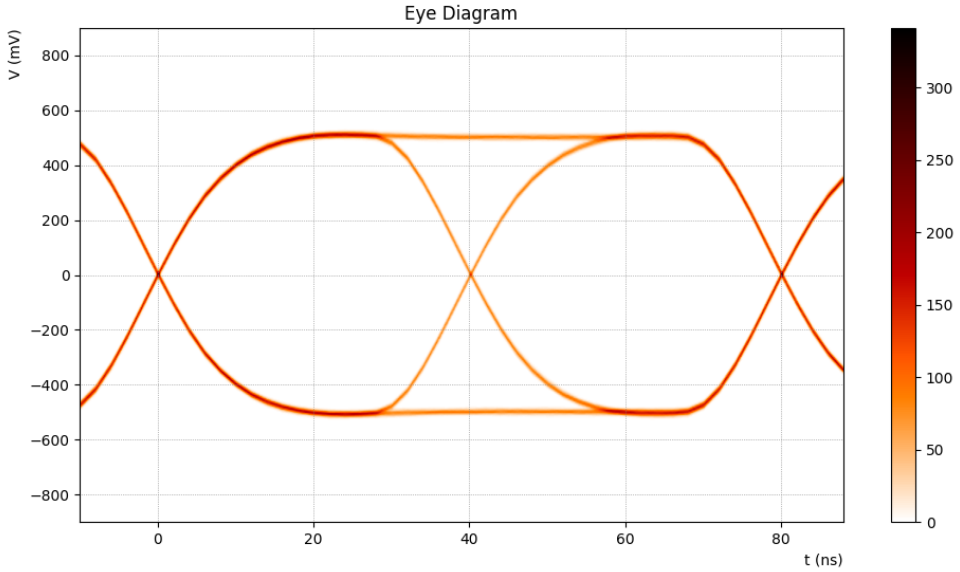


PSD is calculated from V(DP)-V(DN) i.e., differential output voltage on Z = 50Ω

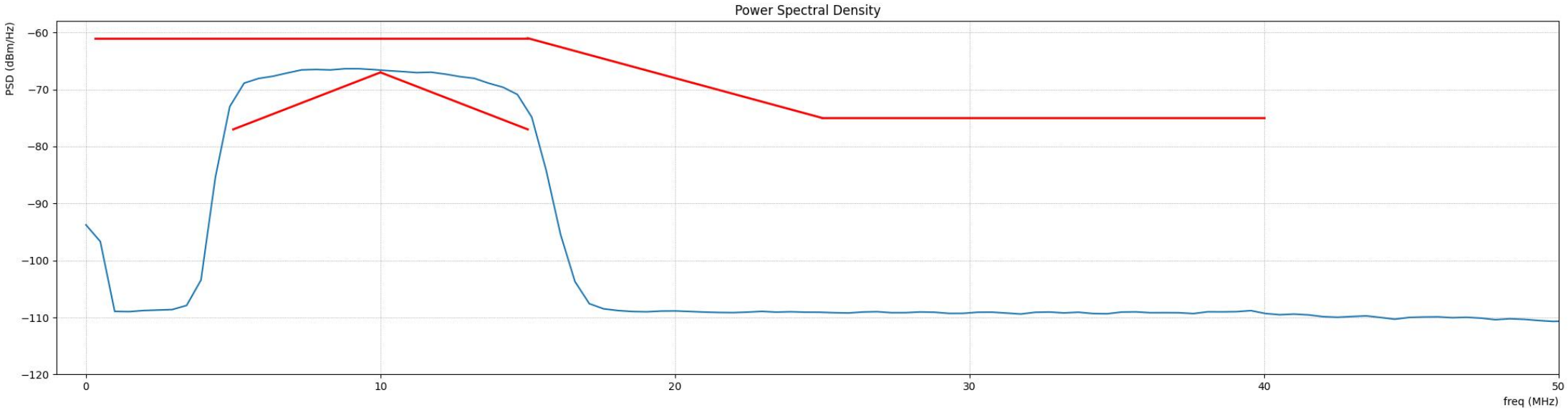
Typical 802.3cg transmitter model

