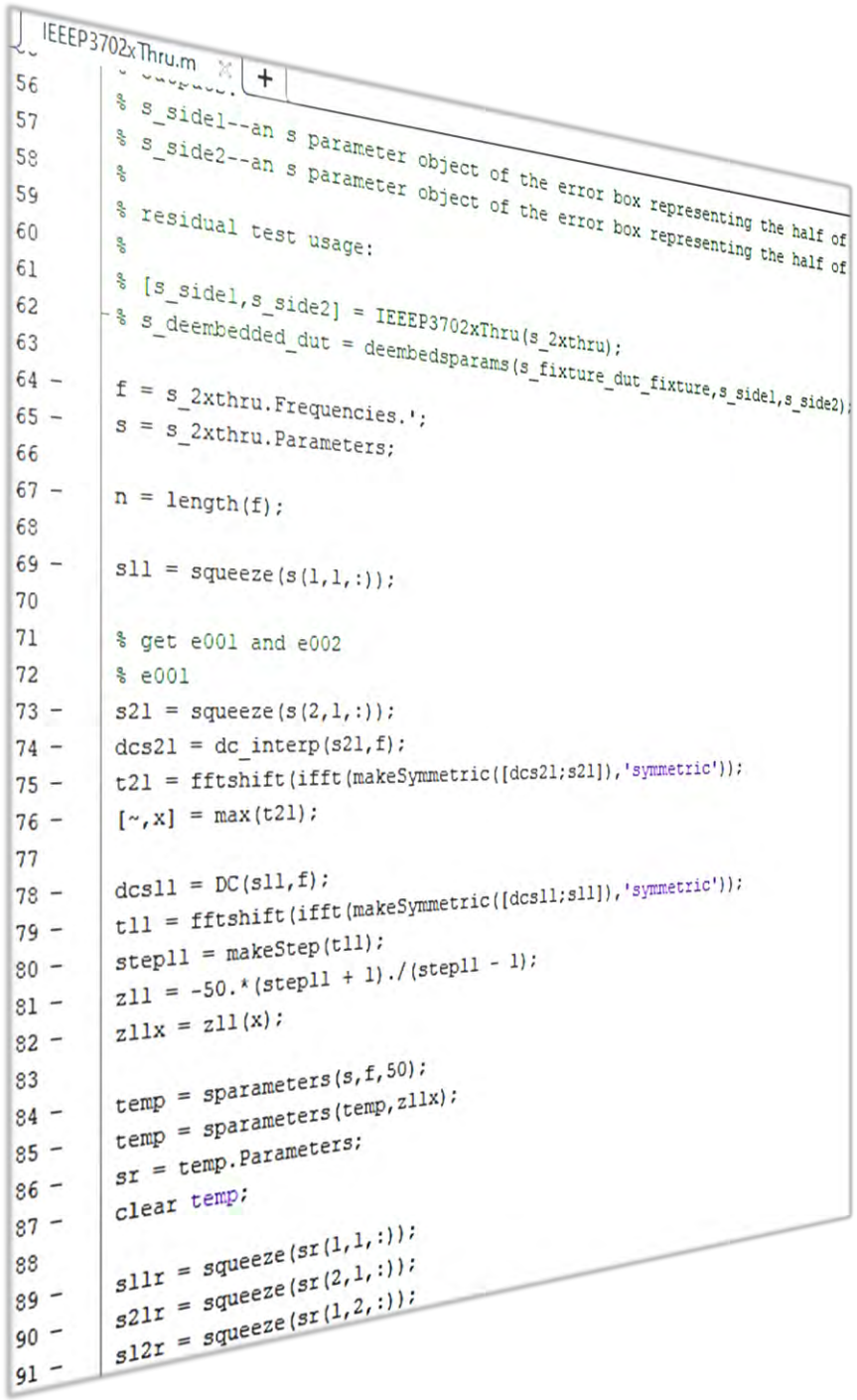
Open-Source De-embedding

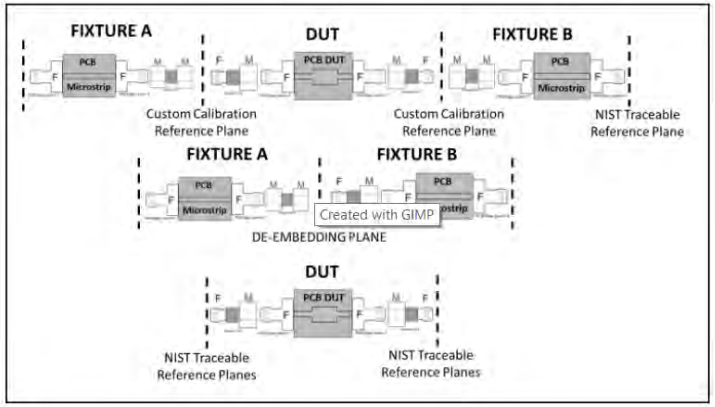


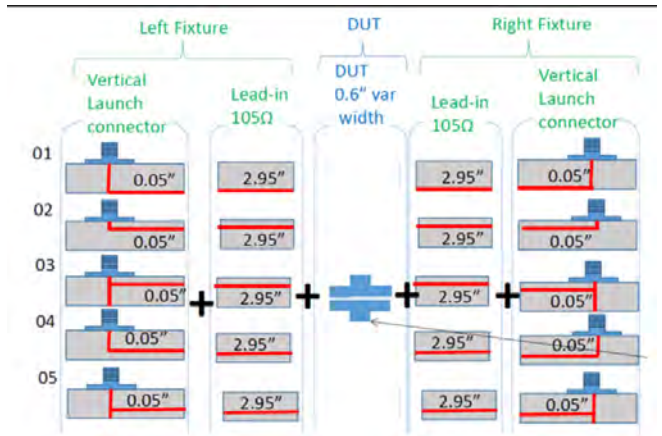
Agenda

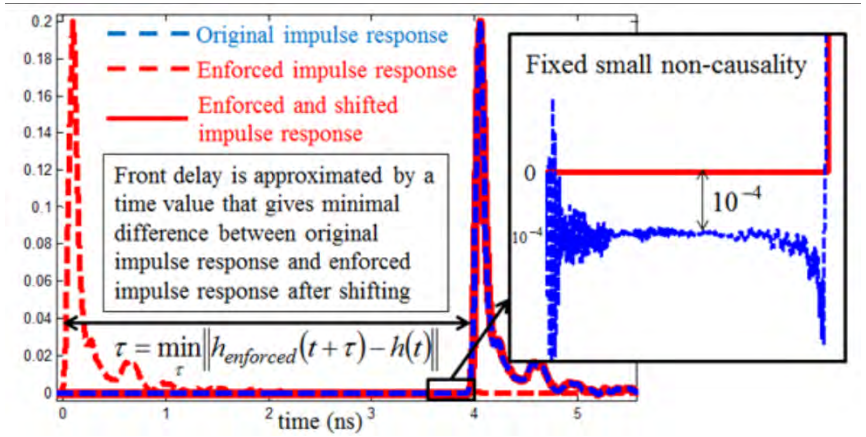
- **MOTIVATION**
- **QUICK OVERVIEW OF DE-EMBEDDING**
- **WHERE DO I FIND THE CODE?**
- **HOW DO I USE THE CODE?**
- **HOW MUCH DOES IT COST?**
- **FUTURE ENHANCEMENTS**
- **HOW CAN I CONTRIBUTE TO THE ALGORITHM?**

Motivation

IEEE 370-2020 - IEEE Standard for Electrical Characterization of Printed Circuit Board and Related Interconnects at Frequencies up to 50 GHz

