Part I

Open-Source Software

Outline

Open-source Software

Open-source software mostly started in the 90's. Probably a reaction to Microsoft's dominance back then. Linux was released in 1991. Today lots of people contribute to open-source software. The reasons are:

- Allows people to establish expertise by making something that is useful, can be read, understood, modified and contributed to, and can displace dominant closed-source applications.
- Is sometimes a by-product of other work.
- In research: allows true duplication of experiments.

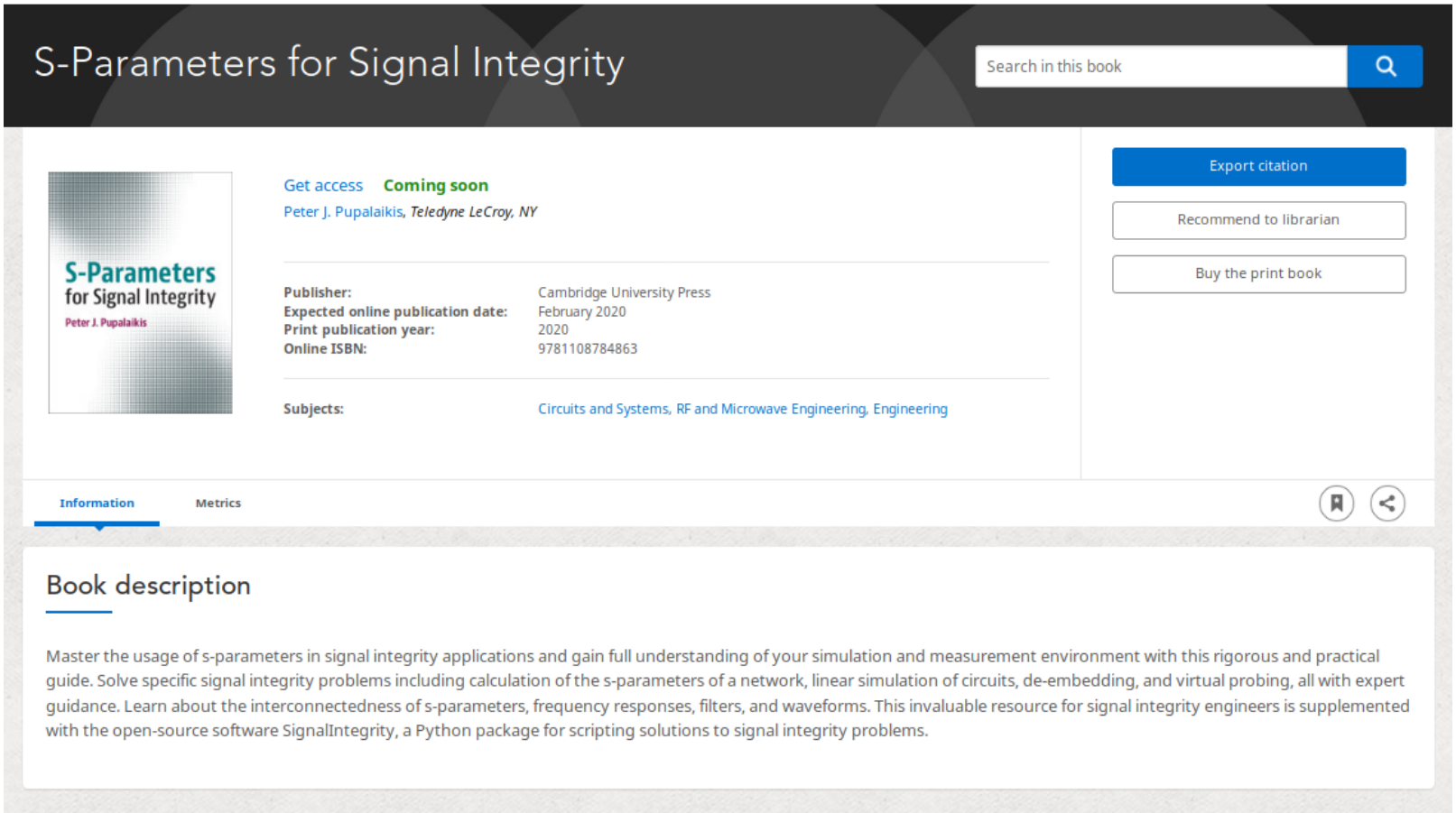
Some examples that are applicable to signal integrity:

- Qucs
- SPICE
- PyBert and PyIBIS-AMI
- Elmer FEM Solver

SignalIntegrity

S-Parameters for Signal Integrity Textbook

Approximately ten years of development. Published in Feb. 2020.