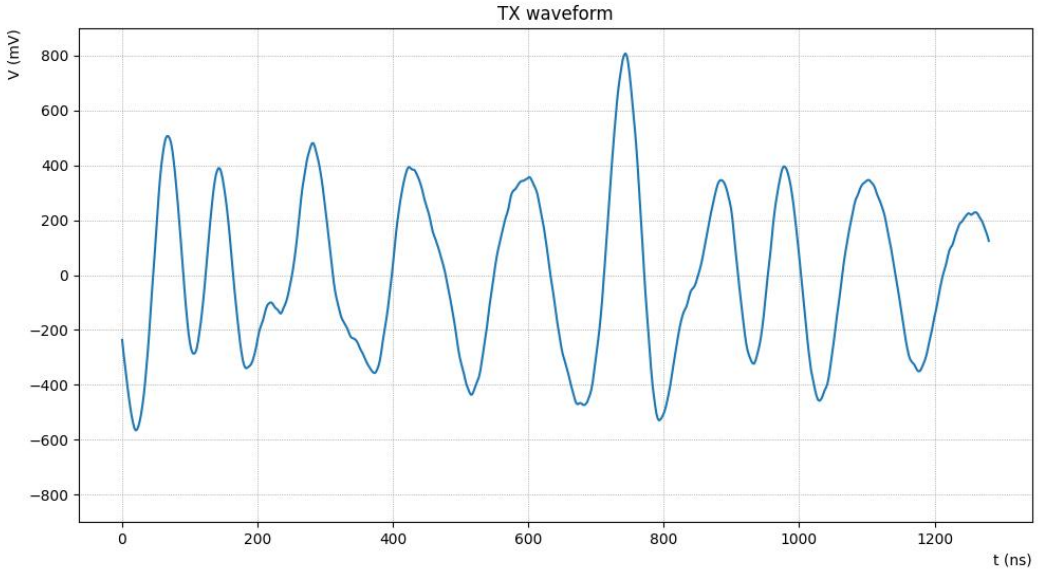


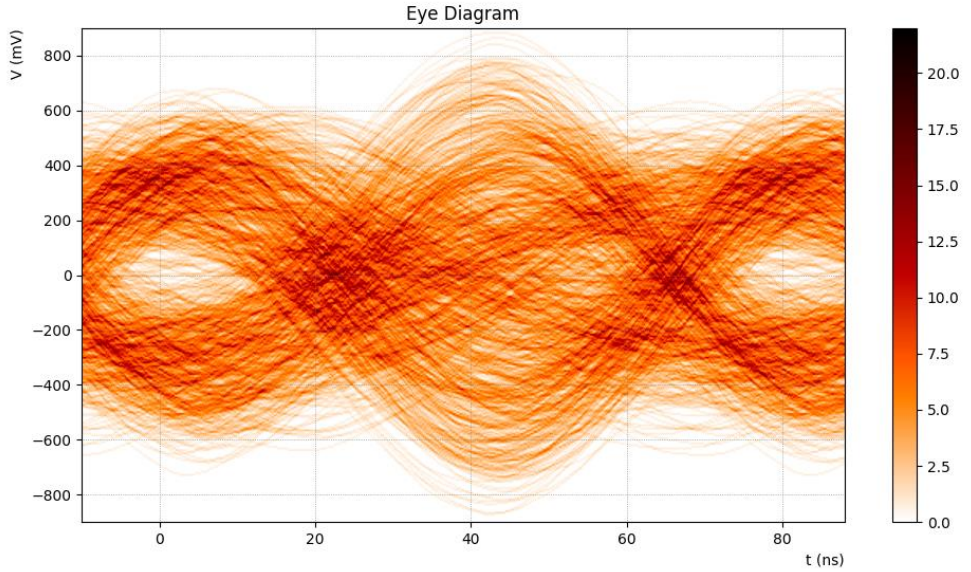




“Worst” 802.3cg transmitter model allowed by PSD mask

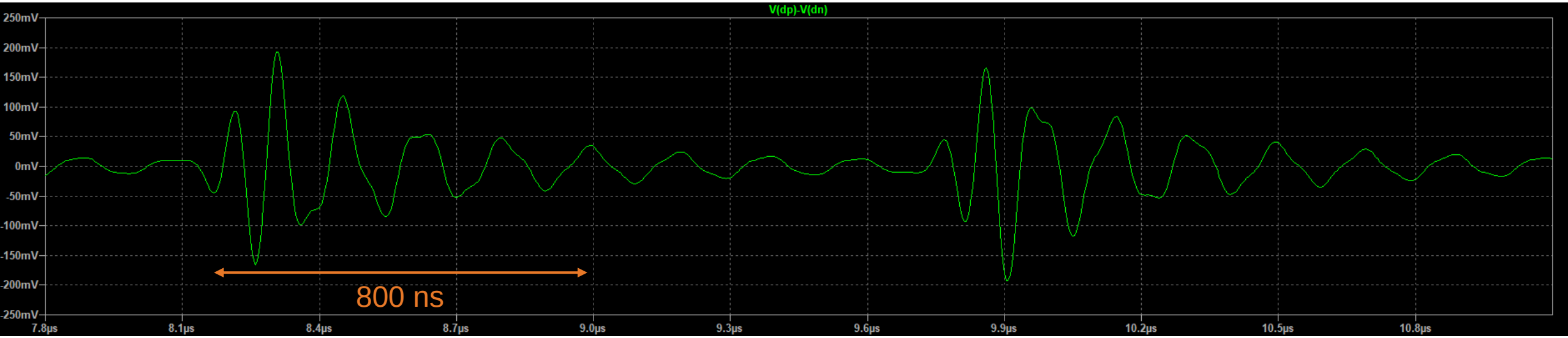






Eye