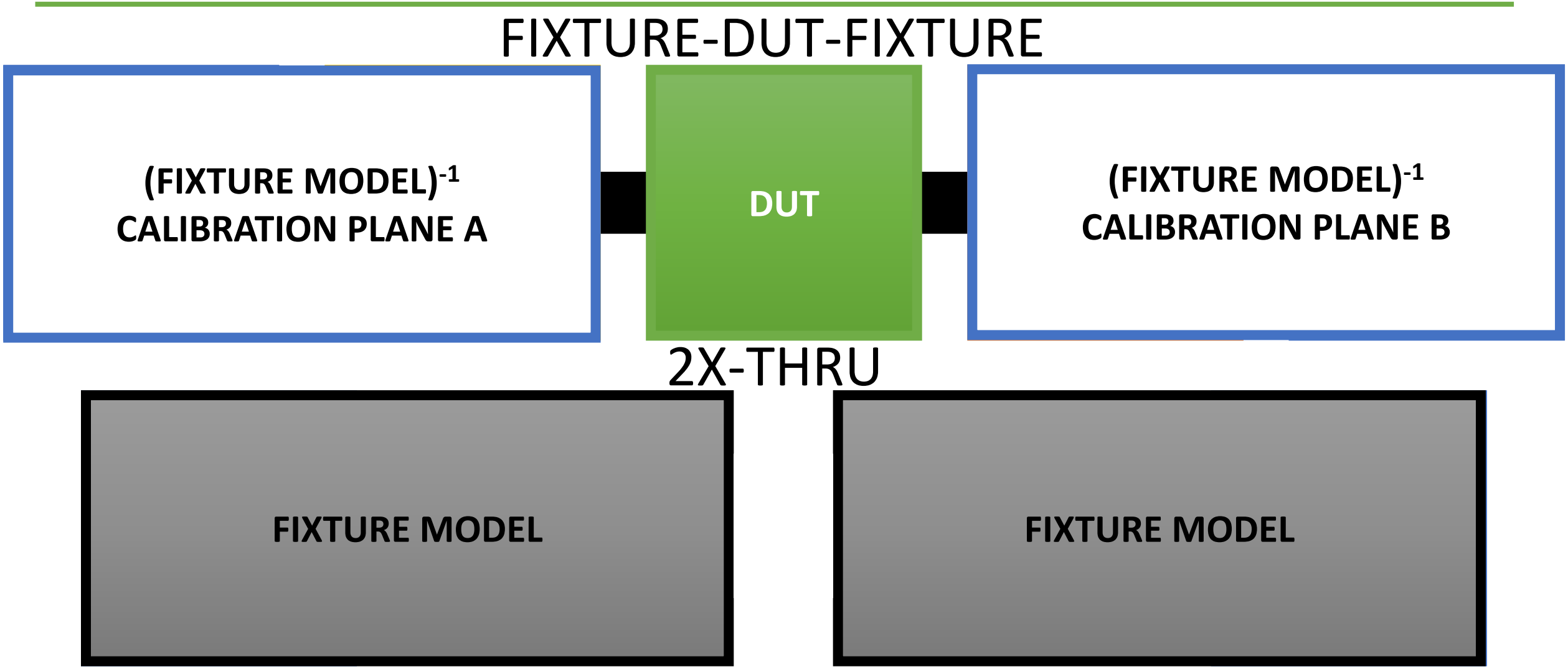






https://standards.ieee.org/standard/370-2020.html

Quick Overview of De-embedding

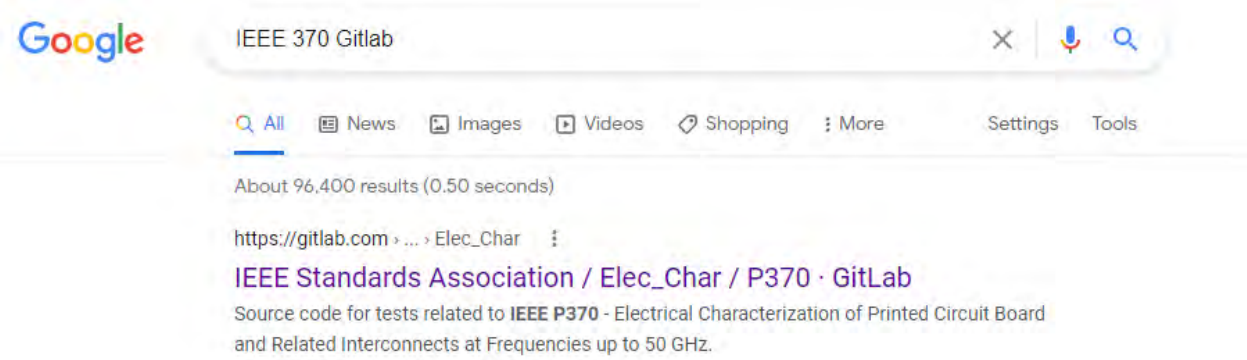


Where Do I find the Code?

Website:

https://gitlab.com/IEEE-SA/ElecChar/P370