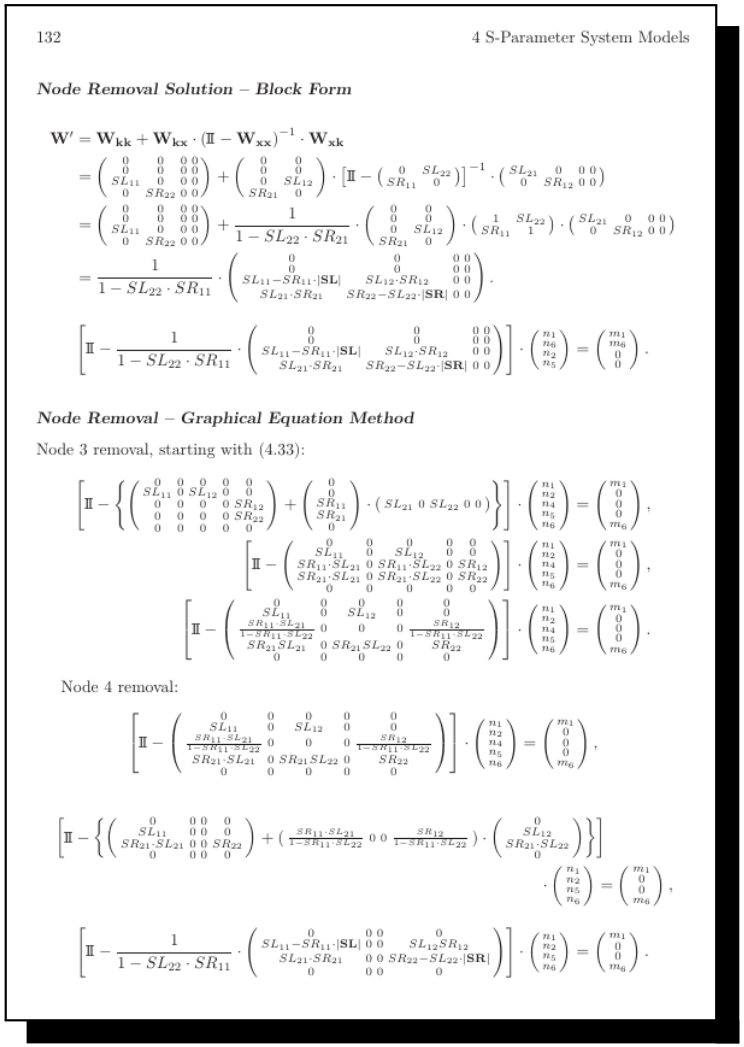


A Disappointing Fact...

- During the writing of this book, it became apparent that while all of the math and explanations were interesting, aside from a handful of people, no one would actually use the book!
- Thus, the idea for *SignalIntegrity* was born.

Page 132

The page on the right is multiple solution methods of two cascaded two-port networks.



132 4 S-Parameter System Models

Node Removal Solution – Block Form

$$\mathbf{W}' = \mathbf{W}_{\mathbf{kk}} + \mathbf{W}_{\mathbf{kx}} \cdot (\mathbb{I} - \mathbf{W}_{\mathbf{xx}})^{-1} \cdot \mathbf{W}_{\mathbf{xk}}$$
$$= \begin{pmatrix} 0 & 0 & 0\,0 \\ 0 & 0 & 0\,0 \\ SL_{11} & 0 & 0\,0 \\ 0 & SR_{22} & 0\,0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & SL_{12} \\ SR_{21} & 0 \end{pmatrix} \cdot \left[\mathbb{I} - \begin{pmatrix} 0 & SL_{22} \\ SR_{11} & 0 \end{pmatrix}\right]^{-1} \cdot \begin{pmatrix} SL_{21} & 0 & 0\,0 \\ 0 & SR_{12} & 0\,0 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 & 0\,0 \\ 0 & 0 & 0\,0 \\ SL_{11} & 0 & 0\,0 \\ 0 & SR_{22} & 0\,0 \end{pmatrix} + \frac{1}{1 - SL_{22} \cdot SR_{21}} \cdot \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & SL_{12} \\ SR_{21} & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & SL_{22} \\ SR_{11} & 1 \end{pmatrix} \cdot \begin{pmatrix} SL_{21} & 0 & 0\,0 \\ 0 & SR_{12} & 0\,0 \end{pmatrix}$$
$$= \frac{1}{1 - SL_{22} \cdot SR_{11}} \cdot \begin{pmatrix} 0 & 0 & 0\,0 \\ 0 & 0 & 0\,0 \\ SL_{11} - SR_{11} \cdot |\mathbf{SL}| & SL_{12} \cdot SR_{12} & 0\,0 \\ SL_{21} \cdot SR_{21} & SR_{22} - SL_{22} \cdot |\mathbf{SR}| & 0\,0 \end{pmatrix}.$$