is 100%
closed

Well, not really… The droop spec implicitly constrains LFs



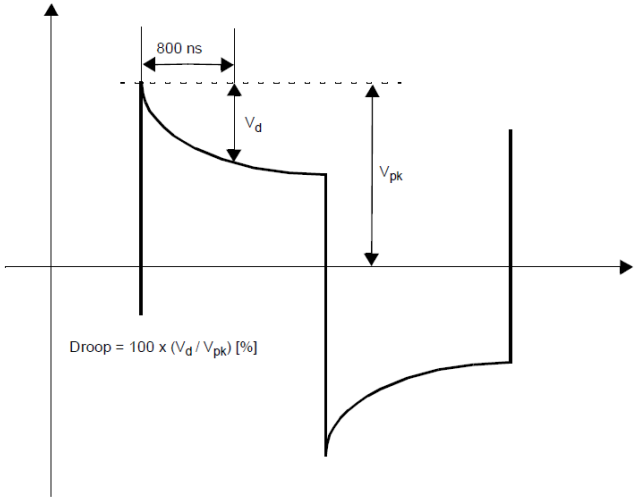
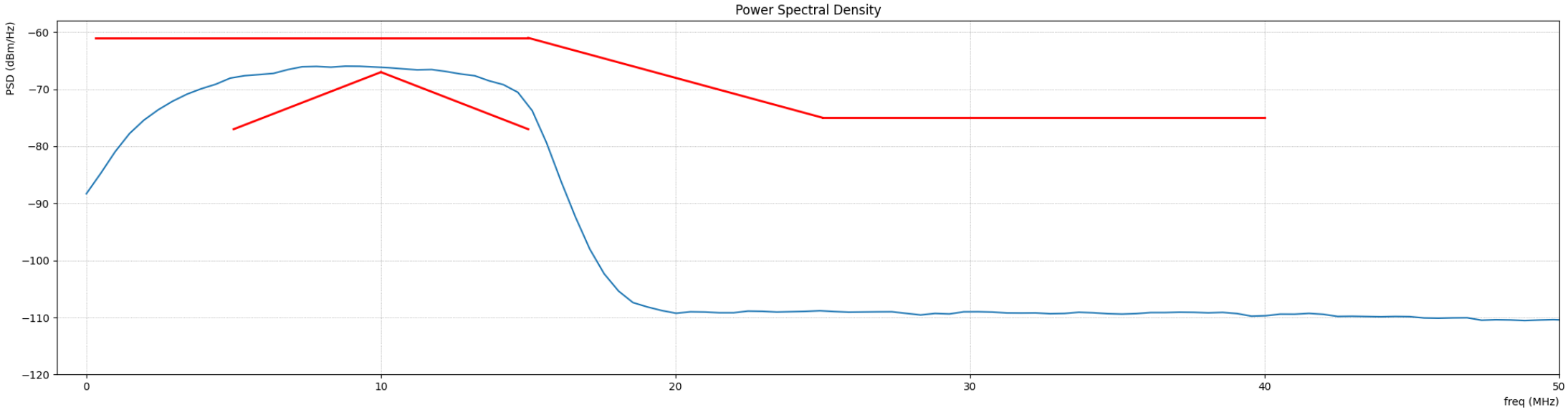
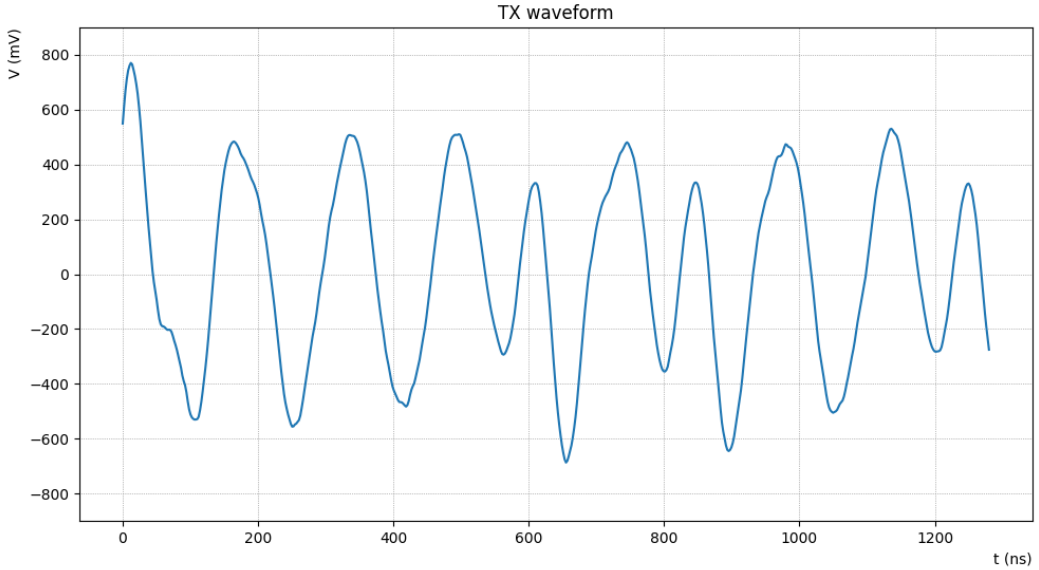


