

How to Identify Material Models for PCB & PKG Interconnects With Simbeor SDK

Introduction

Meaningful interconnect design and compliance analysis must start with the identification of broadband dielectric and conductor roughness models. Such models are usually not available from manufacturers and the model identification is the most important element of the successful interconnect design for link paths with 10 Gbps and higher data rates. Analysis of interconnects without such models may be simply not accurate and misleading. Support for material model identification is available in Simbeor software since 2009 [1], [2] and advanced identification with Generalized Model S-parameters (GMS-parameters) was introduced back in 2010 [2]-[3]. Though, the process of the model identification is relatively tedious and requires multiple steps as demonstrated in demo-videos #2011_04 and #2015_02 at <https://www.simberian.com/ScreenCasts.php?view=list> Simbeor SDK facilitate the process by providing API and scripted examples in C/Python/Matlab for dielectric and conductor roughness model identification. The process can be easily adjusted for automation of the material model identification. This article explains how it works with practical examples provided mostly in Matlab script in AdvMaterialKit (matching examples in C and Python are provided in test_gms.cpp test_gms.py files also available in Simbeor SDK).

Causal Dielectric and Conductor Roughness Models

The largest part of interconnects can be formally defined and simulated as transmission line segments. **Accuracy of transmission line models (impedance, insertion loss and dispersion) is mostly defined by availability of broadband dielectric and conductor roughness models.** Let's first explore the most common and important causal dielectric and conductor roughness models. It is important for understanding the process and the results of the identification. Dk and LT extracted at one frequency point describe a frequency-continuous dielectric model that can be used over very wide frequency bandwidth and, usually, extrapolate dielectric properties beyond the bandwidth of the original measured S-parameters. SR and RF identified for a frequency-continuous conductor roughness model can be also used over and beyond the bandwidth of the original measured S-parameters. This is the main advantage of the broadband models identification, comparing to point by point models extracted with resonator techniques. The models are also portable – can be used in different EDA simulation tools.

Dielectric Models

Wideband Debye (aka Djordjevic-Sarkar or Swensson-Dermer) is the most widely used causal dielectric model for accurate analysis of PCB and packaging interconnects. This model has been developed and described independently by authors of [4] and [5]. The authors of [4] and [5] assumed that composite materials used in PCBs, PKGs and MCMs do not have distinct relaxation or polarization frequencies, but has rather a continuous spectrum of the relaxation frequencies over a wide frequency band. They showed that the model is in good agreement with the measurements of dielectric constant

for composite types of dielectrics and can be defined with just one measurement of dielectric constant and loss tangent. See more on that in the [“Material World...”](#) tutorial. The accuracy of the model was confirmed by dozens of research papers and the model is available in practically all EDA tools (though how the model is defined may be different). Note that the wideband Debye model was implemented in the very first version of Simbeor software released in 2007.

Relative dielectric constant of the wideband Debye model is defined as follows:

$$\varepsilon_{wd}(f) = \varepsilon(\infty) + \Delta\varepsilon \cdot F_d(f)$$

where

$$F_d(f) = \frac{1}{[m_2 - m_1] \cdot \ln(10)} \cdot \ln \left[\frac{f_2 + if}{f_1 + if} \right]$$

i is imaginary unit $i^2 = -1$, and $f_1 = 10^{m_1}$ Hz, and $f_2 = 10^{m_2}$ [Hz] define frequency range where dielectric constant drops from approximately DC value $\varepsilon(\infty) + \Delta\varepsilon$ almost to the value at infinity $\varepsilon(\infty)$. Values of $\varepsilon(\infty)$ and $\Delta\varepsilon$ can be defined **with only one measurement of dielectric constant ε_r and loss tangent $\tan \delta$ at a frequency f_0 as follows:**

$$\Delta\varepsilon = - \frac{\tan \delta \cdot \varepsilon_r \cdot \ln(10) \cdot [m_2 - m_1]}{\operatorname{Im} \left(\ln \left[\frac{f_2 + if_0}{f_1 + if_0} \right] \right)}$$

$$\varepsilon(\infty) = \varepsilon_r \cdot \left(1 + \tan \delta \cdot \frac{\operatorname{Re} \left(\ln \left[\frac{f_2 + if_0}{f_1 + if_0} \right] \right)}{\operatorname{Im} \left(\ln \left[\frac{f_2 + if_0}{f_1 + if_0} \right] \right)} \right)$$

Authors of [4] suggested to set $m_1 = 4$ and $m_2 = 12$ for the laminate materials. In that case, the real part of the dielectric constant gradually decreases in the frequency range from 10 KHz to 1 THz. Though, the imaginary part of the dielectric constant is almost constant over the frequency range from 0.1 MHz to 100 GHz. That corresponds to the following frequencies

$$f_{low} = 10^{m_1+1} [Hz], \quad m_1 = \log(f_{low}) - 1,$$

$$f_{high} = 10^{m_2-1} [Hz], \quad m_2 = \log(f_{high}) + 1$$

They are used in Simbeor to define the frequency band for the wideband Debye model. The loss tangent is slightly rising approximately over the frequency band defined by f_{low} and f_{high} . It corresponds to observed behavior of a typical PCB laminate material (see data from [2]-[3] for example). So, knowing Dk, LT, measurement frequency and f_{low} and f_{high} , one can easily get frequency dependent Dk and LT by using the formulas provided above in Excel, Matlab, Mathcad or other software, or use the corresponding parameters directly in any EDA simulation software.

Here how the original Djordjevic-Sarkar model is represented by variables in Simbeor and Simbeor SDK:

ε_r is Relative Permittivity (Dk) in GUI and dRelativePermittivity in SDK – it is actually the real part of relative permittivity, aka Dielectric Constant or Dk at the Measurement Frequency.

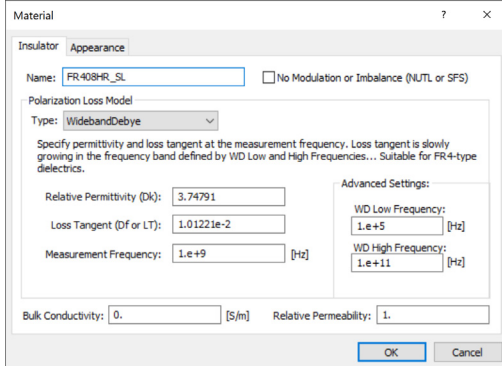
$\tan \delta$ is Loss Tangent (Df or LT) in GUI and dLossTangent in SDK – it is negative ratio of imaginary and real part of the complex relative permittivity, aka Loss Tangent or LT at the Measurement Frequency.

f_0 is the Measurement frequency in GUI and dMeasurementFrequency in SDK – all in Hz.

f_{low} is WD Low Frequency in GUI and dWDLowFrequency in SDK with default value 1.0e5 Hz, that corresponds to low frequency with $m1=4$ in the original Djordjevic-Sarkar model.

f_{high} is WD High Frequency in GUI and dWDHighFrequency in SDK with default value 1.0e11 Hz, that corresponds to high frequency with $m2=12$ in the original Djordjevic-Sarkar model.

Below is an example of the model definition in Simbeor THz (example for CMP-28 case):



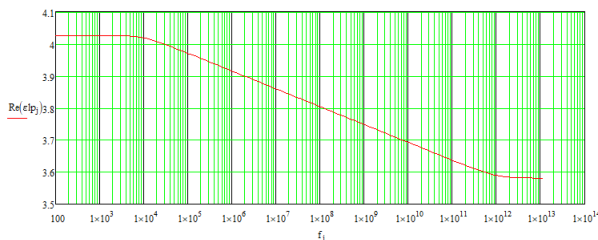
It corresponds to the identified results in Matlab structure of AdvMaterialKit Identified.Dielectrics(1):

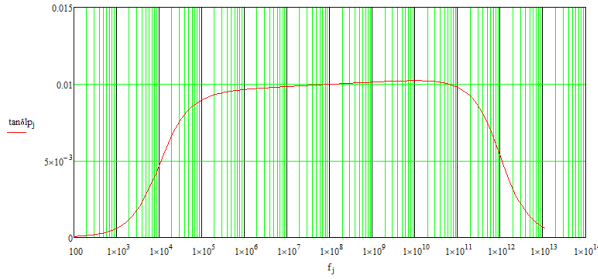
```
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
Dk: 3.7479
LT: 0.0101
Fmq: 1.0000e+09
```

Or similar parameters in SimbeorSDK_DielectricWidebandDebye C-style structure (same fields in Python):

```
dRelativePermittivity=3.7479;
dLossTangent=1.01221e-2;
dMeasurementFrequency=1.0e9;
dWDLowFrequency=1.0e5;
dWDHighFrequency=1.0e11;
```

That corresponds to $\epsilon_r = 3.7479$, $\tan \delta = 0.0101221$, $f_0 = 10^9$, $m1=4$, $m2=12$ and the following dispersion plots for real part of the relative permittivity (top graph) and loss tangent (bottom graph):





Note that f_{low} may be also used as an additional model fitting parameter during the optimization. Though, it is typically fixed. If Wideband Debye model is not acceptable for some reasons, universal Multipole Debye, multipole Debye-Lorentz and multipole Havriliak-Negami models are also available in Simbeor and SDK – any possible interconnect dielectric can be described by one of those models. See more at https://kb.simberian.com/browse_item.php?id=768, Simbeor User Manual and models definition in Simbeor SDK documentation.

Conductor Roughness Models

To simulate effect of conductor roughness, causal Huray-Bracken (derived from the original Huray’s snowball model by Eric Bracken in [6]) and causal Modified Hammerstad (derived from the Modified Hammerstad model by Vladimir Dmitriev-Zdorov in [7]) conductor roughness models can be effectively used. Expression for the multilevel conductor surface impedance correction coefficient based on the Huray’s snowball model and Bracken modification [6] can be written in the unified multilevel additive form [8] as follows:

$$K_{sr} = 1 + \sum_k (RF_k - 1) \cdot F_{hb}(r_k, \delta_s)$$

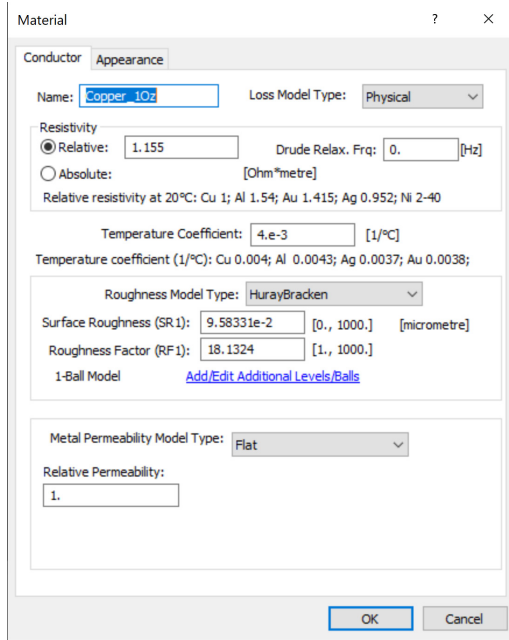
$$F_{hb}(r_k, \delta_s) = \left(1 + (1 - i) \frac{\delta_s}{2r_k} \right)^{-1}$$

$$\delta_s = (\pi \cdot f \cdot \mu \cdot \sigma)^{-1/2}$$

δ_s is the frequency-dependent skin depth. With the number of levels $k=1$, this model has 2 parameters: ball radius r and roughness factor RF . Both are usually not known for commonly used copper foils (copper is roughened separately by the foil and PCB manufacturers). Model parameter r [m] is called in Simbeor Surface Roughness SR and is defined in micrometres. Roughness factor can be converted into Hall-Huray surface ratio sr and back as follows:

$$sr = \frac{2}{3}(RF - 1), \quad RF = 1 + \frac{3}{2}sr$$

Example of Huray-Bracken model definition in Simbeor THz for CMP-28 case is as follows (Conductor -> Properties)



Corresponding Matlab structure with parameters identified in AdvMaterialKit have the following form:

```
Name: 'Copper_10z'
RR: 1.1550
Roughness: [1x1 struct]
Model: 'HurayBracken'
SR: 0.0958
RF: 18.1324
```

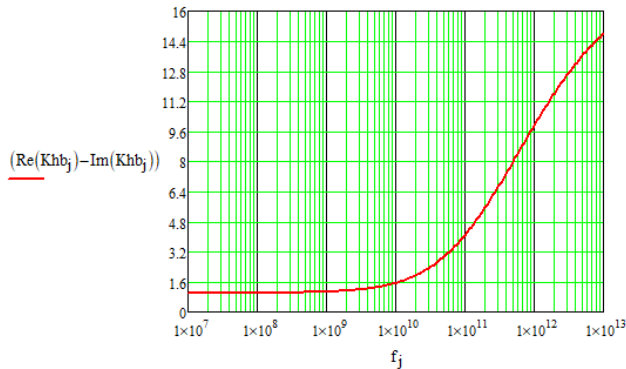
Or in C and Python:

```
SimbeorSDK_Conductor::dRelativeResistivity=1.1550;
SimbeorSDK_Roughness::roughnessModel='HurayBracken'
SimbeorSDK_Roughness::SR=0.0958
SimbeorSDK_Roughness::RF=18.1324
```

This model corresponds to increase of surface impedance by the value of K_{sr} as follows:

$$Z_{rough} = \frac{K_{sr}}{\sigma \cdot \delta_s} \cdot (1 + i)$$

That corresponds to increase of losses with frequency due to additional roughness losses by the value $\text{Re}(K_{sr}) - \text{Im}(K_{sr})$ as illustrated on the following plot for the example provided above (SR=0.0958 um, RF=18.1324, same as for the regular Huray’s snowball model):



Another, practically useful for interconnects, surface impedance correction coefficient is causal version of the Modified Hammerstad model [7] that can be expressed in unified multilevel multiplicative form [8] as follows:

$$K_{sr} = \prod_k \left[1 + (RF_k - 1) \cdot (F_{hb}(\Delta_k, \delta_s)) \right]$$

$$F_{hb}(\Delta_k, \delta_s) = \frac{1}{\pi} \cdot \left(\ln \left[1 + \frac{2\sqrt{ix_k}}{1 + ix_k} \right] + 2 \arctan \left[\sqrt{ix_k} \right] \right)$$

$$x_k = 1.4 \left(\frac{\Delta_k}{\delta_s} \right)^2$$

For the one-level case it has also two parameters: Δ [m] or surface roughness parameter (SR defined in micrometres in Simbeor interface) and roughness factor RF (maximal possible increase of losses due to roughness). Causal Hammerstad model predict exactly the same increase in losses as the non-causal Modified Hammerstad model with the same parameters. It also corresponds to the original Hammerstad model if RF is fixed to 2 (very restrictive). The causal models predict larger growth of the inductive part of surface impedance that is usually observed through measurements.

Material Model Identification

Manufacturers of dielectrics usually provide dielectric parameters at a few discrete frequency points. Possible ways to use those points were discussed at [“Material World...”](#) tutorial. The simplest way is to use one of the points to define the Wideband Debye model – it requires just one point as described in the previous chapter. The constructed model becomes useful over extremely broad frequency range. Though it may not correlate with the other points provided by manufacturer, especially if the points are measured with different techniques. Overall accuracy of Wideband Debye model constructed with the data from laminate manufacturer is usually acceptable and may require small correction only due to anisotropic properties of the laminate for instance.

