Simulating Noise Environment

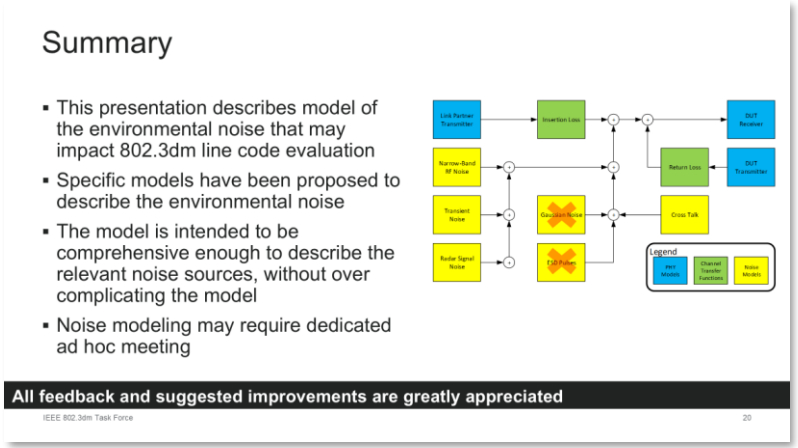
Contribution to 802.3dm Task Force

February 24, 2025

Ragnar Jonsson - Marvell

Introduction

- Specific noise models were proposed in [jonsson_3dm_01_07_15_24.pdf]
- The simDM simulation environment was introduced and then used in
  - [jonsson_3dm_01_12_19_24.pdf]
  - [jonsson_3dm_02a_12_19_24.pdf]
  - [jonsson_3dm_01b_01_20_25.pdf]
- This presentation discusses how the environmental noise models can be included in the simDM simulations



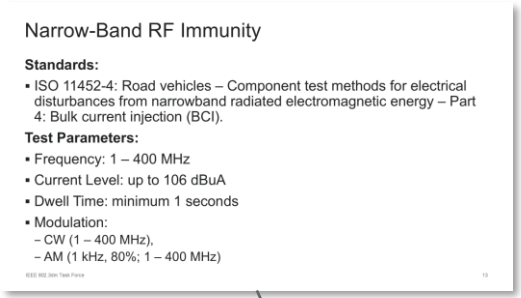
From https://www.ieee802.org/3/dm/public/0724/jonsson_3dm_01_07_15_24.pdf

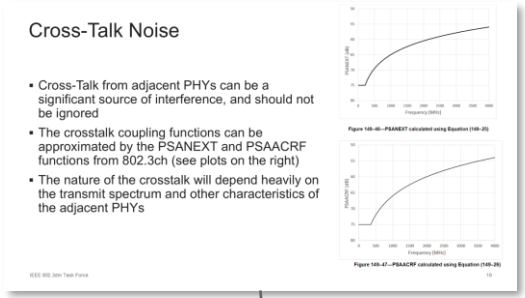
From https://www.ieee802.org/3/dm/public/adhoc/121924/jonsson_3dm_01_12_19_24.pdf

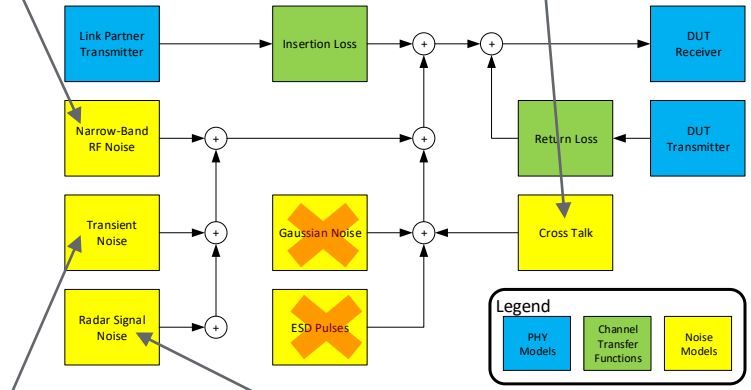
Environmental Noise

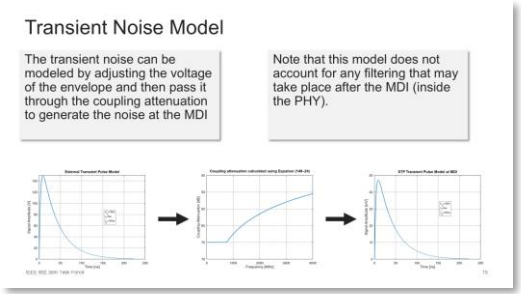
The presentation [jonsson_3dm_01_07_15_24.pdf] focused on four kinds of environmental noise:

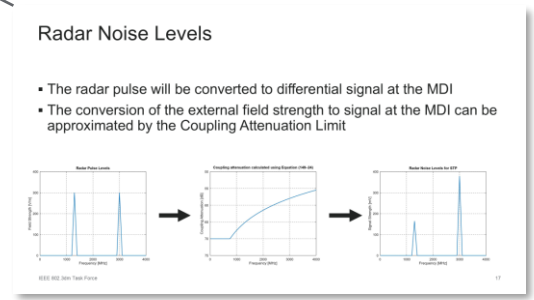
- Narrow-Band RFI
- Transient Noise
- Radar Noise
- Cross Talk







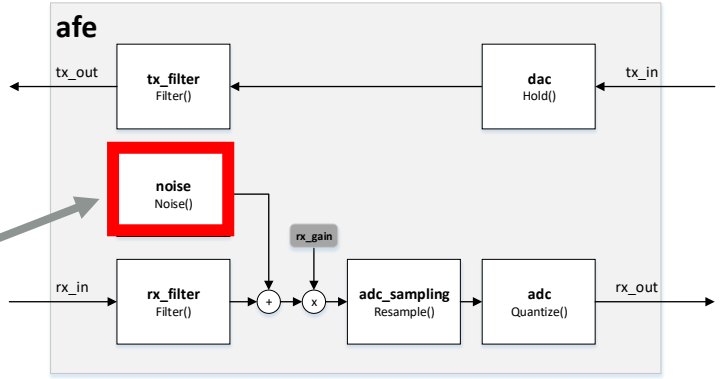


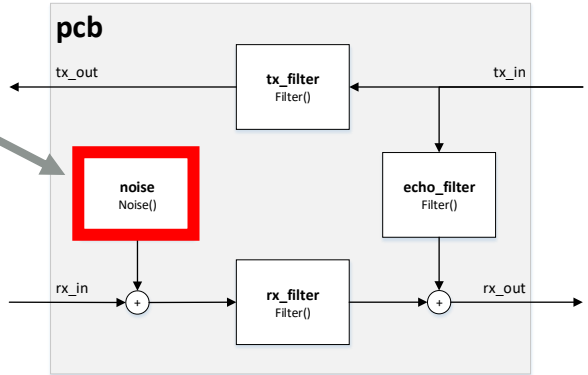


All figures above are from https://www.ieee802.org/3/dm/public/0724/jonsson_3dm_01_07_15_24.pdf

Hooks for Noise Injection in simDM

- The simDM simulation code has hooks to inject noise at the PCB input and in the AFE (see block diagrams)
- The noise injection point in the AFE is intended to model the AFE internal noise (e.g. Gaussian noise)
- The noise injection point in the PCB is intended to inject the environmental noise:
  - Narrow-Band RF Noise
  - Transient Noise
  - Radar Signal Noise
  - Cross Talk





Noise Hooks in Test Code

```
 99> %%% Configure the PCB impairments %%%
100> pcb_config.hybrid_echo = h_hybrid_echo; % Set hybrid echo for PCB
101> pcb_config.cutoff = pcb_cutoff;         % Set PCB cutoff frequency
102> pcb_config.noise = '0;';                % Set PCB noise to zero
```

- The Test1.m function in the simDM V1.1 code [1] shared in the January meeting has hooks for configuring the PCB noise
- In the Test1.m code the PCB noise is explicitly set to zero (see line 102 above)
- All that is needed to enable the environmental noise models is to add MATLAB code to generate the noise

[1] https://www.ieee802.org/3/dm/public/0125/simDM_V1_1.zip

Transient Pulse Envelope

It was suggested in [2] to use transient pulse envelope to shape a modulated tone to model Transient Noise