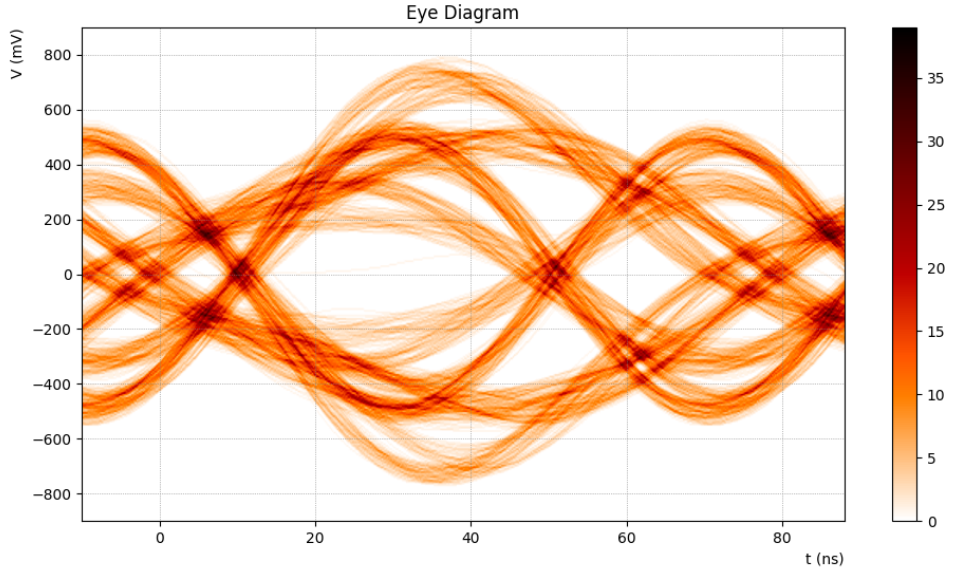
Figure 147–17—Transmitter output droop

- Using the droop pattern (3.2 µs square wave) there's basically no signal to measure!
- Target droop in 802.3cg is 30% max