Things are not so good for the copper roughness models. Manufacturers of copper laminates typically do not have parameters for the electrical roughness models (such as described in the previous chapter) at all. Parameters in datasheets are usable for mechanical purpose, but not for the electrical characterization. RMS peak-to-valley value R_q can sometime be used for reverse treatment foils as parameter Δ in the Modified and Causal Hammerstad models. But, the roughness factor has to be identified. Thus, meaningful interconnect design and compliance analysis must start with the identification or validation of dielectric and conductor roughness models over the frequency band of interest. **Availability of accurate broadband material models is the most important element for design success.** Validation or identification of dielectric and conductor models can be done with the reflection-

less generalized modal S-parameters (GMS-parameters) as shown in [2]-[3]. Main steps of the process are described in this section for better understanding of the process. Possible methods for separation of dielectric and conductor roughness loss and dispersion effects are also described and demonstrated with practical examples. The user has to decide which loss separation method to use (if any).

Dielectric and conductor roughness models identification can be done by matching measured and computed GMS-parameters for a transmission line segment. S-parameters for two line segments with different length and substantially identical cross-sections and transitions to probes or connectors must be measured first to compute measured GMS-parameters. Before proceeding with the identification of the material models, it is important to verify all dimensions of the test structures on the board. In particular, cross-sections of the transmission lines and length difference between two line pairs have to be accurately measured. Next, quality of measured transmission line S-parameters has to be estimated and TDR used to verify consistency of the test fixtures.

The basic procedure for the dielectric and conductors surface roughness models identification is illustrated in Fig. 1 can be performed as follows (the details of the procedure are described in [2],[3]):

(1) Measure scattering parameters (S-parameters) for at least two transmission line segments of different length (L_1 and L_2) and substantially identical cross-section and conductor roughness profile filled with dielectric with known dielectric model.

(2) Compute generalized modal S-parameters of the transmission line segment difference $L = |L_2 - L_1|$ from the measured S-parameters.

(3) Compute GMS-parameters of line segment difference L :

(3a) Guess dielectric or conductor surface roughness model and model parameters – use spreadsheet data as the starting point.

(3b) Compute generalized modal S-parameter of line segment difference L by solving Maxwell's equations for line cross-section with the broadband material models.

(4) Compare GMS-parameters and adjust model to minimize the difference or output the identified model.

(4a) Compare the measured and computed generalized modal S-parameters - compute metric of difference of two complex GMS-parameters.

(4b) If the difference is larger than a threshold, change model parameters (or model type) and repeat steps (3b)-(4).

(4c) If the difference is less or equal to threshold, the dielectric or conductor roughness model is found.

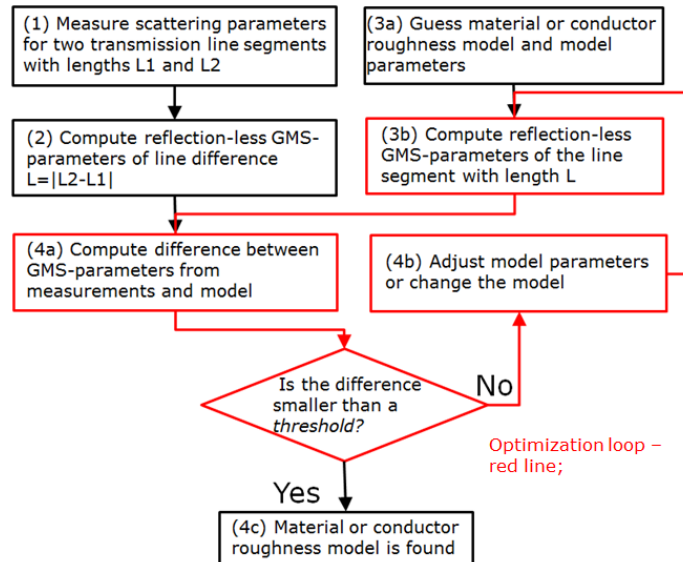


Fig. 1. Dielectric material or conductor surface roughness model identification procedure implemented in Simbeor THz and SDK.

This procedure, including the model parameters optimization, is implemented Simbeor software and further automated in Matlab, C and Python scripts provided with Simbeor SDK. **The key in this approach is availability of algorithms for analysis of transmission lines that supports the frequency-continuous material models (1-4) in step (3b) of the algorithm shown in Fig. 1.**

It is known that the conductor roughness effect causes signal degradation (losses and dispersion) that are similar to the signal degradation caused by dielectrics. **Thus, it is important to separate the effects of losses and dispersion properly between the conductor roughness and dielectric models, or understand the consequences of not doing such separation.** The consequences of not separating the losses are discussed in [this presentation](#). There are four scenarios to build the conductor surface roughness model without and with separation of the loss and dispersion effects between the dielectric and conductor surface roughness models [3]:

- 1) **No roughness losses.** Optimize dielectric model to fit measured and modelled GMS-parameters following the procedure in Fig. 1 and do not use any additional conductor roughness model. The dielectric model will include effect of conductor surface roughness. Such model may be suitable for the analysis of a particular transmission line and has to be rebuilt if strip width or line type is changed. This combined model may be acceptable in cases of high-loss dielectrics when the effect of conductor roughness is minimal.
- 2) **Fix loss tangent and identify roughness model only.** Define dielectric constant and loss tangent with the data available from the dielectric manufacturer and then identify a roughness model (a roughness correction coefficient) with GMS-parameters following the procedure in Fig. 1. This approach works well if a manufacturer has reliable procedure to identify the dielectric properties (most of them do). Wideband Debye model can be defined with just one value of dielectric constant and loss tangent specified at one frequency point. This is the simplest way to identify the conductor roughness model for low and extremely low loss dielectrics.
- 3) **Identify dielectric loss tangent and conductor models separately over different bandwidths.** Optionally, identify conductor roughness at frequencies below onset of skin-effect (50 MHz for instance). Next, identify dielectric loss tangent at low frequencies (below 0.5-1 GHz) where the effect of the conductor roughness is negligible or not substantial. Next, identify conductor

roughness model at frequencies above 2-3 GHz. This is the simplest way to separate loss and dispersion effects in conductor surface roughness and dielectric models. Additional iteration can be added to fine tune the dielectric and roughness losses.

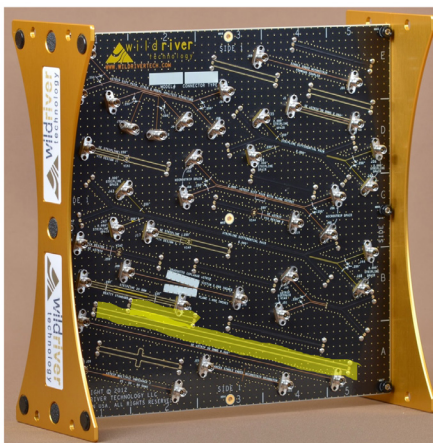
- 4) **Identify dielectric loss tangent and conductor roughness models simultaneously over same bandwidth.** This scenario is also supported in Simbeor. Though the procedure is the most complicated and may lead to multiple possibilities (ambiguity).

All those scenarios are implemented and supported by Matlab scripts in AdvMaterialKit and by C/Python as well. Instructions on how to define the identification parameters for a particular scenario are provided in the next chapters.

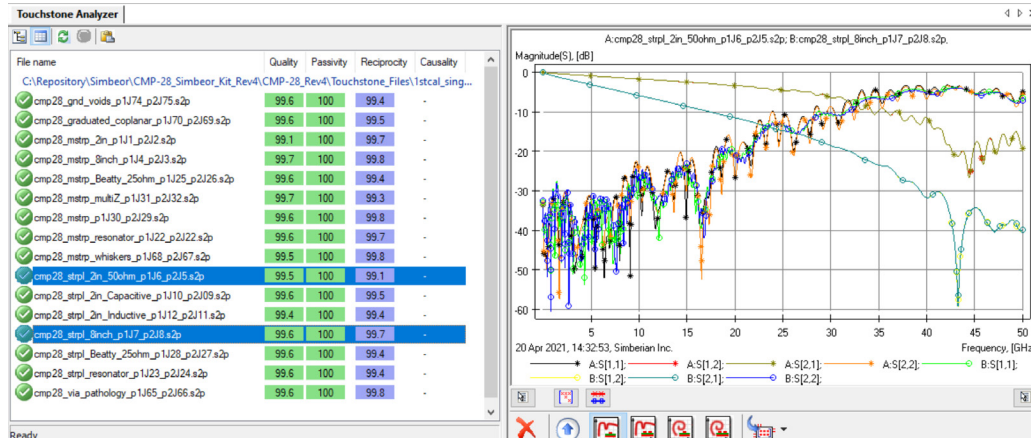
Overall, the material identification procedure described here is the simplest possible. It needs measurements for 2 t-lines with any geometry of cross-section and transitions. No extraction of propagation constants (Γ) from measured data is required. The extraction of Γ is difficult and error-prone. Also, no de-embedding of connectors and launches is required. De-embedding of PCB structures is usually difficult or even impossible due to inhomogeneity of dielectrics and manufacturing variations. The approach needs the simplest numerical model - only propagation constant has to be computed for a given cross-section and with the material models to identify. No 3D electromagnetic models of the transitions are required. Procedure with S-parameters has minimal number of smooth complex functions to match during the identification process. Specifically, one S-parameter for single and two S-parameters for differential lines have to be matched. All reflection and modal transformation parameters are exactly zeroes. The loss separation by the bandwidth is natural and also very simple. **Identified models are frequency-continuous and models described in the previous section are not restricted to the frequency band used in the identification process – they are naturally extendable above the upper and below the lower frequencies.**

Example 1: Material Model Identification On CMP-28

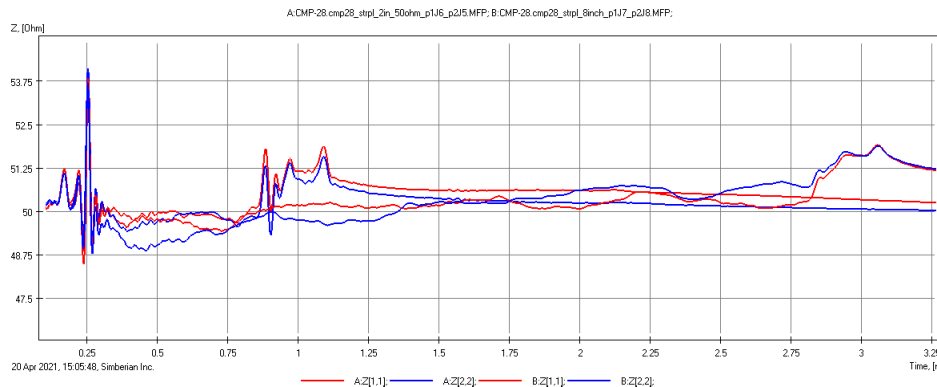
To start the material identification, one needs measured S-parameters for 2 line segments with substantially identical cross-sections, cross-section of the transmission lines (dimensions) and preliminary data to define starting points for dielectric model (Wideband Debye) and for conductor surface roughness (Huray-Bracken or Causal Hammerstad). As the first example of the material model identification we will use CMP-28 validation platform with all measured and board description data provided by Wild River Technology <https://www.wildrivertech.com/> Complete CMP-28 description is available at #2014_06 at <https://www.simberian.com/TechnicalPresentations.php> Two stripline segments highlighted on the picture below are used for the identification here.



The first step is to measure S-parameters for 2 line segments with substantially identical cross-sections and pre-qualify them for the material identification. The pre-qualification procedure is usually done manually at the location where the measurements done. Though, it can be also automated with Simbeor SDK – all functions for the quality evaluation and computation of TDR are available through the API provided in SDK. Here we use the manual process of the quality evaluation in Touchstone Analyzer as shown below (here is how to do it https://kb.simberian.com/browse_item.php?id=777):



As we can see, the final quality metrics for 2 segments of transmission line are excellent (Quality is over 99%). Pre-qualification must also include evaluation of the differences in TDRs of 2 line segments. Specifically, observed line impedance and connector or probes + launch discontinuities should be consistent in both segments. For extraction up to 30 GHz, the difference in TDR should not exceed 2-3 Ohm – see app note #2010_03 at <https://www.simberian.com/AppNotes.php> on the sensitivity of GMS-parameters to the manufacturing variations. TDR plot for 2 stripline segments on CMP-28 (one of the version of it from Simbeor CMP-28 Kit) is shown on the following graph:



The consistency of the launches and impedance deviation are sufficient for the material identification with GMS-parameters extracted up to 30 GHz in this case. Though, the identified material models will be usable well above of that frequency. Note that this is just a version of CMP-28 validation platform from Simbeor CMP-28 Kit (available on request). Newer versions of CMP-28 have much better consistency of the test fixtures that allows GMS-parameters extraction well above 30 GHz (in some cases up to 50 GHz).

Strictly speaking, the data preparation should include the cross-sectioning step. In this case the cross-section measurements are done by PCB manufacturer with the data used in the scripts below.

The second step is to describe the data for the identification (S-parameter files and cross-section). Here is how to do it with Matlab data structure. When Simbeor SDK is installed (see instructions at https://kb.simberian.com/browse_item.php?id=985), S-parameter files for 2 and 8 inch segments of stripline from CMP-28 platform will be located at the “..\Simbeor SDK Examples\touchstones” subdirectory (one can dig them from CMP-28 Kit as well). To run the model identification scripts, either define the path to that directory or create and copy the files into directory “C:\work\Touchstones”, to use the identification script as is. Here is the complete description of the data for CMP-28 using JSON format for the material and stackup definitions (file defineMaterialIdentificationCMP28_JSON.m, order of the structures and data does not matter and may be altered for better understanding):

```

prj.Format = 'JSON'; %STACKUP AND MATERIALS IN JSON FORMAT
prj.ProjectName = 'CMP-28'; %PROJECT NAME TO HOLD ALL MODELS AND MEASURED DATA
prj.LengthUnit = 'mil';

diel = 'FR408HR_SL'; %DIELECTRIC TO IDENTIFY
cond = 'Copper_10z'; %CONDUCTOR TO IDENTIFY

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATERIALS AND STACKUP FOR JSON TRANSFER
prj.Materials.ListOfMaterials = {
    struct("Type", "Insulator",
        "Name", "Air", "RelativePermittivity", 1.0, "LossTangent", 0.0), ...
    struct("Type", "Insulator", "Name", diel, "PolarizationLossModel",
        "WidebandDebye", ...
        "RelativePermittivity", 3.6, "LossTangent", 0.0117, "MeasurementFrequency",
        1.0e9), ... %starting point for Dk and LT
    struct("Type", "Conductor", "Name", cond, "RelativeResistivity", 1, ... %starting
    point for conductor relative bulk resistivity
        "Roughness", struct("RoughnessModelType", "HurayBracken", "SR0", 0.001, "RF0",
        2.0)) ... %starting point for roughness identification
};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prj.StackUp.LengthUnit = prj.LengthUnit;
prj.StackUp.Layers = { ...
    struct("Type", "Plane", "Name", "Plane1", "Thickness", 1.3, "PourMaterialName",
    cond, "FillInsulatorName", diel)...
    struct("Type", "Medium", "Thickness", 12.0, "FillInsulatorName", diel), ...
    struct("Type", "Signal", "Name", "Signal1", "Thickness", 1.3,
    "FillInsulatorName", diel, "DefaultConductorName", cond), ...
    struct("Type", "Medium", "Thickness", 10.5, "FillInsulatorName", diel), ...
    struct("Type", "Plane", "Name", "Plane2", "Thickness", 1.3, "PourMaterialName",
    cond, "FillInsulatorName", diel)...
};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%COMMON SWEEP FOR GMS FROM MEASUREMENTS, MODEL AND OPTIMIZATION
%unit default for frequency sweep is GHz
prj.GMS.FrqSweep.Start = 0.01; /*GHz*/
prj.GMS.FrqSweep.Stop = 30.0; /*GHz*/
prj.GMS.FrqSweep.Count = 3000; /*max number of poins for the adaptive sweep*/
prj.GMS.FrqSweep.AdaptivityTolerance = 0.1; /*defines sensitivity of the adaptive
sweeper*/

prj.GMS.Measured.FileLocation = 'C:\work\Touchstones';
prj.GMS.Measured.ShortSegmentFile = 'cmp28_strpl_2in_50ohm_p1J6_p2J5.s2p';
prj.GMS.Measured.LongSegmentFile = 'cmp28_strpl_8inch_p1J7_p2J8.s2p';
prj.GMS.Measured.LengthDifference = 6000; %mil
  