Google Search:

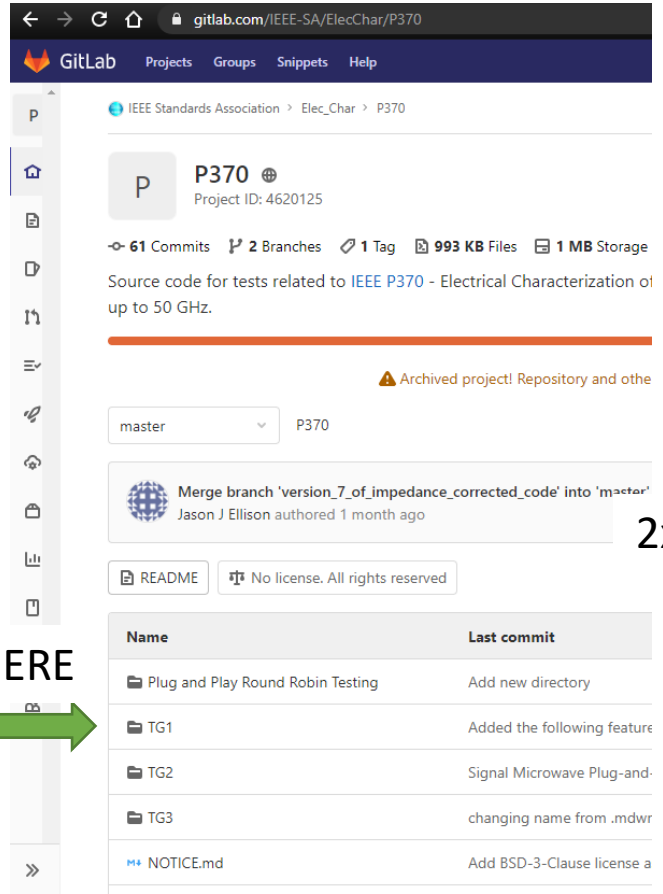


New website (IEEE membership required):

https://opensource.ieee.org/elec-char/ieee-370

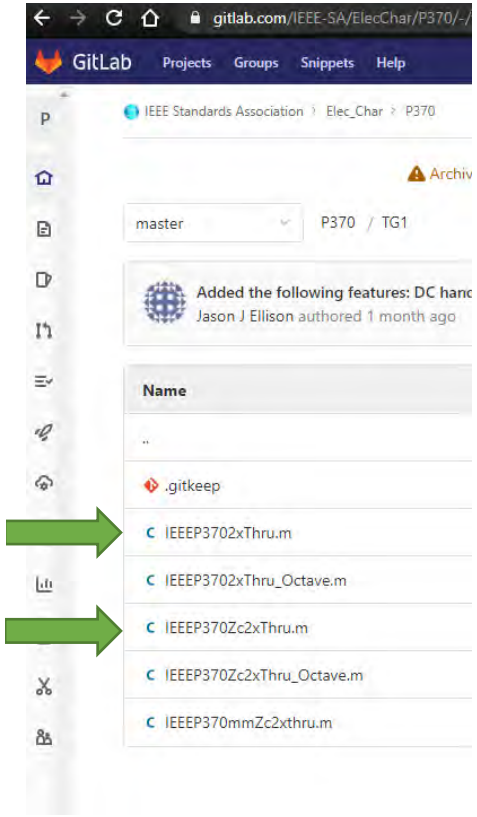
Where Do I find the Code?