“Worst” 802.3cg transmitter model that honors droop spec

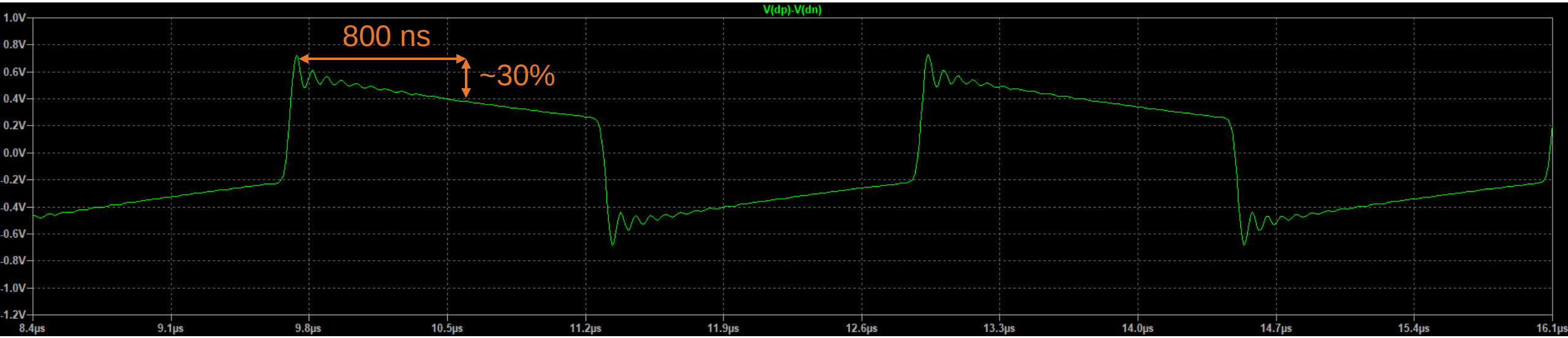






Still not
Very good!
But allowed

droop is now within the limits



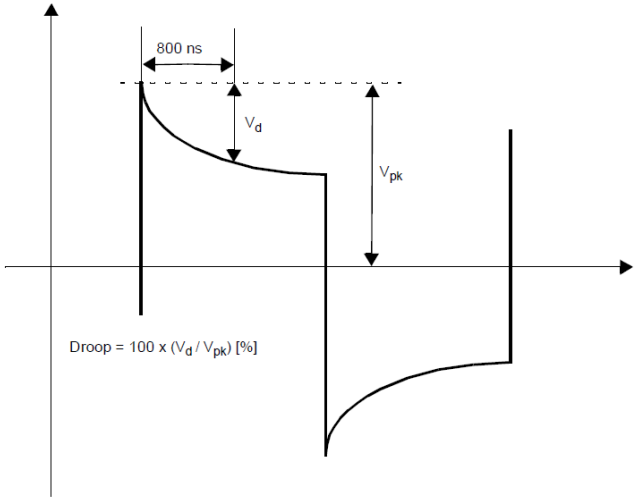


Figure 147–17—Transmitter output droop

- This model meets the 802.3cg droop specifications
- Eye diagram is still much worse than the typical case
  - a lot more ISI!

Conclusions

Conclusions

- The current PSD mask definition allows for transmitter implementations that produce very different eye diagrams
  - This should be considered when modeling / validating the mixing-segment
- Both the droop specification and the PSD specification contribute to the eye opening
  - Could we just extend the PSD limits in the lower frequencies to compensate for the droop?
    - Problem: we lack the required sensitivity. A small change in the PSD may significantly affect the droop
      - it is easier to keep the droop specs although…
      - Measuring the droop could be difficult/imprecise in practice, therefore it would be beneficial to avoid it
      - Changing the PSD pattern (which is low limited by the DME) could be an option
        - » more work is needed
- The transmitter models and tests are implemented in LTSpice / Python
  - Could be included in the SPE-MD simulations available on github
  - Some help would be appreciated

onsemi™

Intelligent Technology. Better Future.

Follow Us @onsemi

www.onsemi.com