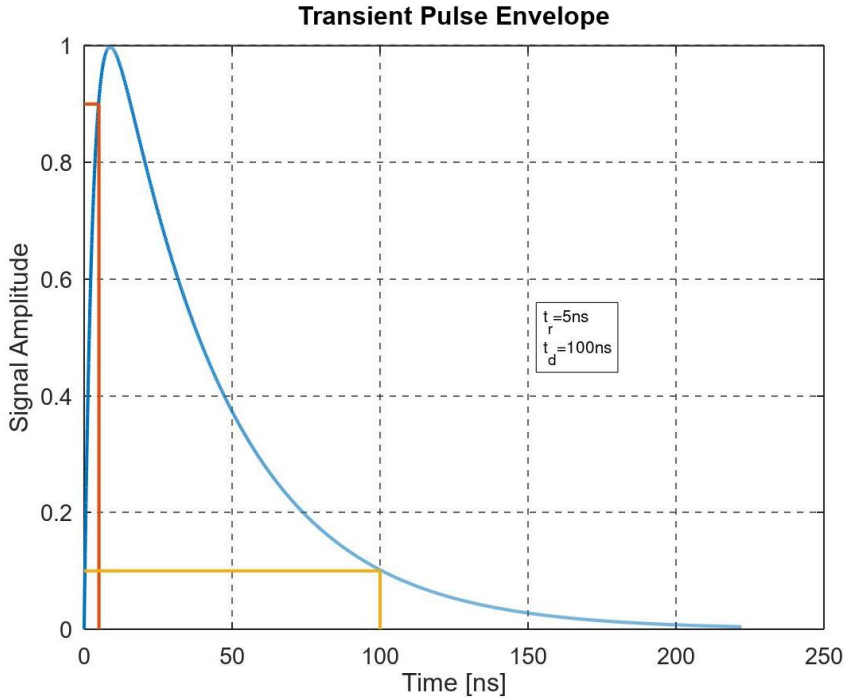
Such pulse envelope can be modeled with

$$f(t) = K \times (e^{-at} - e^{-bt})$$

where a, b, and K are chosen based on the desired rise and decay times.

In some cases, it is desirable to modulate the tone with a square wave

$$f(t) = \begin{cases} 1 & 0 \le t < t_d \\ 0 & else \end{cases}$$



New function was added to simDM to support such modulated pulse

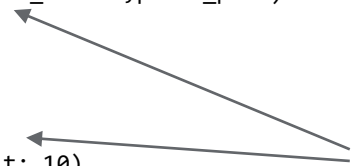
[2] https://www.ieee802.org/3/dm/public/0724/jonsson_3dm_01_07_15_24.pdf

```
function h = simDM_modulated_pulse(t, t_r, t_d, A, f_1, t_interval)
% Function to generate modulated pulse (transient noise) signal
% Arguments:
%   t - time vector
%   t_r - rise time constant in ns
%   t_d - decay time constant in ns
%   A - amplitude of the transient noise in mV
%   f_1 - frequency of the tone in MHz (default is 0)
%   t_interval - time interval for modulation in ms (default is new transient every call)
% Returns:
%   h - modulated pulse signal
% Zero t_r means that the pulse becaumes a square pulse
% If t_interval is not provided, a new transient is generated every call
% Usage:
%   h = simDM_modulated_pulse(t, t_r, t_d, A, f_1, t_interval)
% Example:
%   t = [0:1000]*1e9; % Time vector in ns
%   h = simDM_modulated_pulse(t, 10, 100, 100, 10);
%   plot(t, h);
%   xlabel('Time [ns]');
%   ylabel('Amplitude [mV]');
%   title('Modulated Pulse Signal');
%   grid on;
% This will plot modulated pulse with rise time of 10ns, decay time of 100ns,
% amplitude of 100mV, and tone frequency of 10MHz.
%
% This is simulation code provided to help with the development of
% IEEE 802.3dm.
%
% This code is provided for reference to allow independent evaluation
% of the accuracy and applicability of the simulation results shared in
% IEEE 802.3dm presentations by the author.
%
% Written by Ragnar Jonsson, affiliated with Marvell Technology, Inc.
% Version 1.1.1, February 24th, 2025
%
% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
% OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
% FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
% THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
% LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
% FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
% DEALINGS IN THE SOFTWARE.

    % Set default frequency if not provided
    if ((nargin < 5) || isempty(f_1))
        f_1 = 0;
    end

    % Set default time interval if not provided
    if ((nargin < 6) || isempty(t_interval))
        t_interval = t(1)*1e9 + 100; % Default to new transient 100ns into every call
    end

    % change argument units
    t_r = t_r * 1e-9; % Convert rise time to seconds
    t_d = t_d * 1e-9; % Convert decay time to seconds
    t_interval = t_interval * 1e-9; % Convert time interval to seconds
    A = A * 1e-3; % Convert amplitude to V
    f_1 = f_1 *1e6; % Convert frequency to MHz

    % Modulate time vector by the time interval
    t = mod(t, t_interval);

    % Generate the pulse shape
    if(t_r == 0)
        pulse = (t <= t_d); % Generate a square pulse
else
        % Calculate rise and decay constants
        a_r = log(0.1 * 2.45) / t_r;
        a_d = log(0.1 / 1.35) / t_d;

        % Generate the rising and decaying pulse shape
        h = ((1 - exp(a_r * t)) .* exp(a_d * t));
        pulse = h ./ max(h(:)); % Normalize the pulse
    end

    % Generate the tone
    tone = cos(2 * pi * t * f_1);

    % Combine pulse and tone to create transient noise
    transient = A * (pulse .* tone);
    h = transient; % Output the transient noise signal
end
```