CLICK HERE



2x-thru algorithm

ZC algorithm



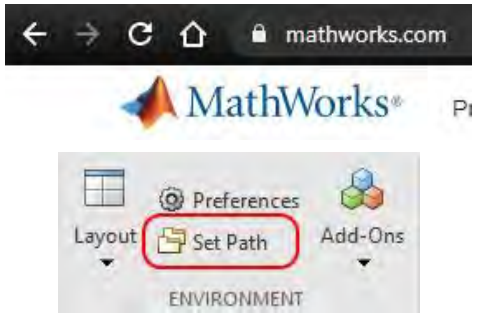
The **2x-thru algorithm** creates error-boxes which match the 2x-thru. perfectly.
Usage cases:

- Where speed matters
- The 2x-thru and Fixture-DUT-Fixture traces are well matched.

The **ZC algorithm** creates error-boxes which match the Fixture.
Usage cases:

- Causality is important.
- Reference plane movement is required.
- Measurement-Simulation correlation.

How do I use the code?




```
>> s2x = sparameters("example_2x_thru.s2p");
```

```
>> fdf = sparameters("example_fdf.s2p");
```

```
>> [side1,side2] = IEEEP3702xThru(s2x);
```

OR

```
>> [side1,side2] = IEEEP370Zc2xThru(s2x,fdf);
```

```
>> dut = deembedsparams(fdf,side1,side2);
```

- Install MATLAB and the RF-TOOLBOX
- Put the functions in your MATLAB path
- Load a 2x-thru
- Load a Fixture-DUT-Fixture
- Extract the error-boxes
- Remove the error-boxes

How do I use the code?

```
[side1,side2] = IEEEP370Zc2xThru(s2x,fdf,"z0",45);
```

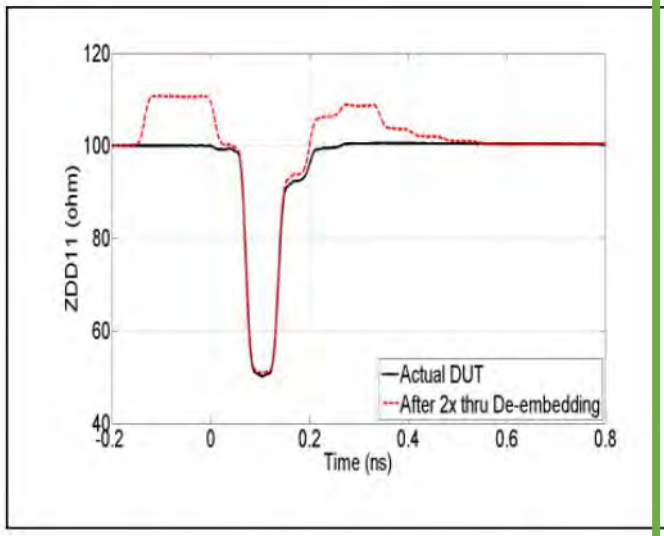
Options:

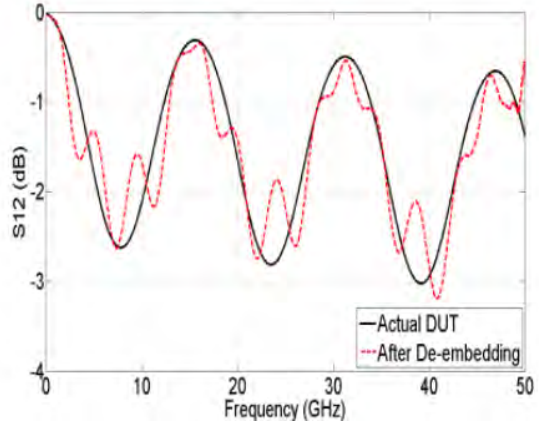
The options are setup as name-value pairs.

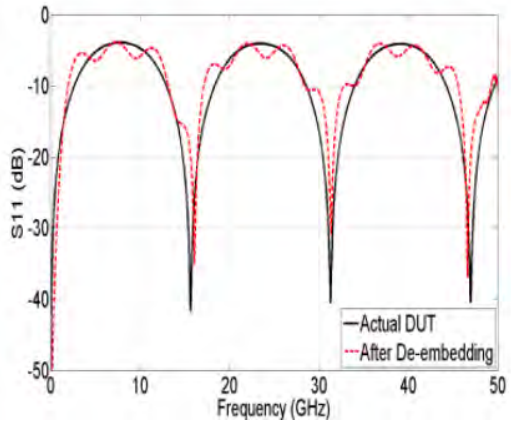
NAME	VALUE
“z0”	Reference Impedance | Default: 50
“bandwidth”	Fit attenuation to a limited bandwidth| Default: 0
“view”	View the de-embedding process| Default: false
“pullback”	Number of discrete points to omit| Default: 0
“pullback1”	Pullback on side1 only| Default: 0
“pullback2”	Pullback on side2 only| Default: 0
“side1”	Enable side1 de-embedding| Default: true
“side2”	Enable side2 de-embedding| Default: true

How do I use the code?