$$\left[\mathbb{I} - \frac{1}{1 - SL_{22} \cdot SR_{11}} \cdot \begin{pmatrix} 0 & 0 & 0\,0 \\ 0 & 0 & 0\,0 \\ SL_{11} - SR_{11} \cdot |\mathbf{SL}| & SL_{12} \cdot SR_{12} & 0\,0 \\ SL_{21} \cdot SR_{21} & SR_{22} - SL_{22} \cdot |\mathbf{SR}| & 0\,0 \end{pmatrix}\right] \cdot \begin{pmatrix} n_1 \\ n_6 \\ n_2 \\ n_5 \end{pmatrix} = \begin{pmatrix} m_1 \\ m_6 \\ 0 \\ 0 \end{pmatrix}.$$

Node Removal – Graphical Equation Method

Node 3 removal, starting with (4.33):

$$\left[\mathbb{I} - \left\{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ SL_{11} & 0 & SL_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & SR_{12} \\ 0 & 0 & 0 & 0 & SR_{22} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ SR_{11} \\ SR_{21} \\ 0 \end{pmatrix} \cdot \begin{pmatrix} SL_{21} & 0 & SL_{22} & 0 & 0 \end{pmatrix}\right\}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_4 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ 0 \\ m_6 \end{pmatrix},$$

$$\left[\mathbb{I} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ SL_{11} & 0 & SL_{12} & 0 & 0 \\ SR_{11} \cdot SL_{21} & 0 & SR_{11} \cdot SL_{22} & 0 & SR_{12} \\ SR_{21} \cdot SL_{21} & 0 & SR_{21} \cdot SL_{22} & 0 & SR_{22} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_4 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ 0 \\ m_6 \end{pmatrix},$$

$$\left[\mathbb{I} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ SL_{11} & 0 & SL_{12} & 0 & 0 \\ \frac{SR_{11} \cdot SL_{21}}{1 - SR_{11} \cdot SL_{22}} & 0 & 0 & 0 & \frac{SR_{12}}{1 - SR_{11} \cdot SL_{22}} \\ SR_{21} \cdot SL_{21} & 0 & SR_{21} SL_{22} & 0 & SR_{22} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_4 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ 0 \\ m_6 \end{pmatrix}.$$

Node 4 removal:

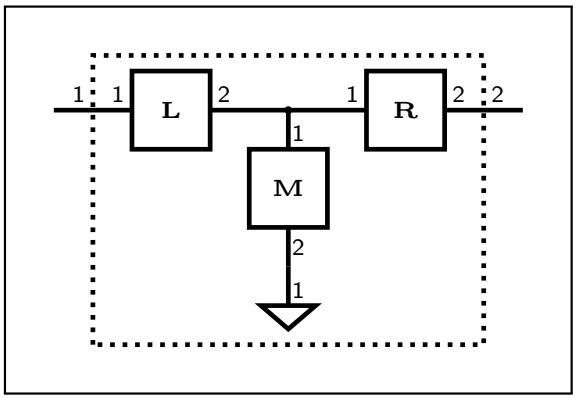
$$\left[\mathbb{I} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ SL_{11} & 0 & SL_{12} & 0 & 0 \\ \frac{SR_{11} \cdot SL_{21}}{1 - SR_{11} \cdot SL_{22}} & 0 & 0 & 0 & \frac{SR_{12}}{1 - SR_{11} \cdot SL_{22}} \\ SR_{21} \cdot SL_{21} & 0 & SR_{21} SL_{22} & 0 & SR_{22} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_4 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ 0 \\ m_6 \end{pmatrix},$$