Test4.m

- New test function Test4.m was added to the simDM code, that takes the environmental noise as an argument
- The new Test4.m function is identical to Test1.m, except for the addition of the new noise argument

```
function Test4(hdr_rate,pam_levels,cable_name,env_noise,pcb_cutoff,print_plot)
% Simulation of ACT transmit and receive signals - Test 4
%   Test4(hdr_rate,pam_levels,cable_name,env_noise,pcb_cutoff,print_plot)
% Function arguments:
%   hdr_rate   - Data rate in Gbps (default: 2.5)
%   pam_levels - Number of PAM levels (default: 4)
%   cable_name - Cable model name (default: 'good')
%   env_noise  - Environmental noise (default: '0')
%   pcb_cutoff - PCB cutoff frequency in MHz (default: 10)
%   print_plot - Flag to save plots (default: 0)
```



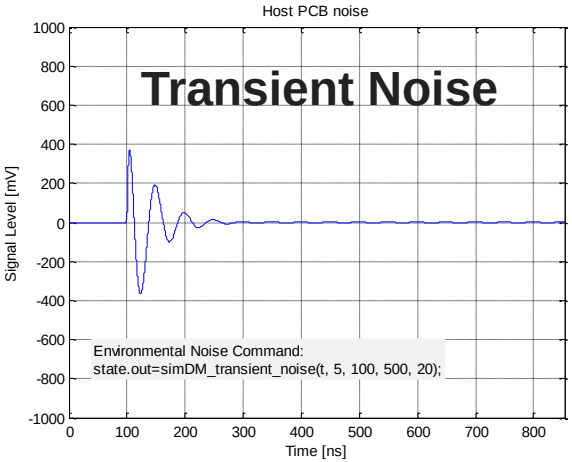
New argument

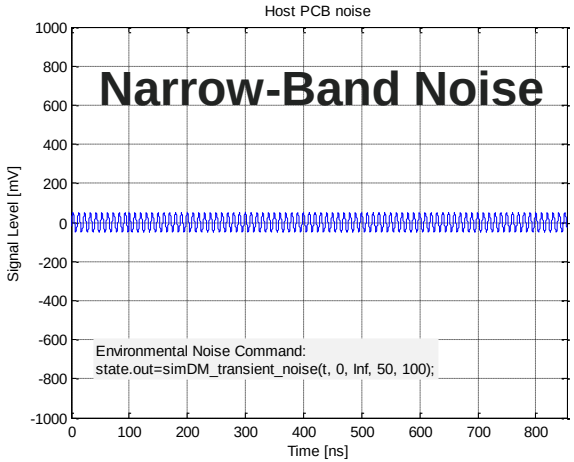
Examples

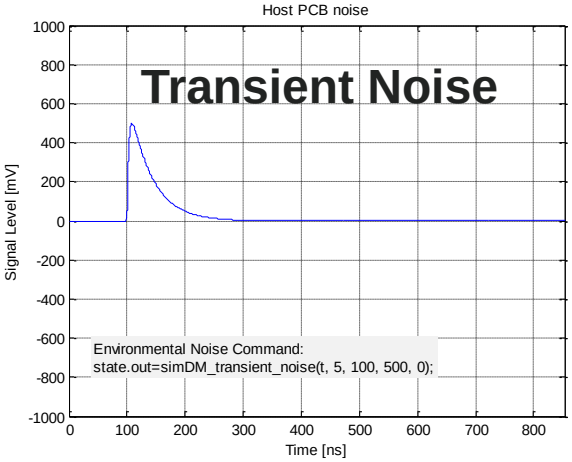
The new functions can now be used to generate three different kinds of noise:

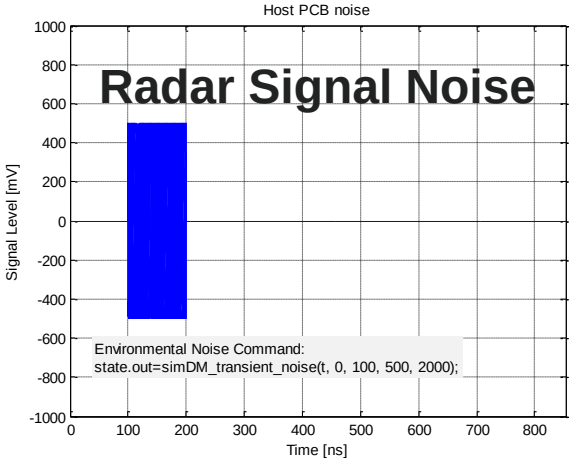
- Narrow-Band RF Noise
- Transient Noise
- Radar Signal Noise

Cros-talk noise is not yet supported in the simDM









Summary

- The simDM simulation code has been updated to more easily support environmental noise
- The supported environmental noise types are
  - Narrow-Band RF Noise
  - Transient Noise
  - Radar Signal Noise
- Cross-talk noise has not yet been added to the simDM
- Separate presentations will use the new noise models to evaluate the performance of different PHY candidates in the presence of environmental noise

The updated MATLAB code will be made available to the 802.3dm Task Force

Essential technology, done right™