```

```

prj.GMS.Measured.DiffName = 'Diff_6in'; %name of circuit with extracted GMS -
parameters to fit(difference)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

prj.GMS.Model.TLine.Width = 10.5; /* Strip width in the middle of the layer =
0.5*(w_top+w_bottom), [meter]*/
prj.GMS.Model.TLine.StripShape = 'Rectangular';
prj.GMS.Model.TLine.EtchFactor = 0.0; /* Etch factor = 0.5*(w_top-w_bottom)/t1 */
prj.GMS.Model.TLine.Clearance = 0.0; /* Clearance to plane metal for coplanar traces
if non-zero, if zero, the strip is regular */
prj.GMS.Model.TLine.LayerName = 'Signal1'; /*layer with the strip in the project
specified by the model name*/
prj.GMS.Model.TLine.Distance = 0.0; %distance between traces in the middle of the layer

prj.GMS.Model.TLineName = 'Strip'; %cross - section model
prj.GMS.Model.TLineSegmentName = 'Seg_6in'; %stripline segment

%FREQUENCY SWEEP FOR THE FINAL ANALYSIS - TO SEE THE MODEL EXTENSION
prj.GMS.Model.FrqSweep.Start = 0.01; /*GHz*/
prj.GMS.Model.FrqSweep.Stop = 50.0; /*GHz*/
prj.GMS.Model.FrqSweep.Count = 5000; /*max number of poins for the adaptive sweep*/
prj.GMS.Model.FrqSweep.AdaptivityTolerance = 0.01; /*defines sensitivity of the
adaptive sweeper*/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

It looks complicated at first glance, but the data structure is relatively straightforward with self-descriptive names of the variables. Let’s walk through some important details.

First at the project level, the following parameters are defined:

```

prj.Format = 'JSON'; %STACKUP AND MATERIALS IN JSON FORMAT
prj.ProjectName = 'CMP-28'; %PROJECT NAME TO HOLD ALL MODELS AND MEASURED DATA
prj.LengthUnit = 'mil';

```

That indicates that Materials and StackUp will be described in JSON format, all data will be placed into Simbeor Project with name CMP-28 (can be loaded into Simbeor THz for verification purpose) and all lengths will be provided in mils.

Next, structures Materials, StackUp and GMS in structure prj define all data necessary for the identification.

Structure Materials defines all necessary dielectric and conductor models with fixed or preliminary parameters of the material models. For CMP-28 it looks as follows:

```

prj.Materials.ListOfMaterials = {
    struct("Type", "Insulator",
        "Name", "Air", "RelativePermittivity", 1.0, "LossTangent", 0.0), ...
    struct("Type", "Insulator", "Name", diel, "PolarizationLossModel",
        "WidebandDebye", ...
        "RelativePermittivity", 3.6, "LossTangent", 0.0117, "MeasurementFrequency",
        1.0e9), ... %starting point for Dk and LT
    struct("Type", "Conductor", "Name", cond, "RelativeResistivity", 1, ... %starting
    point for conductor relative bulk resistivity
        "Roughness", struct("RoughnessModelType", "HurayBracken", "SR0", 0.001, "RF0",
        2.0))... %starting point for roughness identification
};

```

Any type of material supported in Simbeor can be defined in a similar way – name of the parameter followed by the parameter itself. This format is universal for C and Python (the only difference in punctuation marks) – see examples for all types of materials in “..\Simbeor SDK Examples\cpp\testsdk_cpp\testsdk_cpp\test_json.cpp” file. Another way to see what parameters describe a particular model is to create a material of a particular type in Simbeor THz (see what is

available at https://kb.simberian.com/browse_item.php?id=768 and https://kb.simberian.com/browse_item.php?id=39) and then copy it in Simbeor THz and paste into Notepad for instance. For the Wideband Debye model provided above, you will see the material section like that

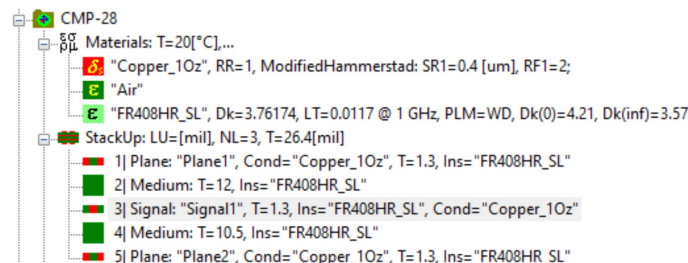
```
<Insulator Name = "FR408HR_SL" PolarizationLossModel = "WidebandDebye"
  RelativePermittivity = "3.74791" LossTangent = "1.01221e-2" MeasurementFrequency =
  "1000000000."
  WDLowFrequency = "100000." WDHighFrequency = "100000000000."
  BulkConductivity="0." RelativePermeability="1." NoModulationOrImbalance="False" / >
```

Exactly the same parameter names and values can be used in the Matlab or Python structures for the Materials.

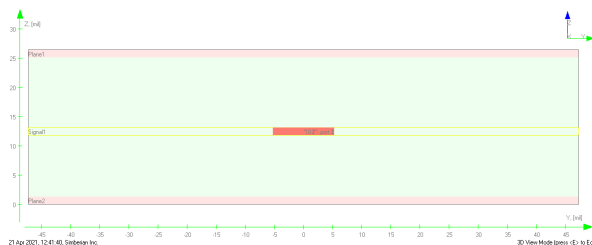
Structure StackUp contains description of the PCB/PKG stackup layer by layer as shown below

```
prj.StackUp.LengthUnit = prj.LengthUnit;
prj.StackUp.Layers = { ...
  struct("Type", "Plane", "Name", "Plane1", "Thickness", 1.3, "PourMaterialName",
  cond, "FillInsulatorName", diel)...
  struct("Type", "Medium", "Thickness", 12.0, "FillInsulatorName", diel),...
  struct("Type", "Signal", "Name", "Signal1", "Thickness", 1.3,
  "FillInsulatorName", diel, "DefaultConductorName", cond),...
  struct("Type", "Medium", "Thickness", 10.5, "FillInsulatorName", diel),...
  struct("Type", "Plane", "Name", "Plane2", "Thickness", 1.3, "PourMaterialName",
  cond, "FillInsulatorName", diel)...
};
```

It defines stackup with 2 plane layers (external), one signal layer (middle) and 2 dielectric layer around the signal layer that corresponds to the following stackup in Simbeor:



This stackup can be used to define stripline cross-section shown below and described in the `GMS.Model` structure:



All possibilities for the stackup description (including) the over-writing the roughness model data for different surfaces is demonstrated in `..\Simbeor SDK Examples\cpp\testsdk_cpp\testsdk_cpp\test_json.cpp` file It can be also explored through copy-pasting method as described for the Materials section.

Structure `GMS.Measured` defines measured Touchstone file location (`GMS.Measured.FileLocation`) and files with S-parameters for 2 line segments (`GMS.Measured.ShortSegmentFile` and `GMS.Measured.LongSegmentFile`), difference of line lengths

(GMS.Measured.LengthDifference) and name for the linear network for extraction of the GMS-parameters (GMS.Measured.DiffName). Measured data for CMP-28 can be defined as follows:

```
prj.GMS.Measured.FileLocation = 'C:\work\Touchstones'; %THE FILES MUST BE THERE
prj.GMS.Measured.ShortSegmentFile = 'cmp28_strpl_2in_50ohm_p1J6_p2J5.s2p';
prj.GMS.Measured.LongSegmentFile = 'cmp28_strpl_8inch_p1J7_p2J8.s2p';
prj.GMS.Measured.LengthDifference = 6000; %mil
prj.GMS.Measured.DiffName = 'Diff_6in'; %name of circuit with extracted GMS -
parameters to fit(difference)
```

Frequency sweep for the extraction of the measured GMS-parameters is defined as follows:

```
prj.GMS.FrqSweep.Start = 0.01; %/*GHz*/
prj.GMS.FrqSweep.Stop = 30.0; %/*GHz*/
prj.GMS.FrqSweep.Count = 3000; %/*max number of points for the adaptive sweep*/
prj.GMS.FrqSweep.AdaptivityTolerance = 0.1; %/*defines sensitivity of the adaptive
sweeper*/
```

Same sweep will be used to build models during the optimization. Note that the order of the parameters and structures is not important (may be different in actual script).

Next structure GMS.Model defines everything to build model of the transmission line segment with length equal to the length difference defined in the GMS.Measured structure. Structure GMS.Model.TLine contains description of the cross-section, GMS.Model.TLineName defines name for the cross-section model and GMS.Model.TLineSegmentName is name for the transmission line segment model where GMS-parameters of the model are computed and used during the material model optimization to match the measured GMS-parameters. Complete model description for CMP-28 example looks as follows:

```
prj.GMS.Model.TLine.Width = 10.5; %/* Strip width in the middle of the layer =
0.5*(w_top+w_bottom), [meter]*/
prj.GMS.Model.TLine.StripShape = 'Rectangular';
prj.GMS.Model.TLine.EtchFactor = 0.0; %/* Etch factor = 0.5*(w_top-w_bottom)/t1 */
prj.GMS.Model.TLine.Clearance = 0.0; %/* Clearance to plane metal for coplanar traces
if non-zero [meter], if zero, the strip is regular */
prj.GMS.Model.TLine.LayerName = 'Signal1'; %/*layer with the strip in the project
specified by the model name*/
prj.GMS.Model.TLine.Distance = 0.0; %center to center distance for differential traces

prj.GMS.Model.TLineName = 'Strip'; %cross - section model
prj.GMS.Model.TLineSegmentName = 'Seg_6in'; %stripline segment
```

In addition to the geometry and names, frequency sweep for the final model can be re-defined, to see how the model is extrapolated above the original measured data bandwidth – frequency sweep for the final analysis up to 50 GHz looks as follows:

```
prj.GMS.Model.FrqSweep.Start = 0.01; %/*GHz*/
prj.GMS.Model.FrqSweep.Stop = 50.0; %/*GHz*/
prj.GMS.Model.FrqSweep.Count = 5000; %/*max number of points for the adaptive sweep*/
prj.GMS.Model.FrqSweep.AdaptivityTolerance = 0.01; %/*defines sensitivity of the
adaptive sweeper*/
```

Third, define the identification algorithm (multiple scenarios described in previous chapter are possible). It can be done by defining parameters and structures in structure GMS.Identify (in the same file with the problem description for convenience) as illustrated next for loss separation scenario #3 applied to CMP-28 case:

```
prj.GMS.Identify.Dielectric.Names = { diel }; %dielectric model to identify
prj.GMS.Identify.Conductor.Name = cond; %conductor model to identify
```

```

prj.GMS.Identify.RoughnessAndLT = false; %simultaneous optimization of LT and Roughness
parameters
prj.GMS.Identify.FineTuneLosses = false; %second pass for LT and Roughness parameters

%GOAL FOR DIELECTRIC CONSTANT
prj.GMS.Identify.Dielectric.DK.Optimize = true;
prj.GMS.Identify.Dielectric.DK.Min = 1.0;
prj.GMS.Identify.Dielectric.DK.Max = 5.0;
prj.GMS.Identify.Dielectric.DK.FrqMin = 1.0e9; %Hz
prj.GMS.Identify.Dielectric.DK.FrqMax = 30.0e9; %Hz

%GOAL FOR LOSS TANGENT
prj.GMS.Identify.Dielectric.LT.Optimize = true;
prj.GMS.Identify.Dielectric.LT.Min = 0.0;
prj.GMS.Identify.Dielectric.LT.Max = 0.1;
prj.GMS.Identify.Dielectric.LT.FrqMin = 0.05e9; %Hz
prj.GMS.Identify.Dielectric.LT.FrqMax = 1.0e9; %Hz

%GOAL FOR RELATIVE RESISTIVITY
prj.GMS.Identify.Conductor.Resistivity.Optimize = true;
prj.GMS.Identify.Conductor.Resistivity.Min = 1.0;
prj.GMS.Identify.Conductor.Resistivity.Max = 1.2;
prj.GMS.Identify.Conductor.Resistivity.FrqMin = 0.01e9; %Hz
prj.GMS.Identify.Conductor.Resistivity.FrqMax = 0.05e9; %Hz

%GOAL FOR CONDUCTOR ROUGHNESS PARAMETERS
prj.GMS.Identify.Conductor.Roughness.SR.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.SR.Min = 0.0;
prj.GMS.Identify.Conductor.Roughness.SR.Max = 2; %um
prj.GMS.Identify.Conductor.Roughness.RF.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.RF.Min = 1.0;
prj.GMS.Identify.Conductor.Roughness.RF.Max = 20;
prj.GMS.Identify.Conductor.Roughness.FrqMin = 3.0e9; %Hz
prj.GMS.Identify.Conductor.Roughness.FrqMax = 30.0e9; %Hz

```

This particular scenario is for identification of dielectric and conductor roughness losses separately over different bandwidths (scenario 3 in the identification chapter). It also has additional step for the conductor relative bulk resistivity (conductivity) optimization. Let's walk through and explore different possibilities.

The first 2 parameters in GMS.Identify structure define dielectric and conductor for the model identification (introduced earlier in the Materials structure):

```

prj.GMS.Identify.Dielectric.Names = { diel }; %dielectric model to identify
prj.GMS.Identify.Conductor.Name = cond; %conductor model to identify

```

Next, parameter `Identify.RoughnessAndLT` defines how to identify loss model. If true, models for Roughness and Loss Tangent are identified over the same bandwidth (overlap of the bandwidths defined for Roughness and LT in corresponding structures), if both `Dielectric.LT` and `Conductor.Roughness` structures have `Optimize=true`. LT and Roughness models are identified separately as because of `Identify.RoughnessAndLT=false`;

Parameter `Identify.FineTuneLosses` can be used to fine tune LT and Roughness models by repeating the identification loop after they are identified first time. Setting it to `true` allows redistributing the losses better between the dielectric and conductor roughness models. It should be used only with very good measured data.

Next goals for the dielectric and conductor properties optimization are set in the structures Identify.Dielectric and Identify.Conductor. For each parameter to be optimized (Dielectric.DK, Dielectric.LT, Conductor.RelativeResistivity, Conductor.Roughness), parameter Optimize can be used to turn on/off optimization of a particular variable. Also, Min/Max values are defined, to restrict the range, and bandwidth in [Hz] is set with FrqMin and FrqMax. Optimize and Min/Max values are defined separately for Conductor.Roughness.SR and Conductor.Roughness.RF. The bandwidth for the Dielectric.LT is set up to 1 GHz (where effect of roughness is not significant). The conductor roughness identification bandwidth is set from 3 to 30 GHz.

Finally, run the optimization for this case - runExampleCMP28() in test_amk.m. The data is defined with defineMaterialIdentificationCMP28_JSON() and the identification requires just one call identifyMaterialModelsGMS() that fills Identified structure with the outcome for all parameters set for the optimization (see complete example in test_amk.m):