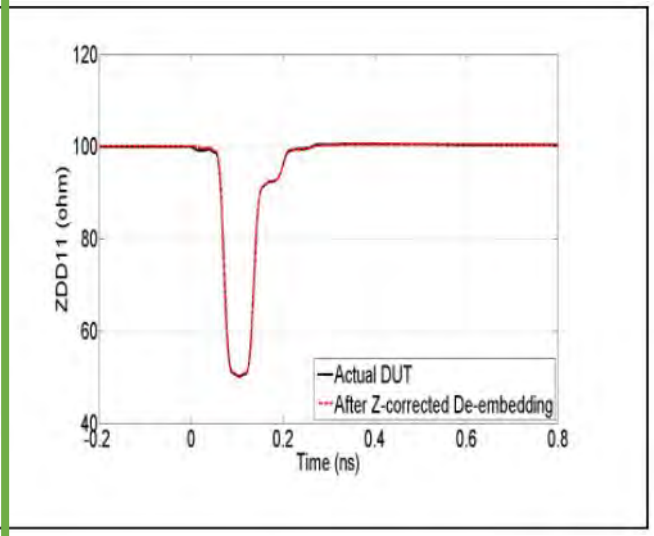
2x-Thru De-embedding

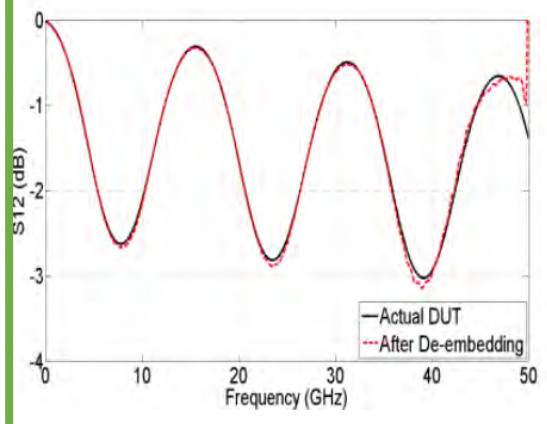


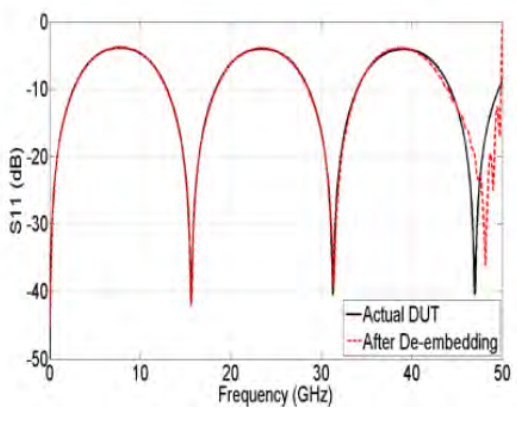




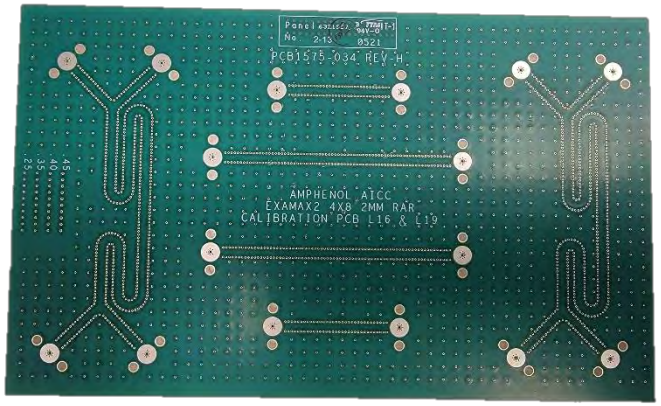
Impedance Corrected 2x-Thru De-embedding