$$\left[\mathbb{I} - \left\{\begin{pmatrix} 0 & 0\,0 & 0 \\ SL_{11} & 0\,0 & 0 \\ SR_{21} \cdot SL_{21} & 0\,0 & SR_{22} \\ 0 & 0\,0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{SR_{11} \cdot SL_{21}}{1 - SR_{11} \cdot SL_{22}} & 0\,0 & \frac{SR_{12}}{1 - SR_{11} \cdot SL_{22}} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ SL_{12} \\ SR_{21} \cdot SL_{22} \end{pmatrix}\right\}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ m_6 \end{pmatrix},$$

$$\left[\mathbb{I} - \frac{1}{1 - SL_{22} \cdot SR_{11}} \cdot \begin{pmatrix} 0 & 0\,0 & 0 \\ SL_{11} - SR_{11} \cdot |\mathbf{SL}| & 0\,0 & SL_{12} SR_{12} \\ SL_{21} \cdot SR_{21} & 0\,0 & SR_{22} - SL_{22} \cdot |\mathbf{SR}| \\ 0 & 0\,0 & 0 \end{pmatrix}\right] \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_5 \\ n_6 \end{pmatrix} = \begin{pmatrix} m_1 \\ 0 \\ 0 \\ m_6 \end{pmatrix}.$$

Symbolic Solutions

Originally *SignalIntegrity* was only scripted and only symbolic.




```
device L 2
device R 2
device M 2
device G 1 ground
port 1 L 1 2 R 2
connect L 2 R 1 M 1
connect G 1 M 2
```

```
1  import SignalIntegrity.Lib as si
2  sdp = si.p.SystemDescriptionParser().File('SymbolicSolution3.txt')
3  ssps=si.sd.SystemSParametersSymbolic(sdp.SystemDescription(),size='small')
4  ssps.LaTeXSolution().Emit()
```

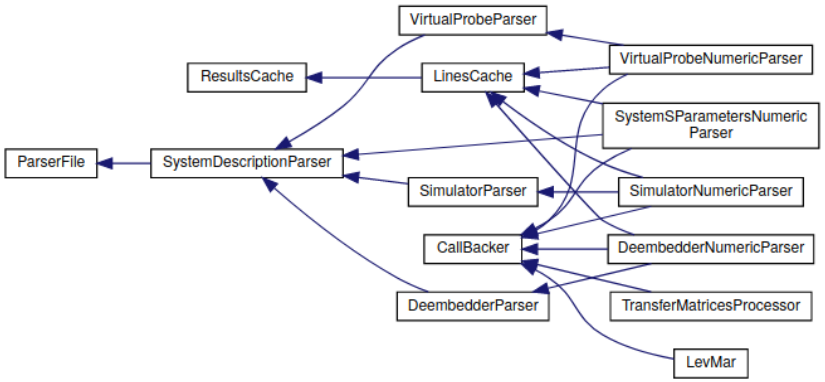
$$\mathbf{S} = \begin{pmatrix} L_{11} & 0 \\ 0 & R_{22} \end{pmatrix} + \begin{pmatrix} 0&0&0&0&0&0 & L_{12} & 0 \\ 0&0&0&0&0&R_{21} & 0 & 0 \end{pmatrix} \cdot \left[\mathbb{I} - \begin{pmatrix} 0&0&0&0&0&0&L_{22}&0 \\ 0&0&0&0&0&R_{11}&0&0 \\ 0&0&0&0&M_{12}&0&0&M_{11} \\ 0&0&0&0&M_{22}&0&0&M_{21} \\ 0&0&0&-1&0&0&0&0 \\ \frac{2}{3}&-\frac{1}{3}&\frac{2}{3}&0&0&0&0&0 \\ -\frac{1}{3}&\frac{2}{3}&\frac{2}{3}&0&0&0&0&0 \\ \frac{2}{3}&\frac{2}{3}&-\frac{1}{3}&0&0&0&0&0 \end{pmatrix} \right]^{-1} \cdot \begin{pmatrix} L_{21} & 0 \\ 0 & R_{12} \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \end{pmatrix}$$