```
[bResult, prj] = defineMaterialIdentificationCMP28_JSON(sLibName, prj);
if bResult
    Identified = struct();
    [bResult, Identified] = identifyMaterialModelsGMS(sLibName, prj, Identified);
end
```

Though, before running the optimization, it is recommended to validate the GMS extraction and model parameters with the following calls:

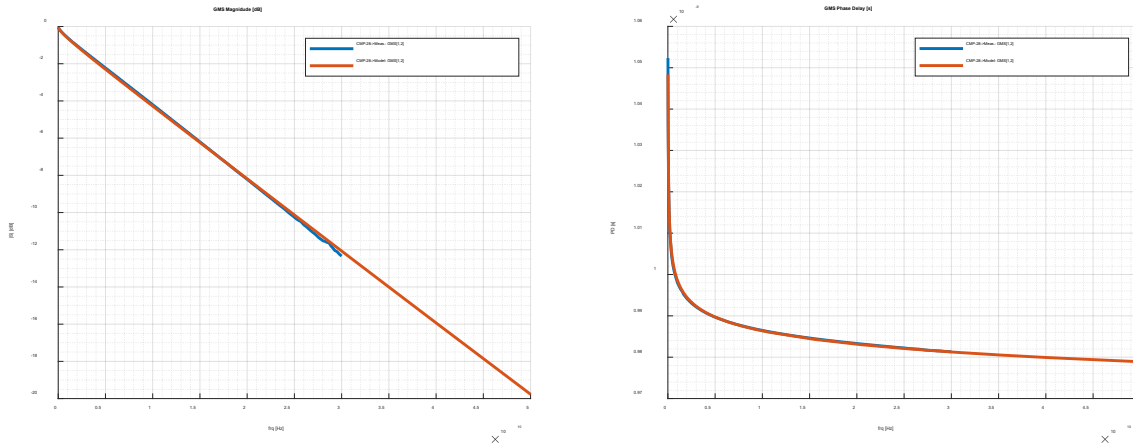
```
bResult1 = buildTLineModel_JSON(sLibName, prj, prj.GMS.Model, prj.GMS.FrqSweep,
prj.GMS.Measured.LengthDifference);
bResult2 = extractMeasuredGMS(sLibName, sProjectName, prj.GMS.Measured,
prj.GMS.FrqSweep);
```

Save the solution, load it into Simbeor THz and visually inspect the data – create views of cross-section, plot the extracted GMS-parameters and so on. The results of the optimization described above should look as follows:

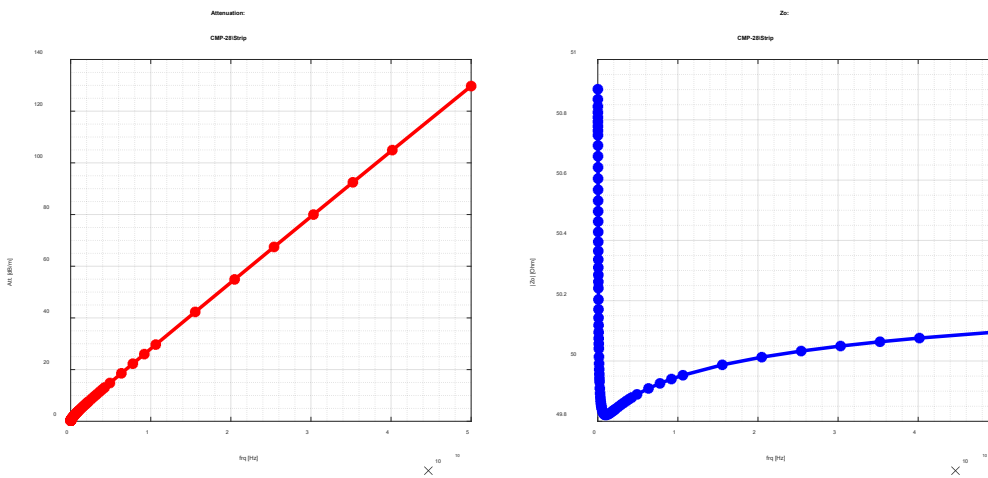
```
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
      Model: 'WidebandDebye'
      Dk : 3.7479
      LT : 0.0101
      Frq : 1.0000e+09

Name: 'Copper_10z'
      RR : 1.1550
      Roughness : [1x1 struct]
      Model : 'HurayBracken'
      SR : 0.0958
      RF : 18.1324
```

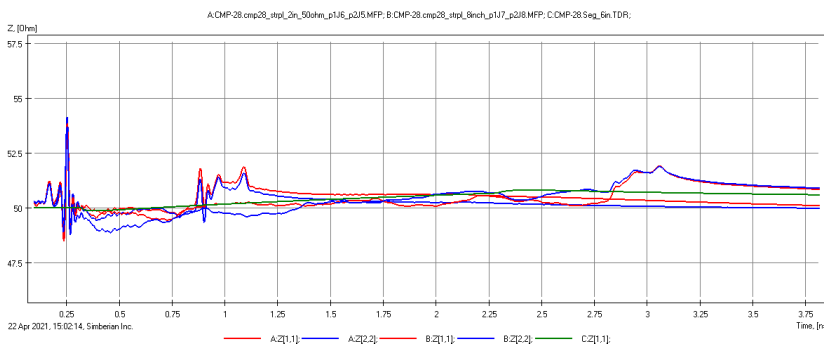
To verify the results, Simbeor create the final model (with the frequency sweep defined for the final run in the Model structure) and plots modeled and measured GMS insertion loss (left plot) and transmission phase delay (right plot) as illustrated below:



Plots of the attenuation per unit length and impedance vs frequency can be optionally created as well (to validate the impedance):



GMS-parameters are reflection-less – no correlation in the reflection or crosstalk parameters is needed. If all data and assumptions are correct, the correlation in GMS-transmission magnitude and phase should also produce correct transmission line impedance. To validate it, the impedance plot provided above can be effectively used. In this case the characteristic impedance at 1 GHz is about 49.8 Ohm that correlates very well with the impedance observed on TDR during the measured data validation. Alternatively, a segment of transmission line can be simulated and TDR plotted on the same graph with the measured data, something like that (green line is the model of 6 inch segment without the connectors and launches):



Large difference in the impedance can be observed due to the following reasons:

- a) Cross-section dimensions in the model are not the same as in the real structure. Boards are not manufactured as designed – request cross-sectioning from the manufacturer or do it yourself.
- b) Length difference of two segments of transmission line is not correct – that will lead to wrong Dk value that alter the characteristic impedance. Validate the lengths.
- c) Launches are very reflective that leads to the impedance transformation. That can lead to wrong impedance control adjustments for the whole board! Very reflective launches are usually very sensitive to the geometrical variations as well – that further reduces the useful bandwidth of GMS-parameters and can cause problems with the model identification in general.
- d) Investigated line segments include additional traces (differential breakout traces) that have substantial resistance (too narrow or thin) – that will create offset in the impedance of the useful part. It can also lead to wrong impedance control adjustment. Though, the material model identification results may be correct.

With the scripts provided in Simbeor SDK, it is easy to experiment with different extraction scenarios. Let’s start with the simplest one – identify only the dielectric parameters (scenario #1 from previous chapter). To do it only the following parameters of the original script have to be re-defined (we will keep the roughness as originally defined):

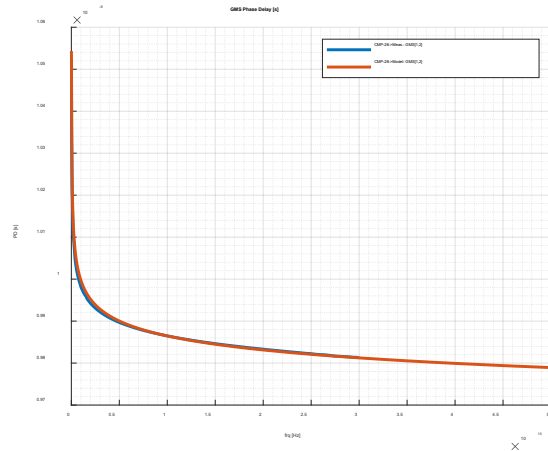
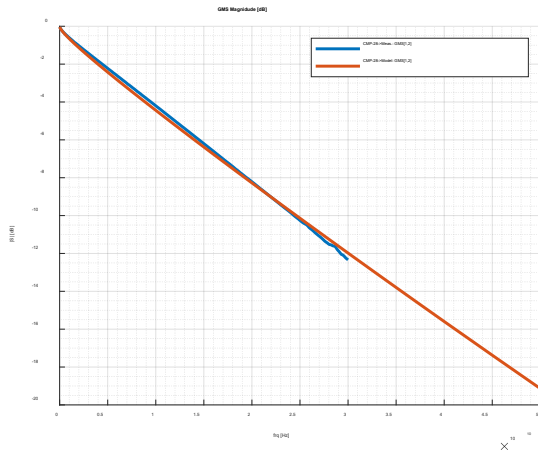
```
prj.GMS.Identify.Dielectric.LT.FrqMax = 30.0e9; %Hz
prj.GMS.Identify.Conductor.Resistivity.Optimize = false;
prj.GMS.Identify.Conductor.Roughness.SR.Optimize = false;
prj.GMS.Identify.Conductor.Roughness.RF.Optimize = false;
```

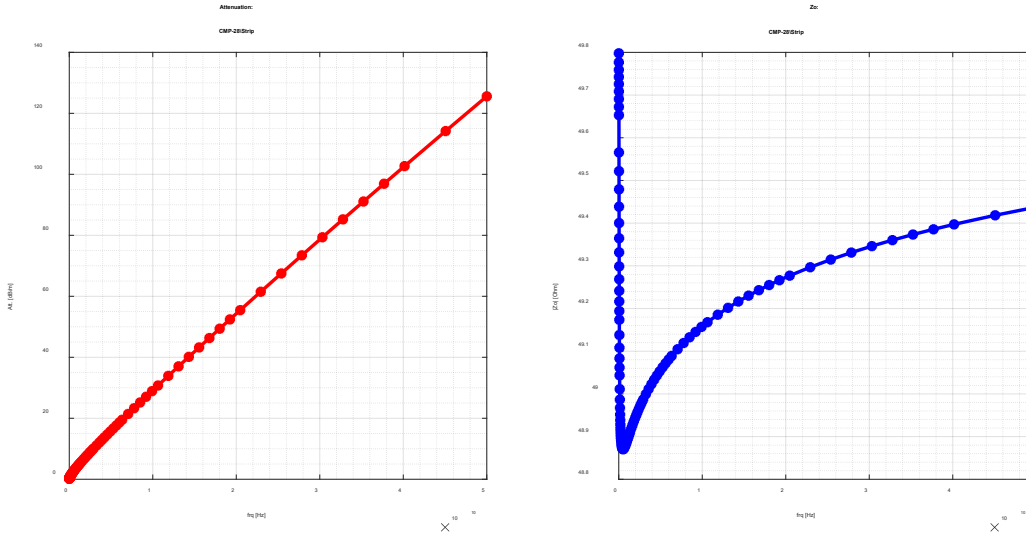
The identified results are as follows:

```
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
      Model: 'WidebandDebye'
      Dk : 3.8230
      LT : 0.0127
      Frq : 1.0000e+09
```

```
Name : 'Copper_10z'
RR : 1
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 1.0000e-03
RF : 2
```

This model produces the following correlation of the measured and model GMS-parameters and impedance:





As we can see, the correlation in the GMS insertion loss is not so good (between 5 and 20 GHz) and the impedance went down by about 1 Ohm (no inductance produced by the rough copper surface). Though, the impedance is within the variations observed on TDR. Note, that the model accuracy with all losses attributed to dielectric is good only for a particular trace width (or small variation of it), even if the insertion loss correlates well (not in this).

Now, let's explore scenario #2 – fix the loss tangent to the value provided by manufacturer (Isola in this case) and identify the conductor roughness model. This can be done by small modification of the original script as follows:

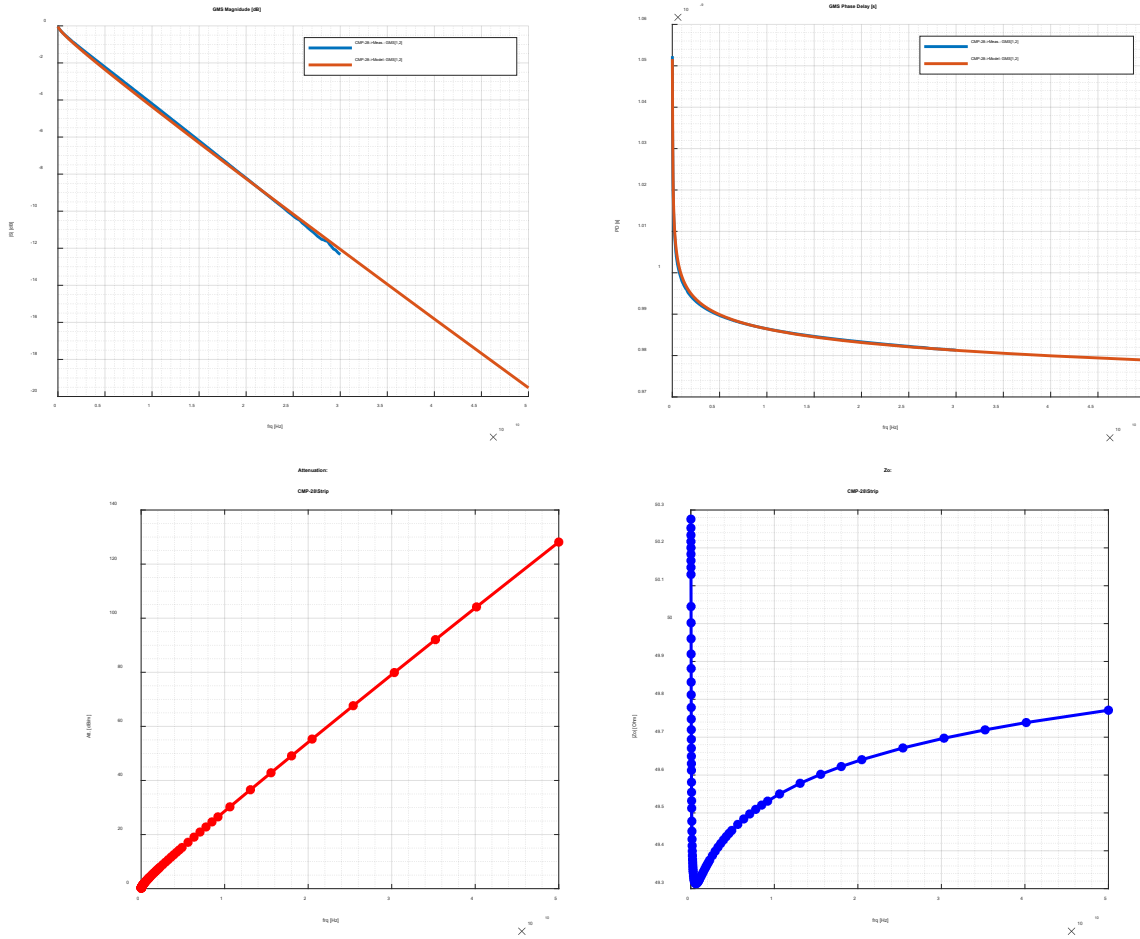
```
prj.GMS.Identify.Dielectric.LT.Optimize = false;
prj.GMS.Identify.Conductor.Resistivity.Optimize = false;
prj.GMS.Identify.Conductor.Roughness.SR.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.RF.Optimize = true;
```

It produces the following results:

```
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
Dk : 3.7877
LT : 0.0117
Frq : 1.0000e+09
```

```
Name : 'Copper_10z'
RR : 1
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 0.0732
RF : 11.8505
```

With the following correlation and impedance:



This is also reasonable outcome for this case, comparing to scenario #3 used first here.

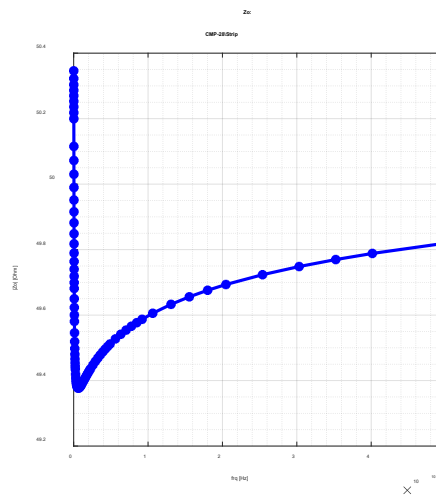
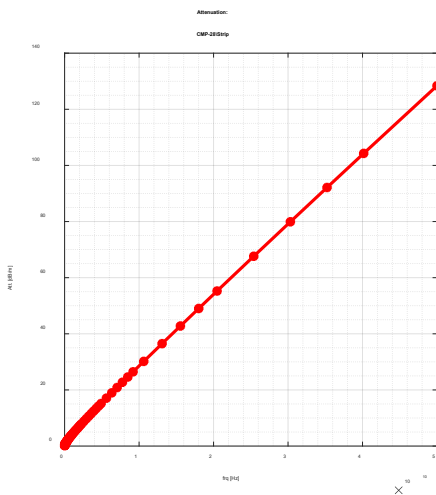
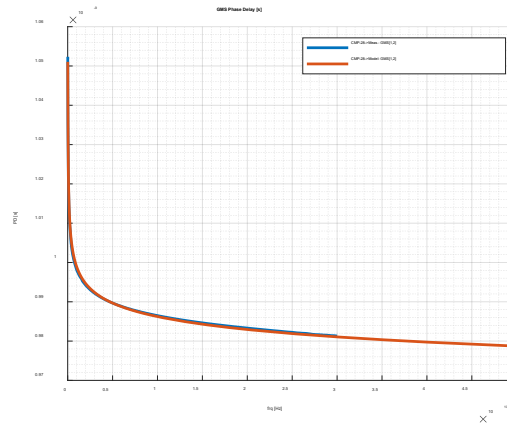
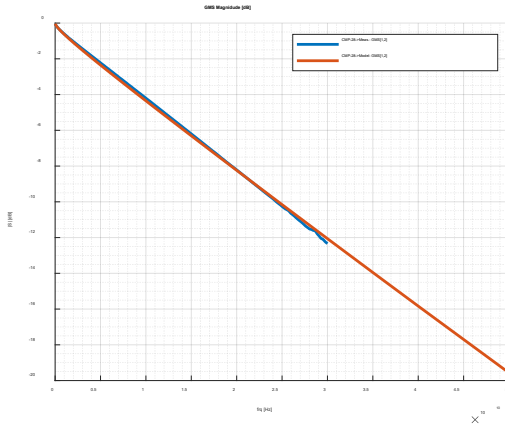
Another option is simultaneous optimization of losses from dielectric and conductor roughness. To check this scenario, small modification is needed as follows:

```
prj.GMS.Identify.RoughnessAndLT = false;
prj.GMS.Identify.Dielectric.LT.Optimize = true;
prj.GMS.Identify.Conductor.Resistivity.Optimize = false;
prj.GMS.Identify.Conductor.Roughness.SR.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.RF.Optimize = true;
```

Unlike the previous scenarios, it can take longer time to optimize 3 parameters affecting the losses simultaneously (LT, SR, RF). The result is as follows:

```
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
Dk : 3.7818
LT : 0.0116
Frq : 1.0000e+09
Name : 'Copper_10z'
RR : 1
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 0.0748
RF : 12.9040
```

And the correlation and impedance are as follows:



Notice, that this model is practically the same as for the previous case with fixed LT and only roughness model parameters optimized.

The relative resistivity was optimized at very low frequencies only in the first case (scenario #3), where the loss tangent was also identified at lower frequencies. It should be done only if S-parameters at lower frequency are reliable (not always the case). The loss tangent provided by manufacturer in this case was larger than the identified. If the loss tangent is fixed or optimized together with the roughness model, the accuracy of the models will degrade at lower frequencies due to slightly larger than observed dielectric losses at lower frequencies.

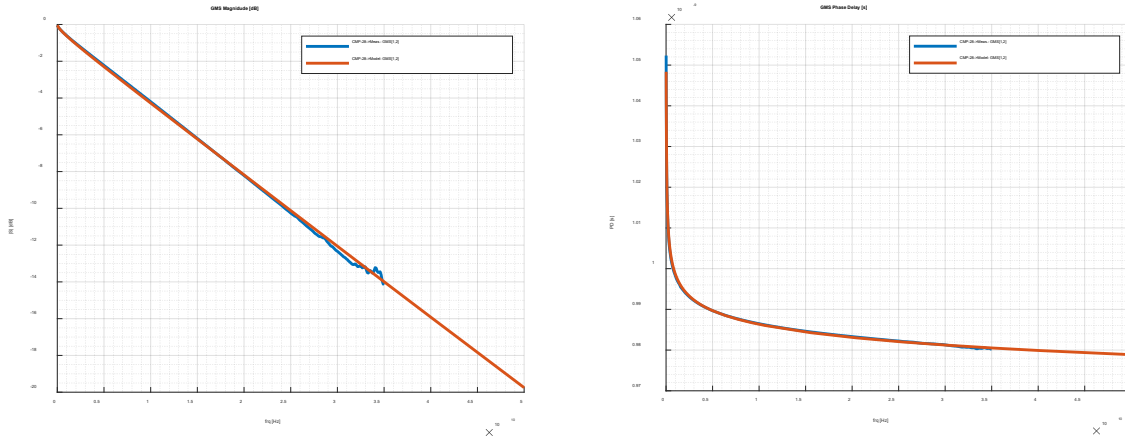
Notice that the maximal frequency in the frequency sweeps for all models above is set to 50 GHz. All models produced consistent results above the 30 GHz frequency in the original extracted GMS-parameters. With the automation script it is very easy to investigate the outcome with difference fitting bandwidths. Let's extend the fitting bandwidth to 35 GHz (maximal possible in this case) and use loss separation scenario #3 for instance. The result is as follows:

```
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
Dk : 3.7479
LT : 0.0101
Frq : 1.0000e+09
```

```
Name: 'Copper_10z'
RR : 1.1550
```

Roughness : [1x1 struct]
 Model : 'HurayBracken'
 SR : 0.0958
 RF : 18.0638

Technically, it is exactly the same result as with 30 GHz bandwidth. Correlation of insertion loss and phase delay is shown next:



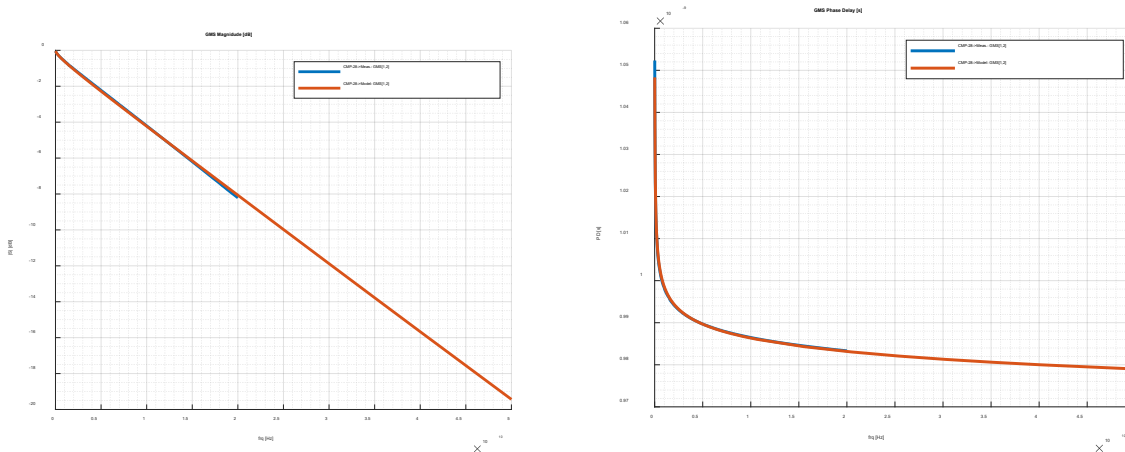
We can see that the noise at the end of GMS insertion loss (wiggling of blue line on the left plot above) did not change the result.

Now, let's see if we reduce the bandwidth to 20 GHz for instance. The result of the identification is as follows:

CMP - 28 Final Results:
 Name: {'FR408HR_SL'}
 Model: 'WidebandDebye'
 Dk : 3.7502
 LT : 0.0101
 Frq : 1.0000e+09

Name: 'Copper_10z'
 RR : 1.1550
 Roughness : [1x1 struct]
 Model : 'HurayBracken'
 SR : 0.0892
 RF : 18.4484

With the following correlation:



Again, the model is practically the same as with 30 GHz bandwidth. **This is the main advantage of the frequency-continuous model identification!** 20 GHz VNA would be sufficient in this case to get the material models usable up to 50 GHz and may be above that frequency. To validate it, just use CMP-28 kit and construct models for 2 and 8 inch segments with connectors and launches (prototypes are in Simbeor CMP-28 Kit) and observe the difference with the measured data.

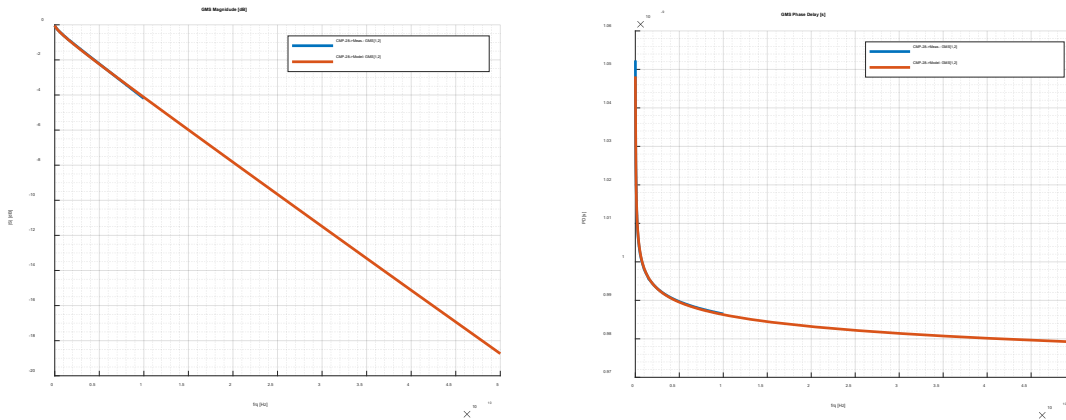
Let's push this case to the limit and further reduce the bandwidth to 10 GHz. Here is the outcome:

```

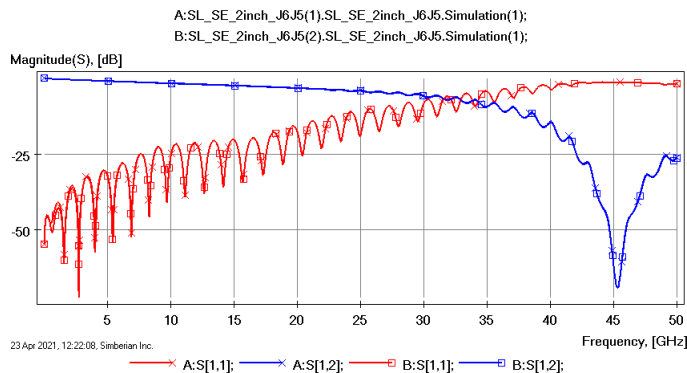
CMP - 28 Final Results :
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
Dk : 3.7549
LT : 0.0101
Frq : 1.0000e+09

Name : 'Copper_10z'
RR : 1.1550
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 0.0748
RF : 19.4651
    
```

With the following correlation:



We can see that the model start deviating from the results obtained with wider bandwidth, but only at frequencies above 25-30 GHz. Difference at 50 GHz is about 0.11 dB/inch (GMS-parameters above are for 6-inch segment). There is no significant degradation in the phase delay. If we use models constructed with 35 GHz fitting bandwidth and with 10 GHz bandwidth for analysis of the original 2 inch segment with connectors and launches, the result of simulations practically overlap:



The phases are even closer, because of mostly the losses are affected by the fitting bandwidth selection.

To facilitate this fitting bandwidth experiment, parameter *fgmax* was used in `defineMaterialIdentificationCMP28_JSON.m` script. It defines the frequency sweep for the optimization as well as `FrqMax` for DK and Roughness parameters.

The final experiment with CMP-28 platform is to investigate how choice of the conductor roughness model affects the results. We used “HurayBracken” model in all previous case study. Let’s try “CausalHammerstad” instead with 30 GHz fitting bandwidth. To do that, only one modification in the original script is needed in the structure describing the conductor:

```

struct("Type", "Conductor", "Name", cond, "RelativeResistivity", 1,... %starting
point for conductor relative bulk resistivity
"Roughness", struct("RoughnessModelType", "CausalHammerstad", "SR0", 0.001, "RF0",
2.0))... %starting point for roughness identification
    
```

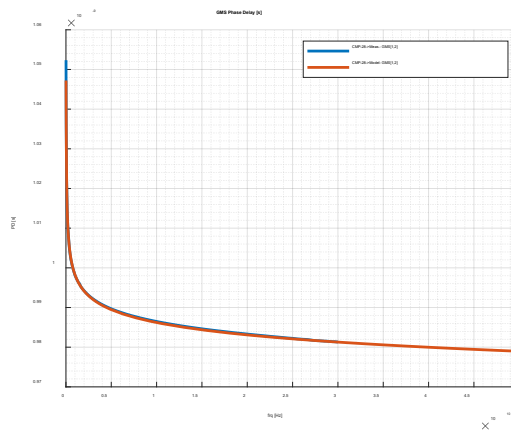
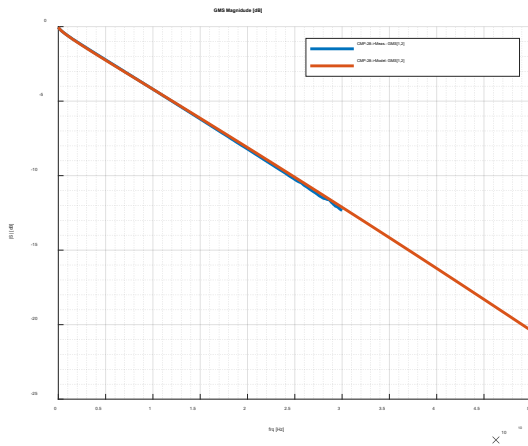
The result is as follows:

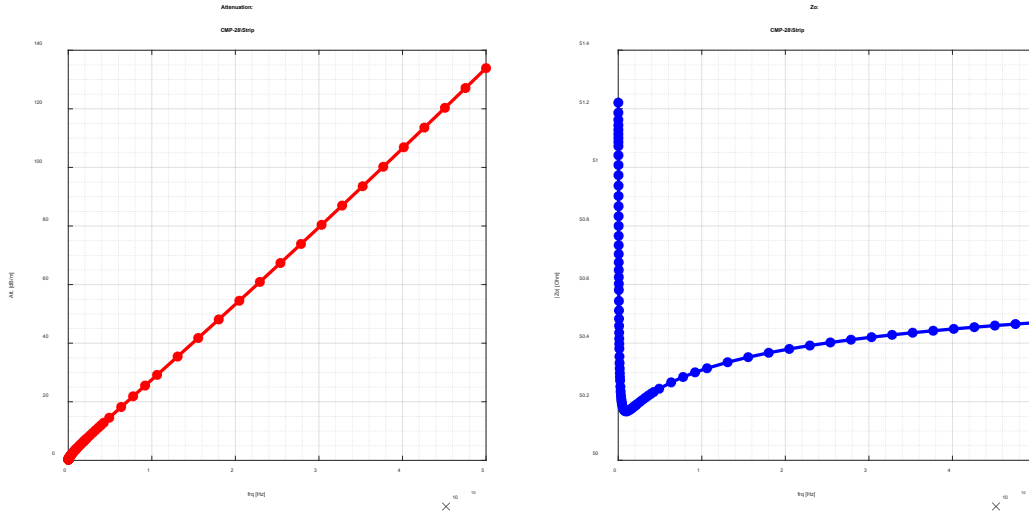
```