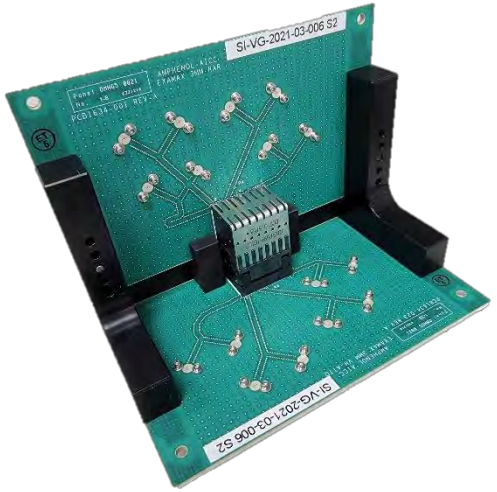


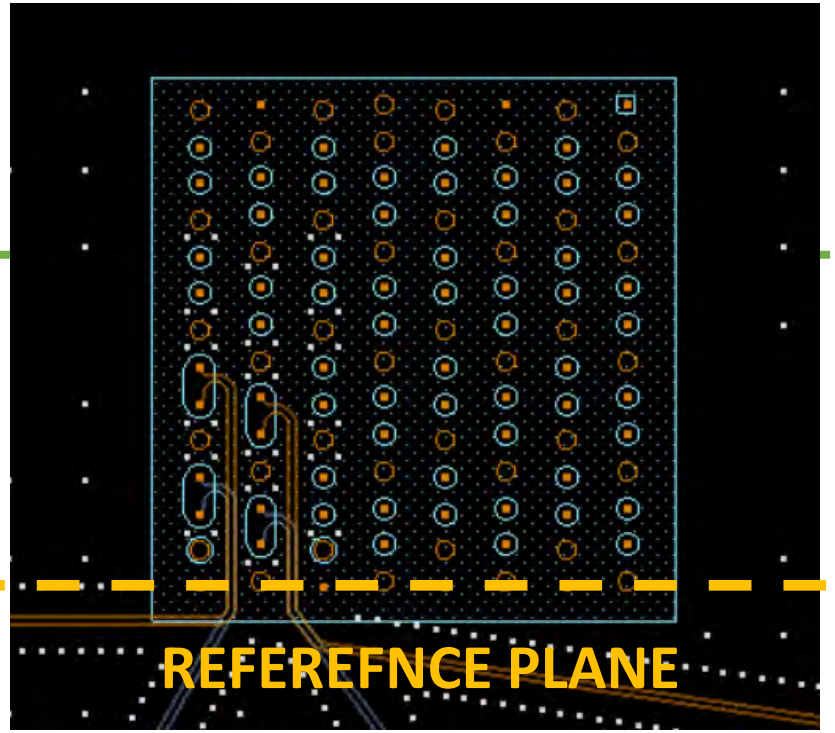


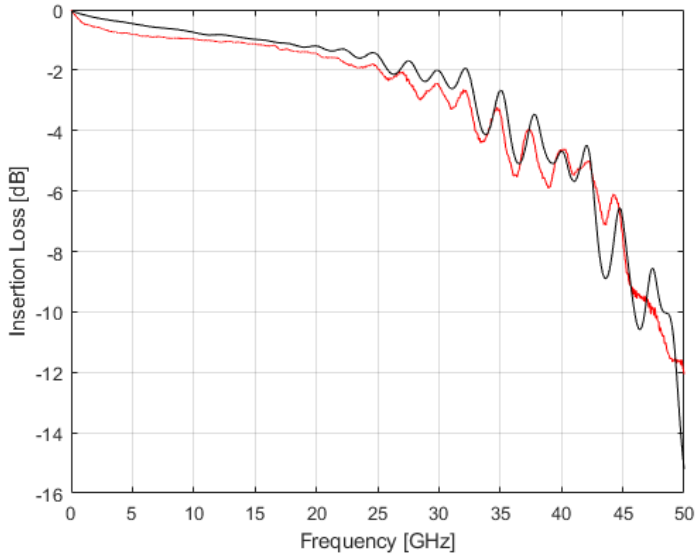


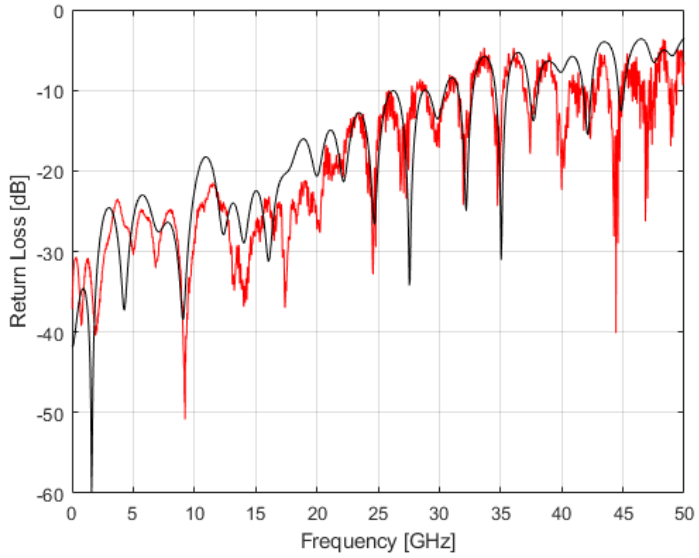
How do I use the code?