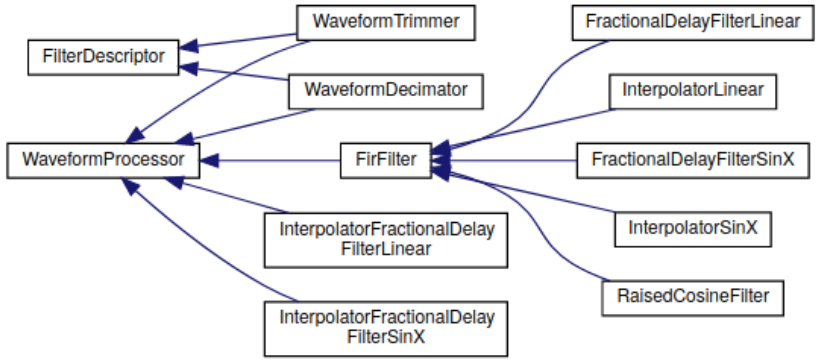
SignalIntegrity.Lib Package

... It's kind of like a Matlab library – but it's object oriented.

code	namespace	Description
si	SignalIntegrity.Lib	Top of SignalIntegrity.Lib Package
si.czt	SignalIntegrity.Lib.ChirpZTransform	Chirp z transform
si.cvt	SignalIntegrity.Lib.Conversions	Conversion Formulas
si.d	SignalIntegrity.Lib.Devices	Single Frequency Devices
si.fit	SignalIntegrity.Lib.Fit	Fitting Algorithms
si.fd	SignalIntegrity.Lib.FrequencyDomain	Frequency Domain
si.helper	SignalIntegrity.Lib.Helper	Helper functions and classes
si.ip	SignalIntegrity.Lib.ImpedanceProfile	Impedance profile
si.m	SignalIntegrity.Lib.Measurement	Measurment
si.m.cal	SignalIntegrity.Lib.Measurement.Calibration	Calibration algorithms
si.m.calkit	SignalIntegrity.Lib.Measurement.CalKit	Calibration kits
si.m.tdr	SignalIntegrity.Lib.Measurement.TDR	Time-domain reflectometry
si.p	SignalIntegrity.Lib.Parsers	Netlist parsers
si.p.dev	SignalIntegrity.Lib.Parsers.Devices	Netlist parser devices

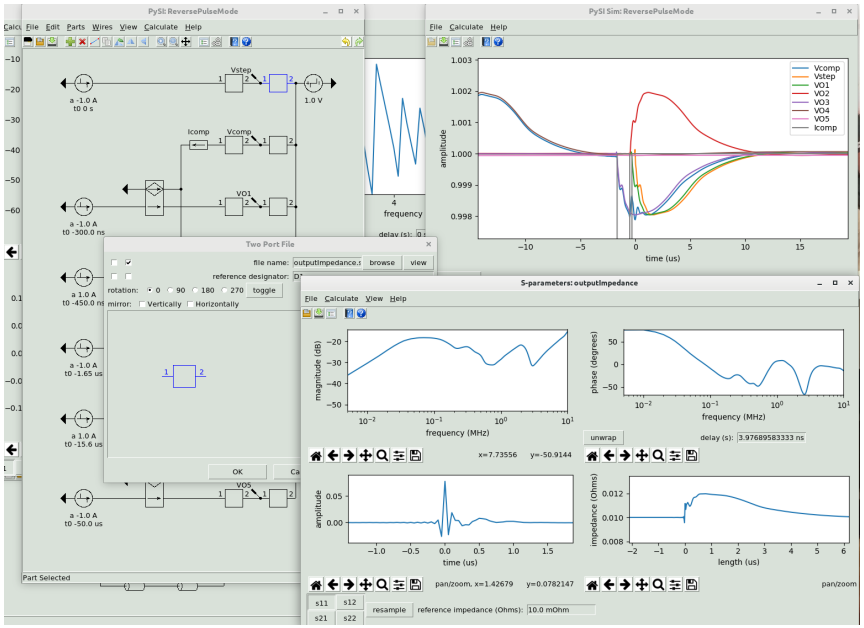
code	namespace	Description
si.prbs	SignalIntegrity.Lib.Prbs	Pseudo-random polynomials and waveforms
si.rat	SignalIntegrity.Lib.Rat	The RAT function
si.sp	SignalIntegrity.Lib.SParameters	S-parameters
si.sp.dev	SignalIntegrity.Lib.SParameters.Devices	S-parameter devices
si.spl	SignalIntegrity.Lib.Splines	Spline functions
si.sub	SignalIntegrity.Lib.SubCircuits	Netlist Subcircuits
si.sy	SignalIntegrity.Lib.Symbolic	Symbolic Solutions
si.sd	SignalIntegrity.Lib.SystemDescriptions	System descriptions
si.test	SignalIntegrity.Lib.Test	Test helpers
si.td	SignalIntegrity.Lib.TimeDomain	Time-domain
si.td.f	SignalIntegrity.Lib.TimeDomain.Filters	Filters
si.td.wf	SignalIntegrity.Lib.TimeDomain.Waveform	Waveforms
si.wl	SignalIntegrity.Lib.Wavelets	Wavelets





SignalIntegrity is...