CMP - 28 Final Results :
Name: {'FR408HR_SL'}
Model: 'WidebandDebye'
DK : 3.7205
LT : 0.0101
Frq : 1.0000e+09

Name : 'Copper_10z'
RR : 1.1550
Roughness : [1x1 struct]
Model : 'CausalHammerstad'
SR : 0.1118
RF : 19.6886
    
```

With the following correlation of the model with the measurements:

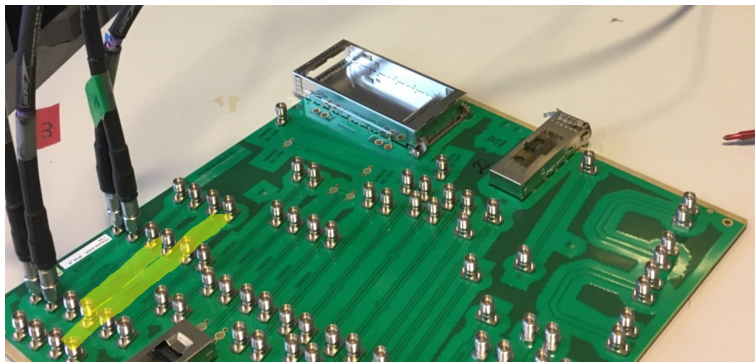




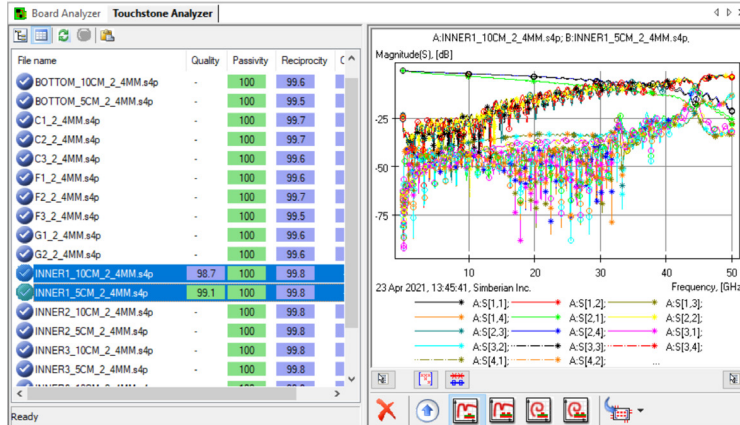
This model predicts slightly larger losses at the frequencies above the fitting bandwidth, but, most important, larger impedance. The impedance at 1 GHz in this case is about 50.1 Ohm – that may be larger than observed on the TDR. Note that the dimensions for the cross-section were provided by PCB manufacturer in this case. It is usually reliable for striplines (valid for some manufacturers). Though, it has to be confirmed for the final conclusion on which roughness model is better in this case.

Example 2: Material Model Identification On EvR-1

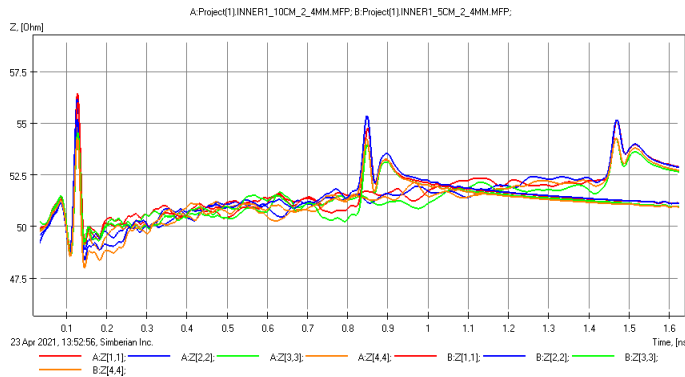
As the second example of the material model identification with differential line segment we will use EvR-1 validation platform with all measured and board description data provided by Marko Marin when he was with Infinera. Complete EvR-1 description is available at #2018_01 at <https://www.simberian.com/TechnicalPresentations.php> and #2018_01 at <https://www.simberian.com/AppNotes.php> Two differential stripline segments in layer INNER1 highlighted on the picture below are used for the identification here.



The first step is to measure S-parameters for 2 line segments and pre-qualify them for the material identification. In this case those are 4-port S-parameters (s4p files). Here we use quality evaluation in Touchstone Analyzer as shown below (here is how to do it https://kb.simberian.com/browse_item.php?id=777):

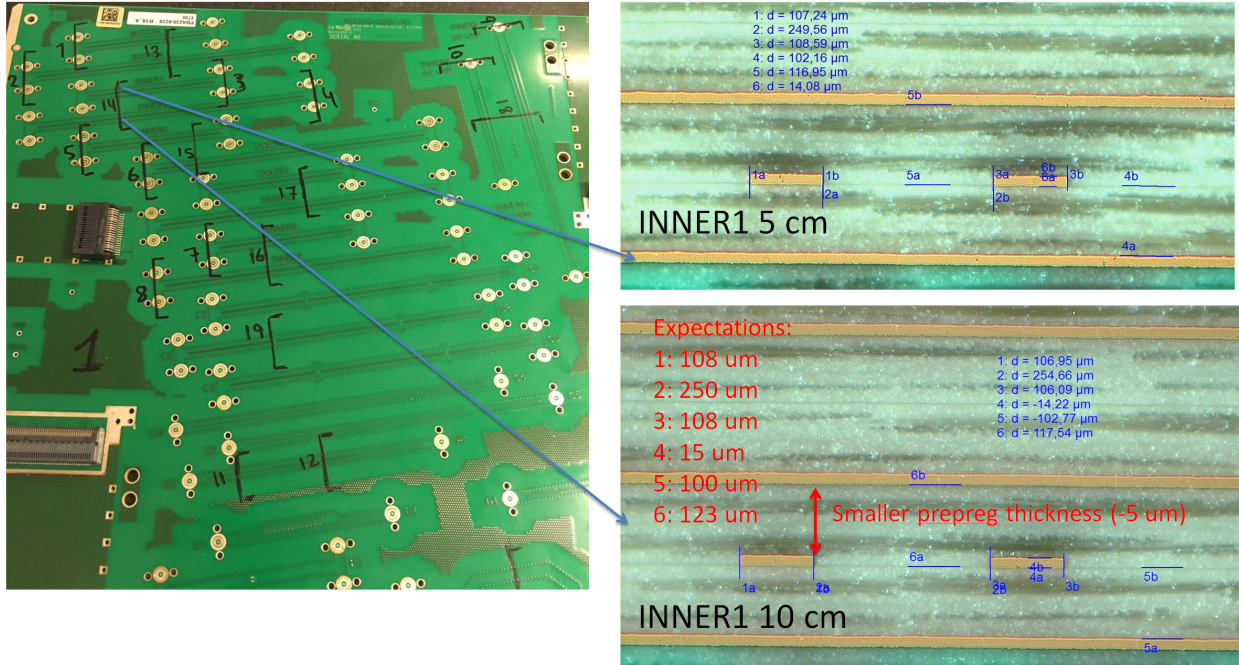


As we can see, the final quality metrics for 2 segments of transmission line is acceptable (Quality is over 98%). Pre-qualification must also include evaluation of the differences in TDRs of 2 line segments. Specifically, observed line impedance and connector or probes + launch discontinuities should be consistent in both segments. For extraction up to 30 GHz, the difference in TDR should not exceed 2-3 Ohm – see app note #2010_03 at <https://www.simberian.com/AppNotes.php> on the sensitivity of GMS-parameters to the manufacturing variations. TDR plot for 2 differential stripline segments on EvR-1 is shown on the following graph (single-ended TDR is used for the consistency evaluation):



As we can see, the consistency of the launches and impedance deviation are sufficient for the material identification with GMS-parameters extracted up to 30 GHz in this case. Though, the identified material models will be usable well above of that frequency.

Cross-sectioning of this board was also done and the results are shown below together with the data obtained from the PCB manufacturer:



The differences between the measured and manufacturer data for the stripline layer are not significant in this case - data from the manufacturer can be effectively used (though it is different from the designed board – see EvR-1 description for additional details).

The second step is to describe the data for the identification (S-parameter files and cross-section). Here is how to do it with Matlab data structure. When Simbeor SDK is installed (see instructions at https://kb.simberian.com/browse_item.php?id=985), S-parameter files for 5 and 10 cm segments of stripline from EvR-1 platform will be located at the “..\Simbeor SDK Examples\touchstones” subdirectory. To run the model identification scripts, either define the path to that directory or create and copy the files into directory “C:\work\Touchstones”, to use the identification script as is. Here is the complete description of the data for EvR-1 example using JSON format for the material and stackup definitions (file defineMaterialIdentification_EvR1_JSON.m, the order of the structures and data does not matter and may be altered for better understanding):

```
prj.Format = 'JSON'; %STACKUP AND MATERIALS IN JSON FORMAT
prj.ProjectName = 'EvR-1'; %PROJECT NAME TO HOLD ALL MODELS AND MEASURED DATA
prj.LengthUnit = 'mil';

%COMMON SWEEP FOR MODELS AND GMS FROM MEASUREMENTS
%unit default for frequency sweep is GHZ
prj.GMS.FrqSweep.Start = 0.07; %/*GHz*/ -NOT RELIABLE DATA AT LOWER FREQUENCIES
prj.GMS.FrqSweep.Stop = 30.0; %/*GHz*/
prj.GMS.FrqSweep.Count = 3000; %/*max number of points for the adaptive sweep*/
prj.GMS.FrqSweep.AdaptivityTolerance = 0.1; %/*defines sensitivity of the adaptive
sweeper*/

prj.GMS.Measured.FileLocation = 'C:\work\Touchstones';
prj.GMS.Measured.ShortSegmentFile = 'INNER1_5CM_2_4MM.s4p';
prj.GMS.Measured.LongSegmentFile = 'INNER1_10CM_2_4MM.s4p';
prj.GMS.Measured.LengthDifference = m_to_mil(5e-2); % 5 cm
prj.GMS.Measured.DiffName = 'Diff_5cm'; %name of circuit with extracted GMS -
parameters to fit(difference)

die11 = 'Meg-6-C'; %DIELECTRIC TO IDENTIFY
```

```

diel2 = 'Meg-6-P'; %DIELECTRIC TO IDENTIFY
diel3 = 'Resin'; %ANOTHER DIELECTRIC TO IDENTIFY, DIFFERENT FROM OTHER
cond = 'Copper'; %CONDUCTOR TO IDENTIFY

dThicknessInternal = m_to_mil(15e-6); %// 15 um - THICKNESS OF EACH SIGNAL OR PLANE
LAYER IS DEFINED EXPLICITELY!!!!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATERIALS AND STACKUP FOR JSON TRANSFER
prj.Materials.ListOfMaterials = {
    struct("Type", "Insulator",
        "Name", "Air", "RelativePermittivity", 1.0, "LossTangent", 0.0, "PolarizationLossModel",
        "Lossless"),...
    struct("Type", "Insulator", "Name", diel1, "PolarizationLossModel",
        "WidebandDebye"),...
    struct("RelativePermittivity", 3.37, "LossTangent", 0.003, "MeasurementFrequency",
        1.0e9),... %starting point for Dk and LT
    struct("Type", "Insulator", "Name", diel2, "PolarizationLossModel",
        "WidebandDebye"),...
    struct("RelativePermittivity", 3.32, "LossTangent", 0.003, "MeasurementFrequency",
        1.0e9),... %starting point for Dk and LT
    struct("Type", "Insulator", "Name", diel3, "PolarizationLossModel",
        "WidebandDebye"),...
    struct("RelativePermittivity", 3.32, "LossTangent", 0.003, "MeasurementFrequency",
        1.0e9),... %starting point for Dk and LT
    struct("Type", "Conductor", "Name", cond, "RelativeResistivity", 1.2,... %starting
    point for conductor relative bulk resistivity
    "Roughness", struct("RoughnessModelType", "HurayBracken", "SR0", 0.001, "RF0",
        2.0))... %starting point for roughness identification
};
prj.StackUp.LengthUnit = prj.LengthUnit;
if ~WithResinLayer
prj.StackUp.Layers = { ...
    struct("Type", "Plane", "Name", "GND1", "Thickness", dThicknessInternal,
        "PourMaterialName", cond, "FillInsulatorName", diel2)...
    struct("Type", "Medium", "Thickness", m_to_mil(100e-6),
        "FillInsulatorName", diel2),...
    struct("Type", "Signal", "Name", "INNER1", "Thickness", dThicknessInternal,
        "FillInsulatorName", diel2, "DefaultConductorName", cond),...
    struct("Type", "Medium", "Thickness", m_to_mil(103e-6),
        "FillInsulatorName", diel1),...
    struct("Type", "Plane", "Name", "GND2", "Thickness", dThicknessInternal,
        "PourMaterialName", cond, "FillInsulatorName", diel2)...
};
else
prj.StackUp.Layers = { ...
    struct("Type", "Plane", "Name", "GND1", "Thickness", dThicknessInternal,
        "PourMaterialName", cond, "FillInsulatorName", diel2)...
    struct("Type", "Medium", "Thickness", m_to_mil(85e-6),
        "FillInsulatorName", diel2),...
    struct("Type", "Medium", "Thickness", m_to_mil(15e-6),
        "FillInsulatorName", diel3),...
    struct("Type", "Signal", "Name", "INNER1", "Thickness", dThicknessInternal,
        "FillInsulatorName", diel3, "DefaultConductorName", cond),...
    struct("Type", "Medium", "Thickness", m_to_mil(15e-6),
        "FillInsulatorName", diel3),...
    struct("Type", "Medium", "Thickness", m_to_mil(88e-6),
        "FillInsulatorName", diel1),...
    struct("Type", "Plane", "Name", "GND2", "Thickness", dThicknessInternal,
        "PourMaterialName", cond, "FillInsulatorName", diel2)...
};

```