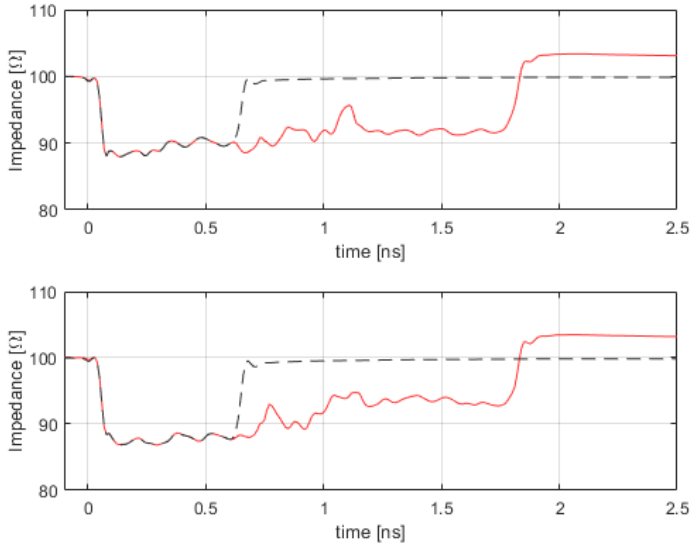






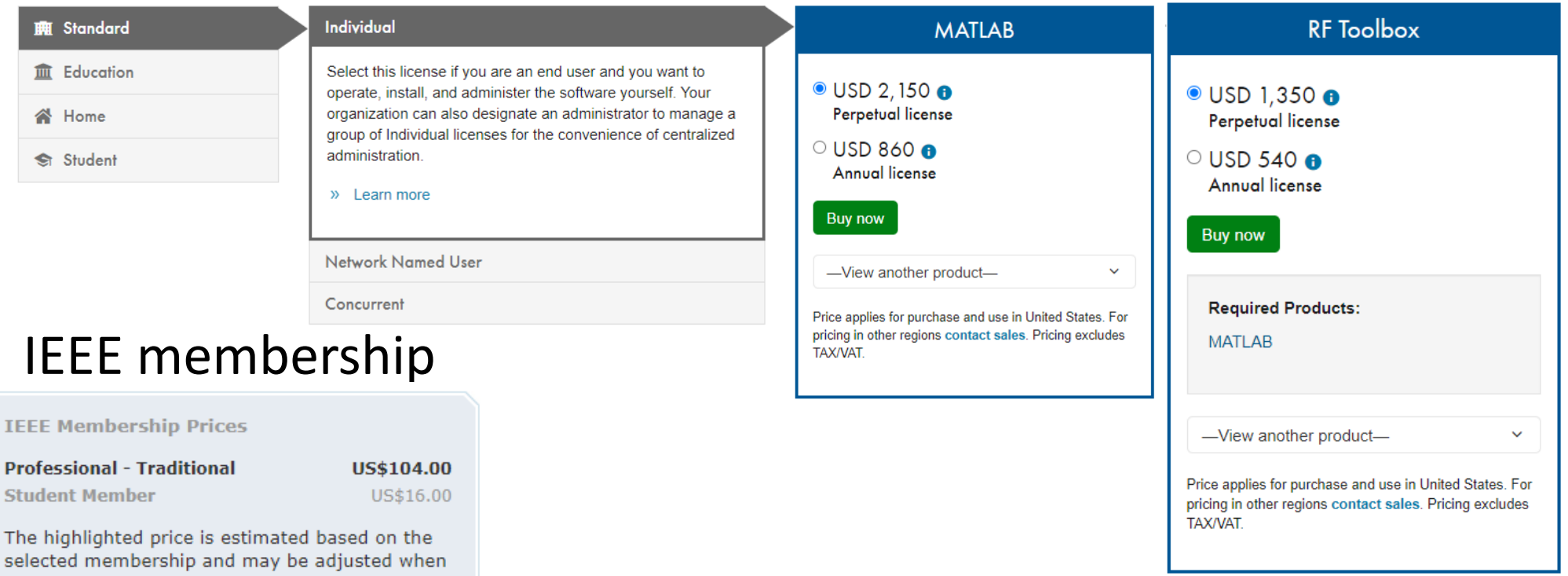






How much does it cost?

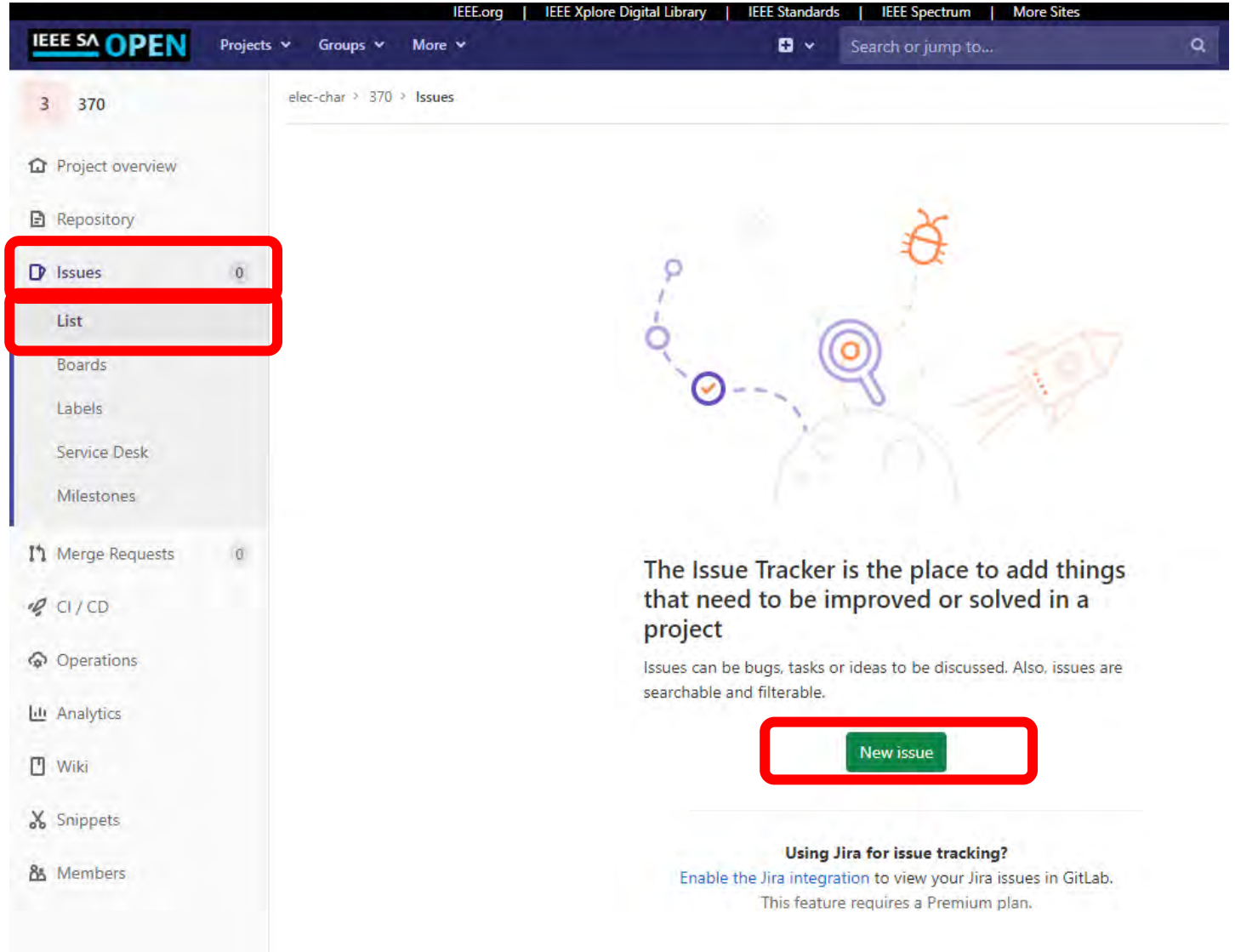
The cost of a MATLAB license with RF-toolbox. (not a paid promotion)