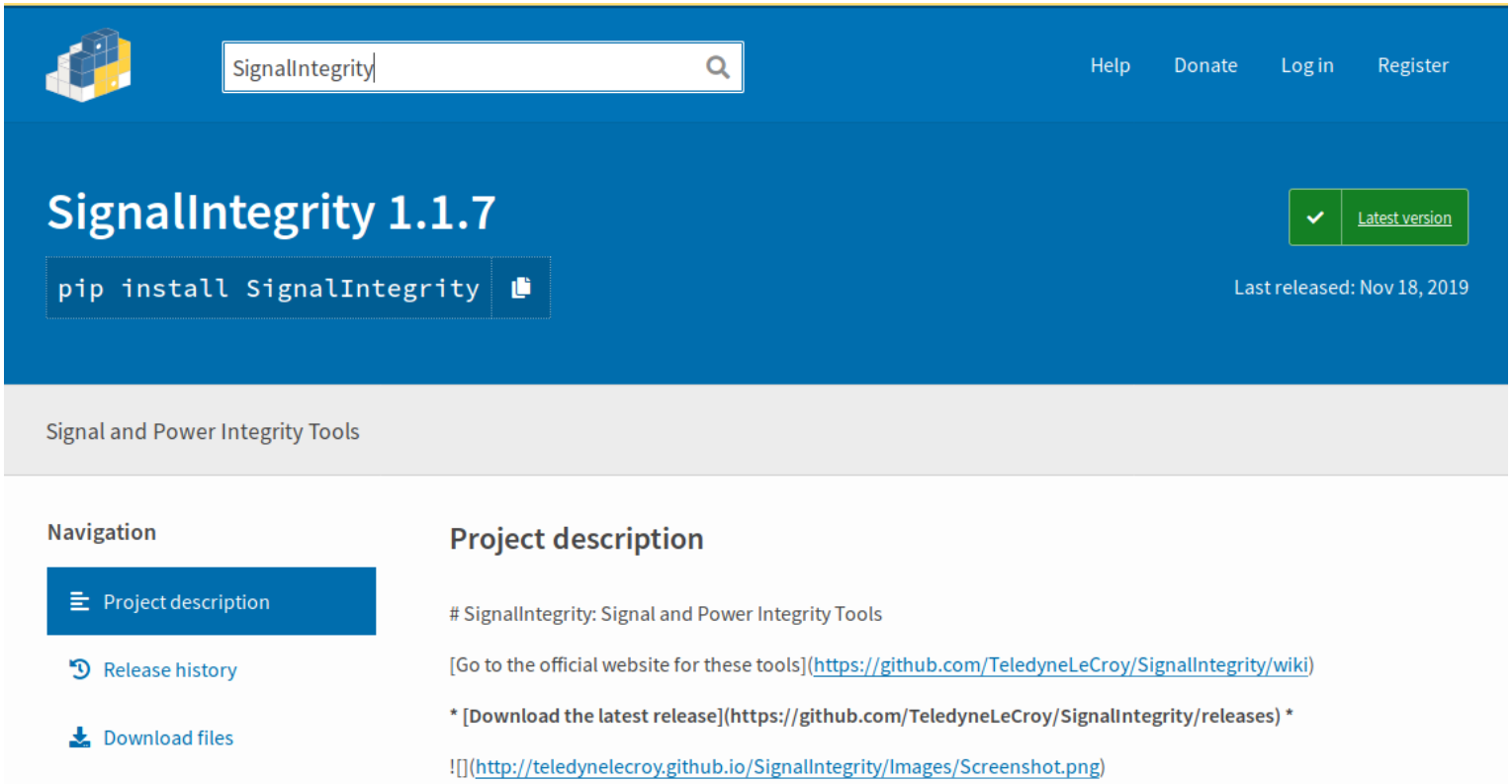
- Open-source software that provides for both scripted and GUI-based solutions to four major signal integrity problems:
  - S-parameters of systems;
  - De-embedding;
  - Virtual probing;
  - Linear simulation;
  - Network analyzer calibration.