```

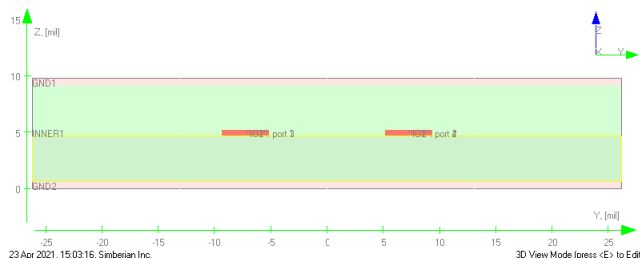
end
%%%%
prj.GMS.Model.TLine.Width = 4.21; /* Strip width in the middle of the layer =
0.5*(w_top+w_bottom), [meter]*/
prj.GMS.Model.TLine.StripShape = 'Rectangular';
prj.GMS.Model.TLine.EtchFactor = 0.0; /* Etch factor = 0.5*(w_top-w_bottom)/t1 */
prj.GMS.Model.TLine.Clearance = 0.0; /* Clearance to plane metal for coplanar traces
if non-zero [meter], if zero, the strip is regular */
prj.GMS.Model.TLine.LayerName = 'INNER1'; /*layer with the strip in the project
specified by the model name*/
prj.GMS.Model.TLine.Distance = 10.29; %center to center distance for differential
traces

prj.GMS.Model.TLineName = 'Strip'; %cross - section model
prj.GMS.Model.TLineSegmentName = 'Seg_5cm'; %strip line segment

%FREQUENCY SWEEP FOR THE FINAL ANALYSIS - TO SEE THE MODEL EXTENSION
prj.GMS.Model.FrqSweep.Start = 0.01; /*GHz*/
prj.GMS.Model.FrqSweep.Stop = 50.0; /*GHz*/
prj.GMS.Model.FrqSweep.Count = 5000; /*max number of poins for the adaptive sweep*/
prj.GMS.Model.FrqSweep.AdaptivityTolerance = 0.01; /*defines sensitivity of the
adaptive sweeper*/
    
```

Structure GMS.Measured is defined similar to the first example (see description in CMP-28 example chapter). Structure Materials includes 3 PCB dielectrics Meg-6-C for core, Meg-6-P for prepreg and optional dielectric Resin, to model filling of the signal layer INNER1 and small areas around it (“butter-coating”). Differential transmission lines have two modes – odd (differential) and even (common). Thus, GMS-parameters will have 2 non-zero parameters – GMS transmission for the odd (#1) and even (#2) modes. They allow constructing 2 goal functions for each of the parameters and finding 2 dielectric constants that correlate with the 2 modal transmission parameters. This is needed only in case if we observe differences in propagation constants of the odd and even mode due to inhomogeneity of the dielectric filling. Resin layer can be used as additional dielectric to fit. This is necessary only in cases if far end crosstalk on striplines required to model with high accuracy. Parameter WithResinLayer in the script above allows definition of the StackUp structure with or without the Resin layer in the middle of the stackup. Conductor material “Copper” in this case is defined with Relative Resistivity 1.2 – it was identified with different measurements because of measured S-parameters had some issues at lower frequencies – details are explained in the Simbeor EvR-1 Kit and corresponding paper.

Structure GMS.Model describes the cross-section and sweep for the final model. The model cross-section with just 2 dielectrics Meg-6-C and Meg-6-P is shown below (core is highlighted):



Third step is to define the identification algorithm (multiple scenarios described in previous chapter are possible). It can be done by defining parameters and structures in the structure

GMS.Identify (in the same file with the problem description for convenience) as illustrated next for one of the possible scenarios of material model identification for EvR-1 case (scenario #2):

```

prj.GMS.Identify.RoughnessAndLT = false; % simultaneous optimization of SR, RF and LT
prj.GMS.Identify.FineTuneLosses = false; %second pass for LT and Roughness parameters

if ~WithResinLayer
    prj.GMS.Identify.Dielectric.Names = { diel1, diel2 }; %dielectric models to
    identify
    prj.GMS.Identify.Dielectric.Identity = [1, 1]; %same properties for both
    dielectrics
else %WITH ADDITIONAL RESIN LAYER AROUND THE SIGNAL LAYER
    prj.GMS.Identify.Dielectric.Names = { diel1, diel2, diel3 }; %dielectric models to
    identify
    %FIRST GROUP OF DIELECTRICS FOR EVEN MODE, SECOND FOR THE ODD MODE(FILL)
    prj.GMS.Identify.Dielectric.Identity = [1, 1, 2]; %same properties for 2
    dielectrics and different for another one
end

prj.GMS.Identify.Conductor.Name = cond; %conductor model to identify

%GOAL FOR DIELECTRIC CONSTANT
prj.GMS.Identify.Dielectric.DK.Optimize = true;
prj.GMS.Identify.Dielectric.DK.Min = 1.0;
prj.GMS.Identify.Dielectric.DK.Max = 5.0;
prj.GMS.Identify.Dielectric.DK.FrqMin = 1.0e9; %Hz
prj.GMS.Identify.Dielectric.DK.FrqMax = 30.0e9; %Hz

%GOAL FOR LOSS TANGENT
prj.GMS.Identify.Dielectric.LT.Optimize = false;
prj.GMS.Identify.Dielectric.LT.Min = 0.0;
prj.GMS.Identify.Dielectric.LT.Max = 0.1;
prj.GMS.Identify.Dielectric.LT.FrqMin = 0.07e9; %Hz
prj.GMS.Identify.Dielectric.LT.FrqMax = 1.0e9; %Hz

%GOAL FOR RELATIVE RESISTIVITY
prj.GMS.Identify.Conductor.Resistivity.Optimize = false; %NO FREQUENCIES TO DO IT FOR
EVR - 1
prj.GMS.Identify.Conductor.Resistivity.Min = 1.0;
prj.GMS.Identify.Conductor.Resistivity.Max = 1.4;
prj.GMS.Identify.Conductor.Resistivity.FrqMin = 0.01e9; %Hz
prj.GMS.Identify.Conductor.Resistivity.FrqMax = 0.05e9; %Hz

%GOAL FOR CONDUCTOR ROUGHNESS PARAMETERS
prj.GMS.Identify.Conductor.Roughness.SR.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.SR.Min = 0.0;
prj.GMS.Identify.Conductor.Roughness.SR.Max = 0.1; %um - HVLP
prj.GMS.Identify.Conductor.Roughness.RF.Optimize = true;
prj.GMS.Identify.Conductor.Roughness.RF.Min = 1.0;
prj.GMS.Identify.Conductor.Roughness.RF.Max = 20;
prj.GMS.Identify.Conductor.Roughness.FrqMin = 3.0e9; %Hz
prj.GMS.Identify.Conductor.Roughness.FrqMax = 30.0e9; %Hz

```

As was already mentioned, the relative resistivity of the copper will be not optimized in this case. No dielectric and conductor loss separation in this case – the loss tangent value is fixed to one provided by laminate manufacturer. This is reasonable choice for very low loss dielectrics and problem with S-parameters at lower frequencies as observed in this case.

Without the additional resin layer (WithResinLayer=false), two dielectrics will be considered to have identical properties. This is defined as follows:

```
Identify.Dielectric.Names = { diel1, diel2 }; %dielectric models to identify
Identify.Dielectric.Identity = [1, 1]; %same properties for both dielectrics
diel1 (core) and diel2 (prepreg) will have the same properties – two 1 in Identity array.
```

With the additional resin layer (WithResinLayer=true), two dielectrics will be considered identical and Resin (diel3) will have different properties (to have differences in the odd and even modes). This is defined as follows:

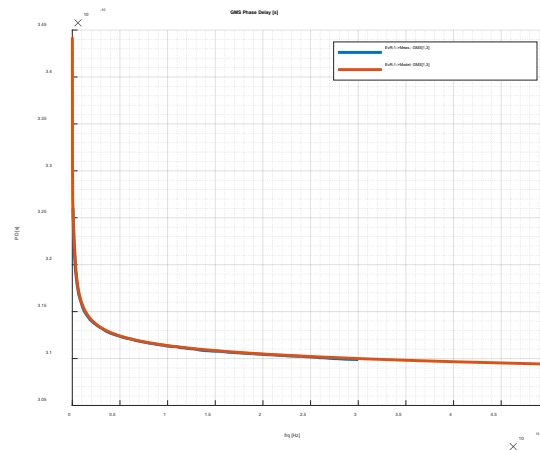
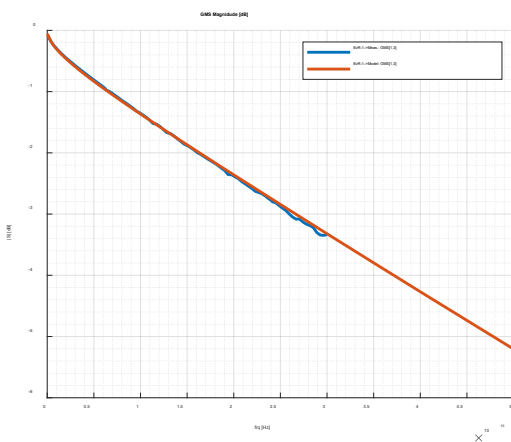
```
Identify.Dielectric.Names = { diel1, diel2, diel3 }; %dielectric models to identify
%FIRST GROUP OF DIELECTRICS FOR EVEN MODE, SECOND FOR THE ODD MODE(FILL)
Identify.Dielectric.Identity = [1, 1, 2]; %same properties for 2 dielectrics and
different for another one
```

Simbeor will use additional optimization loop to fit GMS-parameters for odd and even modes simultaneously.

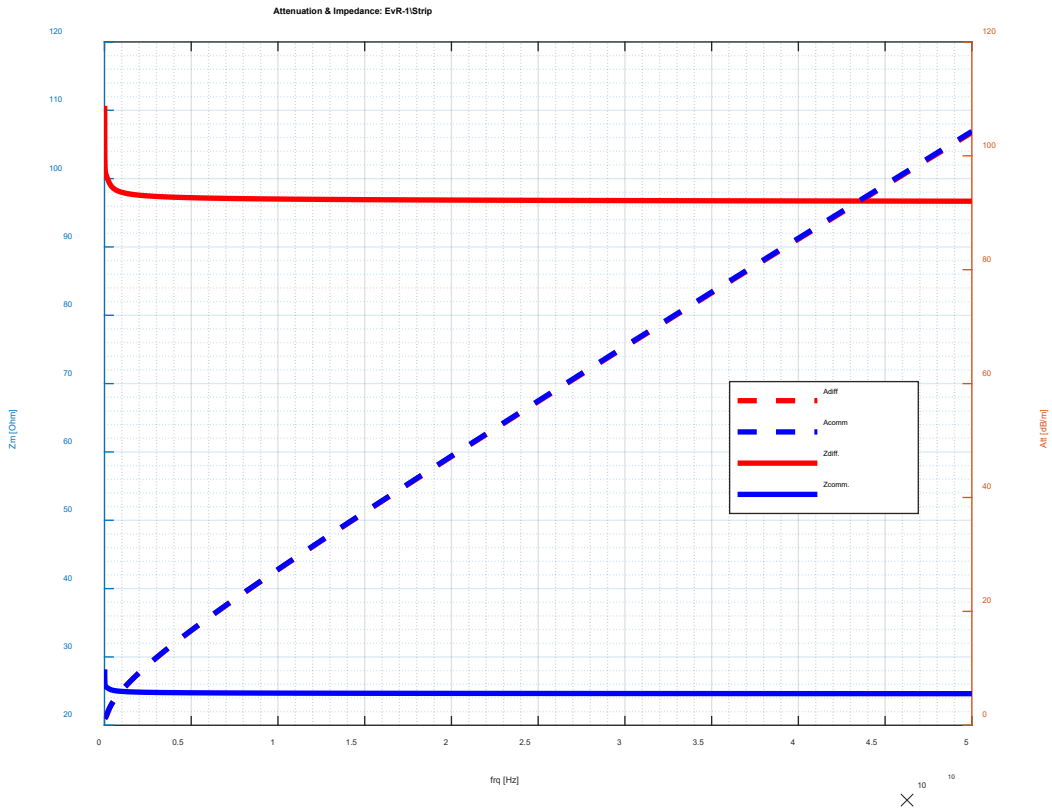
Finally, run the optimization for this case – runExampleEvR1() in test_amk.m. The data is defined with defineMaterialIdentificationEvR1_JSON() and identification require just one call identifyMaterialModelsGMS() that fills Identified structure with the outcome for all parameters set for the optimization. The results of the identification without additional resin layer will look as shown below:

```
EvR - 1 Final Results :
Name: {'Meg-6-C'}
Model: 'WidebandDebye'
Dk : 3.3807
LT : 0.0030
Frq : 1.0000e+09
Name : {'Meg-6-P'}
Model: 'WidebandDebye'
Dk : 3.3807
LT : 0.0030
Frq : 1.0000e+09
Name : 'Copper'
RR : 1.2000
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 0.1000
RF : 10.9729
```

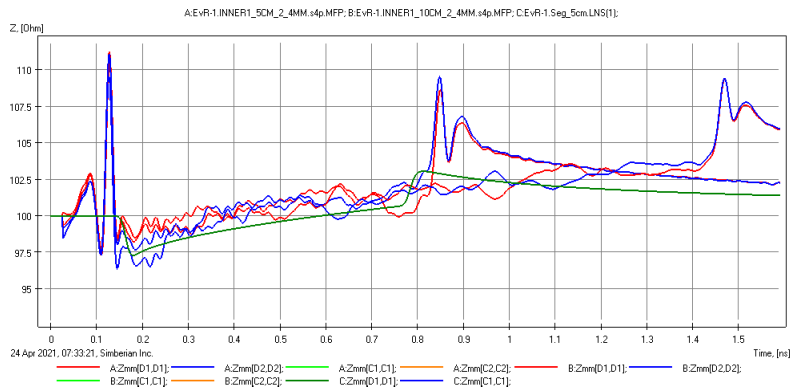
It corresponds to the following correlation in measured and modeled GMS-parameters (odd or differential mode only):



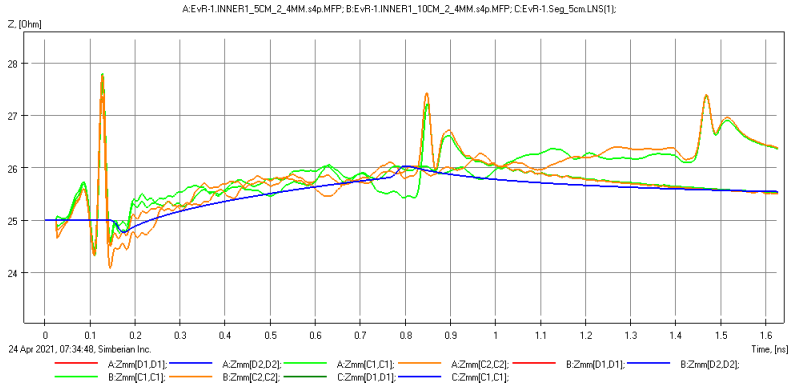
And modal characteristic impedance and attenuation per unit length as shown below:



That correlates with the impedance observed at TDR plots for differential mode (about 97.5 Ohm on average immediately after the launch, green line is the model without the launches, test structure has width of single-ended and differential traces):

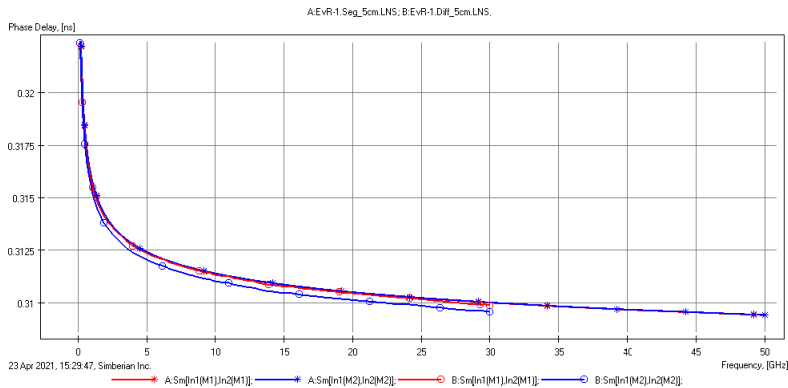


The difference in the common mode impedance seems a little larger (blue line is 5 cm model without the launches):

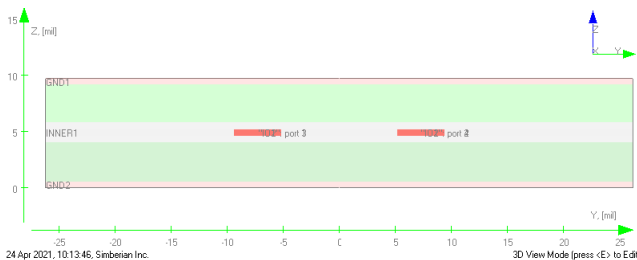


Though, we cannot do the accurate comparison due to the single-ended segments that lead to the differential section in both differential and common mode cases – it creates some offset of the impedance observed on TDR. Overall such approximate comparison and outcome look like perfectly acceptable for practical purpose.

A closer look at the modal phase delays reveal small defect in the model with homogeneous dielectrics (same dielectric properties for core and prepreg) – phase delays of the GMS transmission parameters for the odd (#1) and even (#2) modes are identical in the model, but are different in the measured data:



How to deal with that? Not a big deal, if far end crosstalk not needed. Alternatively, define and identify models for 2 dielectrics separately. Inhomogeneous dielectric leads to differences of modal propagation speed. It can be easily done in Simbeor Matlab scripts with `WithResinLayer = true`. Cross-section of transmission line will have 3 layers of dielectric as illustrated below (prepreg-resin-core):

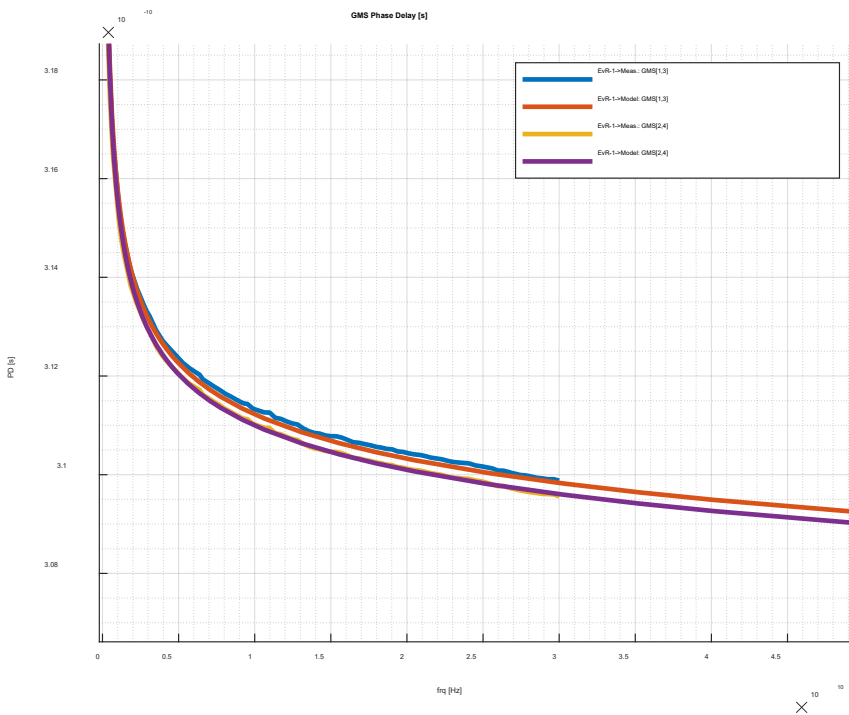


Core and prepreg dielectrics will have properties different from resin dielectric. Resin is not actually resin in this context – it is model for the “butter-coating” layer that usually has resin and a filler (fabric powder for instance or something else). Core and prepreg with the same fabric should have the same properties (cured prepreg should have the same properties as the core). With `WithResinLayer = true` and resin treated as different, the results will be as follows next:

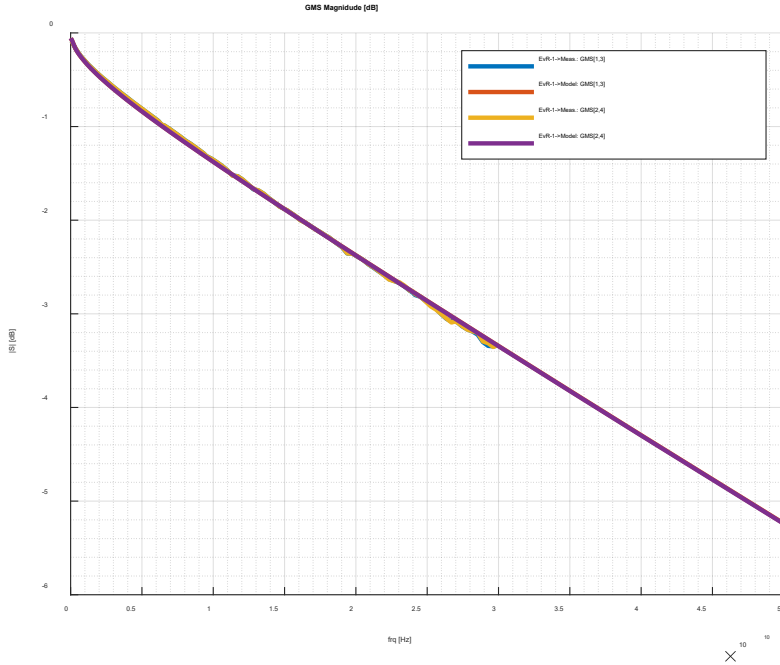
```

EvR - 1 Final Results :
Name: {'Meg-6-C'}
Model: 'WidebandDebye'
Dk : 3.2939
LT : 0.0030
Frq : 1.0000e+09
Name : {'Meg-6-P'}
Model: 'WidebandDebye'
Dk : 3.2939
LT : 0.0030
Frq : 1.0000e+09
Name : {'Resin'}
Model: 'WidebandDebye'
Dk : 3.5371
LT : 0.0030
Frq : 1.0000e+09
Name : {'Copper'}
RR : 1.2000
Roughness : [1x1 struct]
Model : 'HurayBracken'
SR : 0.1000
RF : 10.9729
    
```

Core and prepreg dielectrics have slightly smaller DK and resin dielectric slightly larger DK comparing to the homogeneous dielectric case (same dielectric constant for the core and prepreg). Is this reasonable result? Well, it produces the model with different phase delays for odd and even modes that correlates well with the measured results as illustrated next (GMS parameter [1,3] is the odd mode transmission and [2,4] is the even mode transmission, phase delay in 5 cm segment):



The modal insertion loss does not change – both modes have practically the same attenuation as shown below:



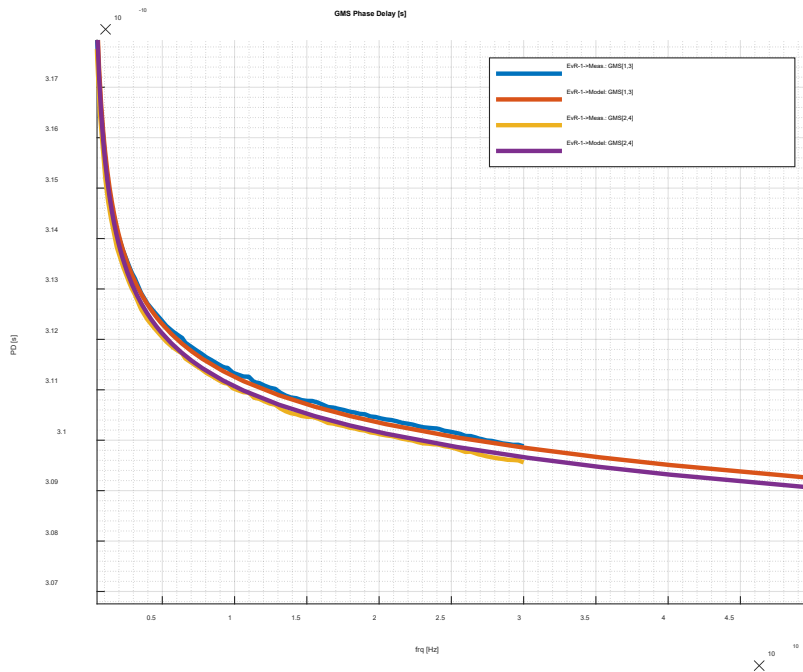
Another option is to have core dielectric different from the prepreg and resin (case of prepreg layer having different fabric style for instance). It requires just a single change in the problem description script:

```
Identify.Dielectric.Names = { diel1, diel2, diel3 }; %dielectric models to identify
%FIRST GROUP OF DIELECTRICS FOR EVEN MODE, SECOND FOR THE ODD MODE(FILL)
Identify.Dielectric.Identity = [1, 2, 2]; %same properties for 2 dielectrics and
different for another one
```

diel2 and diel3 will have properties different from diel1 in this case. The outcome is as follows (larger difference in dielectric properties):

```
EvR - 1 Final Results :
Name: {'Meg-6-C'}
Model: 'WidebandDebye'
Dk : 3.0810
LT : 0.0030
Frq : 1.0000e+09
Name: {'Meg-6-P'}
Model: 'WidebandDebye'
Dk : 3.5218
LT : 0.0030
Frq : 1.0000e+09
Name: {'Resin'}
Model: 'WidebandDebye'
Dk : 3.5218
LT : 0.0030
Frq : 1.0000e+09
Name: 'Copper'
RR : 1.2000
Roughness : [1x1 struct]
Model: 'HurayBracken'
SR : 0.1000
RF : 10.9729
```

It also split the modal phase delay as shown on this plot:



Note that the model with inhomogeneous dielectric may be needed only for precise evaluation of the crosstalk. Models with homogeneous dielectrics are simpler and sufficient for channel performance evaluation without crosstalk. Also, the frequency continuous models can be used up to 50 GHz and may be even beyond that frequency.

Final Notes

Material model identification is a difficult task with multiple steps. It requires design of test fixtures, manufacturing of test boards and accurate measurements. Those steps are not covered in this article – see examples of what can go wrong in “expectation vs. reality” experiment [9] and [10]. Getting everything right may be tedious and time consuming process, but **it is necessary condition for the analysis to measurement correlation**. No other way. This is especially important for the conductor roughness models. The copper foil surface is roughened by two manufacturers – one side by the copper foil manufacturer and another by PCB manufacturer. No way to get such models at one location so far. **Thus, build the test fixtures (two line segments) and measure S-parameters!** From the point when you get good test fixtures and high-quality measurements, **Simbeor SDK provides tools to facilitate dielectric and conductor roughness identification process with scripting in Matlab, C and Python.**

The material model identification can be done in Matlab (run “..\Simbeor SDK Examples\matlab\AdvMaterialKit\test_amk.m”), as used in this article. Same examples are provided in C (see examples at “..\Simbeor SDK Examples\cpp\testsdk_cpp\testsdk_cpp\test_gms.cpp”) and Python (see examples at “..\Simbeor SDK Examples\python\test_gms.py”). The model fitting algorithm for Matlab is located at “..\Simbeor SDK Examples\matlab\AdvMaterialKit\fitModelToMeasuredGMS.m”. Step through it with different scenarios of the loss separation to see how it works. Documentation for each Simbeor SDK function can be found at <https://kb.simberian.com/SimbeorSDKMatlab.php> for Matlab and C and <https://kb.simberian.com/SimbeorSDKPython.php> for Python.

Run times for all examples used here are within a few seconds. It allows quick experiments and mass-processing of large batches of data, to collect material model data to evaluate distribution of material parameters as it was done in [11] and [12]. Filling of data structures can be easily

parameterized to process data defined in a table. Run time may be larger if Simbeor 3DML or 3DTF solvers are used for modeling – it is recommended for microstrip cases with high-frequency dispersion (relatively wide traces).

To fit measured data up to 70 GHz and above, accurate conductor roughness model may need 2 levels. Additional level can be easily added into the roughness description structure as parameters “RF1” and “SR1” and added as parameter #2 during the optimization. The identification of 4 roughness correction coefficients can be done simultaneously or by bandwidth separation similar to the separation by bandwidth method used for dielectric and roughness losses. The first roughness level should have larger SR and optimized from 2-3 GHz to 40-50 GHz and the second level is located on top of the first one (multiplicative model) and optimized above 50-60 GHz.

References

1. Y. Shlepnev, A. Neves, T. Dagostino, S. McMorrow, Measurement-Assisted Electromagnetic Extraction of Interconnect Parameters on Low-Cost FR-4 boards for 6-20 Gb/sec Applications, DesignCon 2009 - #2009_03 at <https://www.simberian.com/AppNotes.php>
2. Y. Shlepnev, A. Neves, T. Dagostino, S. McMorrow, Practical identification of dispersive dielectric models with generalized modal S-parameters for analysis of interconnects in 6-100 Gb/s applications, DesignCpn 2010 - # 2010_01 at <https://www.simberian.com/AppNotes.php>
3. Y. Shlepnev, Broadband material model identification with GMS-parameters, 2015 IEEE 24th Conference on Electrical Performance of Electronic Packaging and Systems, October 25-28, 2015, San Jose, CA. - extended version of paper is #2015_02 at <https://www.simberian.com/AppNotes.php>
4. Djordjevic, R.M. Biljic, V.D. Likar-Smiljanic, T.K.Sarkar, Wideband frequency domain characterization of FR-4 and time-domain causality, IEEE Trans. on EMC, vol. 43, N4, 2001, p. 662-667.
5. C. Svensson, G.E. Dermer, Time domain modeling of lossy interconnects, IEEE Trans. on Advanced Packaging, May 2001, N2, Vol. 24, pp.191-196.
6. J. E. Bracken, A Causal Huray Model for Surface Roughness, DesignCon 2012
7. V. Dmitriev-Zdorov, B Simonovich, I. Kochikov, A Causal Conductor Roughness Model and its Effect on Transmission Line Characteristics, DesignCon 2018.
8. Y. Shlepnev, Unified approach to interconnect conductor surface roughness modelling, 2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS2017), October 15-18, 2017, San Jose, CA. - #2017_04 at <https://www.simberian.com/AppNotes.php>
9. M. Marin, Y. Shlepnev, 40 GHz PCB Interconnect Validation: Expectation vs. Reality, DesignCon2018, January 31, 2018, Santa Clara, CA. - #2018_01 at <https://www.simberian.com/AppNotes.php>
10. M. Marin, Y. Shlepnev, Systematic approach to PCB interconnects analysis to measurement validation, 2018 IEEE Symposium on Electromagnetic Compatibility, Signal and Power Integrity, July 30- August 3, 2018, Long Beach Convention Center, Long Beach, CA. - #2018_07 at <https://www.simberian.com/AppNotes.php>
11. A. Manukovsky, Y. Shlepnev, Effect of PCB Fabrication Variations on Interconnect Loss, Delay, Impedance & Identified Material Models for 56-Gbps Interconnect Designs, **DesignCon 2019**, January 30, 2019, Santa Clara, CA. - #2019_01 at <https://www.simberian.com/AppNotes.php>

12. A. Manukovsky, Y. Shlepnev, Measurement-assisted extraction of PCB interconnect model parameters with fabrication variations, 2019 IEEE 28st Conference on Electrical Performance of Electronic Packaging and Systems, Oct. 6-9, 2019, (EPEPS 2019), October 6-9, 2019, Montreal, Canada. - #2019_04 at <https://www.simberian.com/AppNotes.php>