IEEE membership

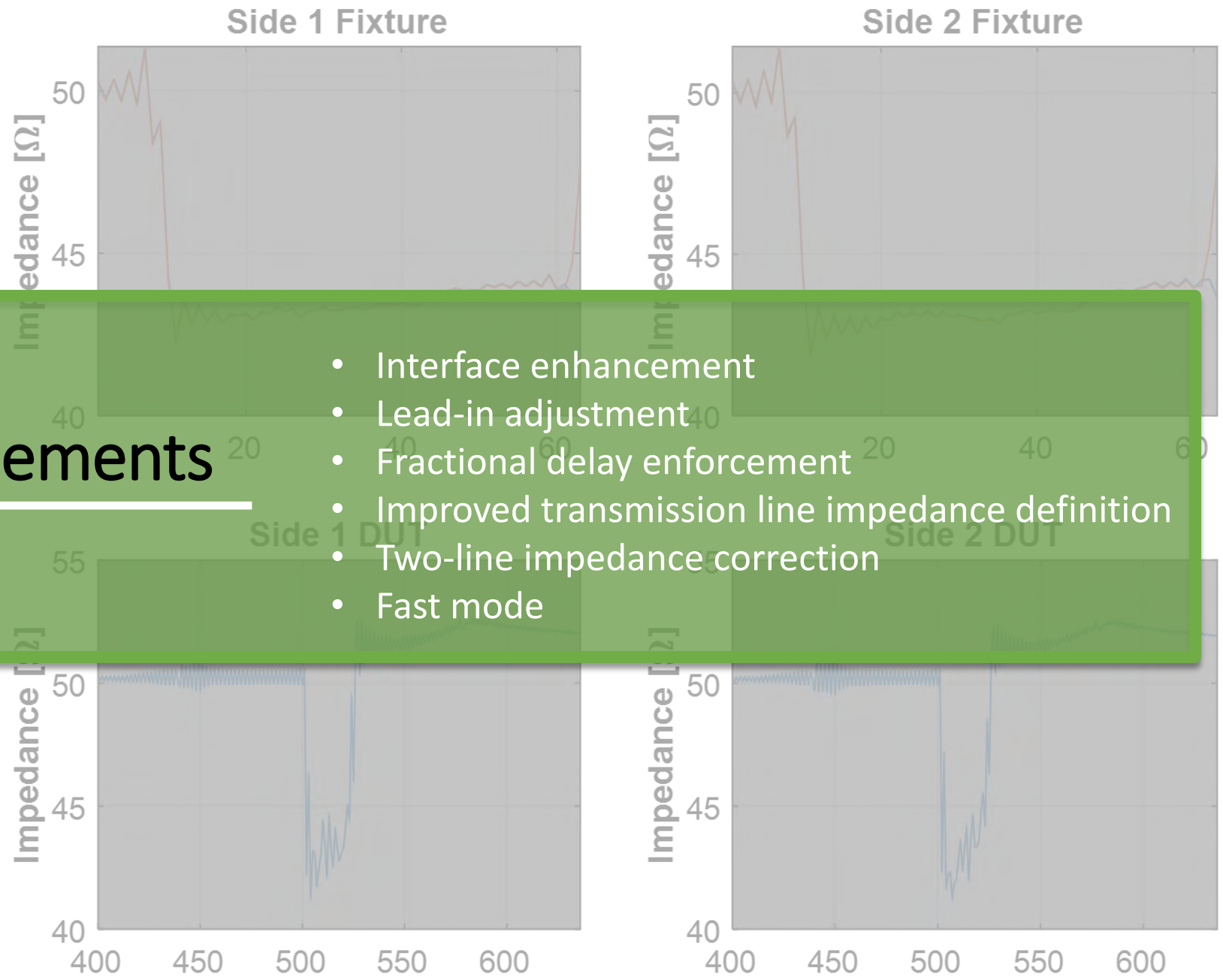
Example of standard pricing. Recorded 4/5/2021
Prices do vary.

How can I Contribute?



Jason Ellison
jason.j.Ellison@ieee.org

Josh Gay
j.gay@ieee.org



Future Enhancements

- Interface enhancement
- Lead-in adjustment
- Fractional delay enforcement
- Improved transmission line impedance definition
- Two-line impedance correction
- Fast mode

THANK YOU!


IEEEP3702xThru.m
% s_side1--an s parameter object of the error box representing the half of
% s_side2--an s parameter object of the error box representing the half of
% residual test usage:
% [s_side1,s_side2] = IEEEP3702xThru(s_2xthru);
% s_deembedded_dut = deembedsparams(s_fixture_dut_fixture,s_side1,s_side2);
f = s_2xthru.Frequencies.';
s = s_2xthru.Parameters;
n = length(f);
s11 = squeeze(s(1,1,:));
% get e001 and e002
% e001
s21 = squeeze(s(2,1,:));
dcs21 = dc_interp(s21,f);
t21 = fftshift(ifft(makeSymmetric([dcs21;s21]),'symmetric'));
[~,x] = max(t21);
dcs11 = DC(s11,f);
t11 = fftshift(ifft(makeSymmetric([dcs11;s11]),'symmetric'));
step11 = makeStep(t11);
z11 = -50.*(step11 + 1)./(step11 - 1);
z11x = z11(x);
temp = sparameters(s,f,50);
temp = sparameters(temp,z11x);
sr = temp.Parameters;
clear temp;
s11r = squeeze(sr(1,1,:));
s21r = squeeze(sr(2,1,:));
s12r = squeeze(sr(1,2,:));