Additionally, it can be used to view s-parameter files in both the time- and frequency-domain, view and fix physicality violations, resampling, and for transmission line model fitting.

It is officially released in the Python Packaging Index

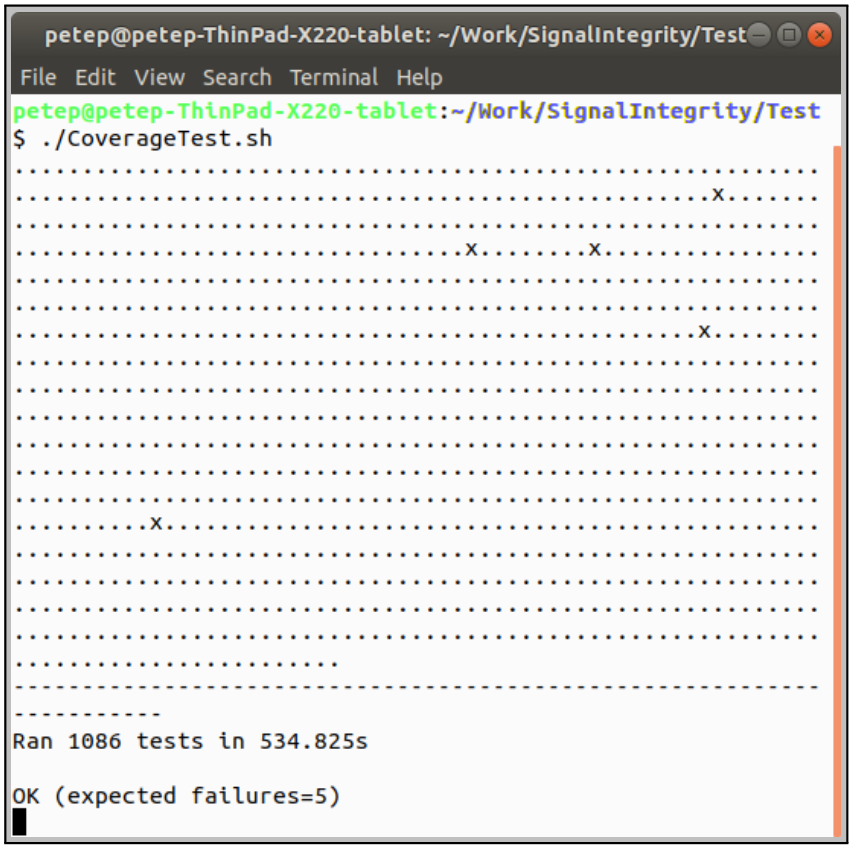
...and it runs under Python 2.x and 3.x on Linux or Windows



https://www.pypi.org/project/SignalIntegrity

Unittest

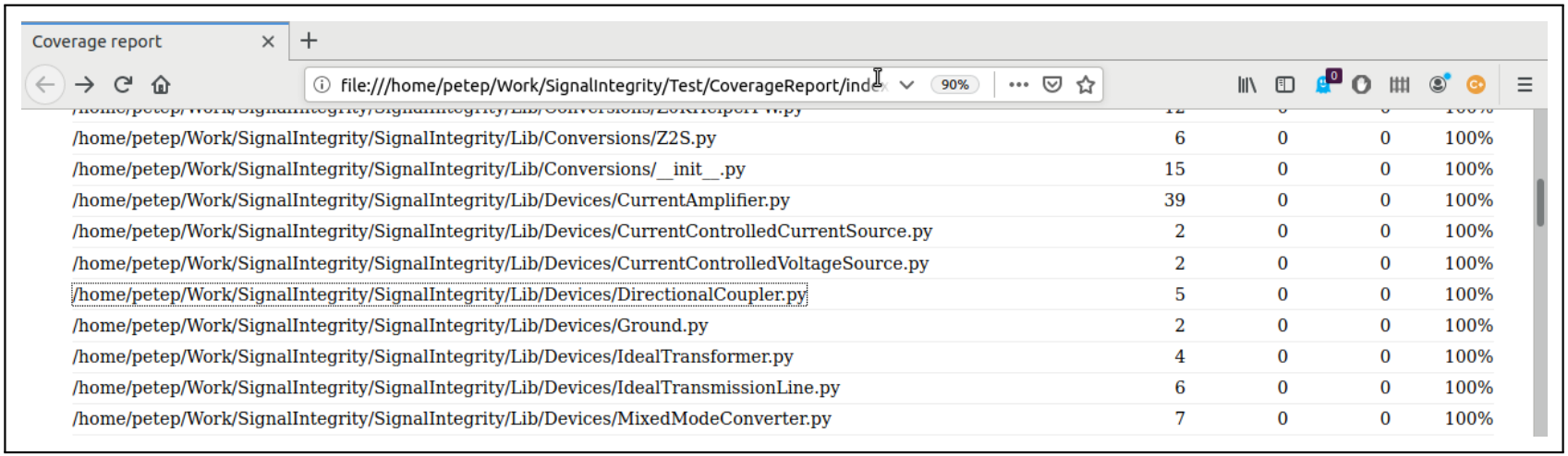
The standard test framework for Python



- *SignalIntegrity* contains over 1000 unit tests.
- The tests take about 10 minutes to run.
- The tests are a mixture of regression and absolute tests.
- They test the validity of the WavePulser 40iX algorithms!

Coverage

The standard Python test coverage framework



/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Conversions/Z2S.py	6	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Conversions/__init__.py	15	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/CurrentAmplifier.py	39	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/CurrentControlledCurrentSource.py	2	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/CurrentControlledVoltageSource.py	2	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/DirectionalCoupler.py	5	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/Ground.py	2	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/IdealTransformer.py	4	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/IdealTransmissionLine.py	6	0	0	100%
/home/petep/Work/SignalIntegrity/SignalIntegrity/Lib/Devices/MixedModeConverter.py	7	0	0	100%

- There are 16,340 lines of code in *SignalIntegrity*.
- There are 6,462 lines of code in the library (surprisingly small).
- There are 9,878 lines of code in the GUI application.
- 6,214 lines of code are covered (96% coverage).
- Lines not covered (no excuse) are exception handling.

SignalIntegrity is "free" software

Copyright (c) 2018 Teledyne LeCroy, Inc.

All rights reserved worldwide.

SignalIntegrity is free software: You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>

What does the GPL license mean?

- You have access to the source code.
- You can use the software for whatever you want.
- You can freely modify the software.
- There are no guarantees that it works.

However...

- The software is copyright Teledyne LeCroy, Inc.
- Any derivatives of the software that are distributed must adhere to the GPL license.
- Any distributed derivatives must maintain the copyright.

This kind of license is called a *copy-left* license, meaning it is constructed to maintain the free-ness.

Download Statistics

SignalIntegrity is downloaded:

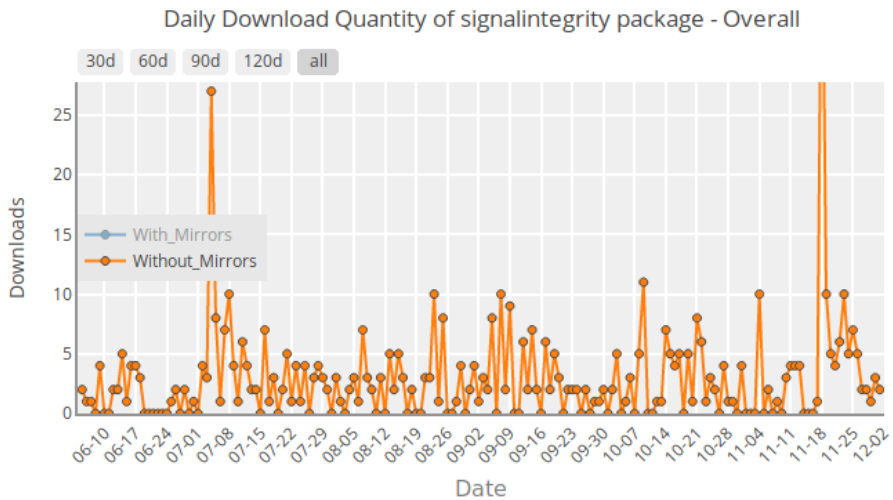
- 3-4 times per day.
- 90-120 times per month.

SignalIntegrity has been downloaded about 2000 times since it's launch.

signalintegrity

PyPI page
Home page
Author: Peter J. Pupalaikis
License: License :: OSI Approved :: GNU General Public License v3 or later (GPLv3+)
Summary: Signal and Power Integrity Tools
Latest version: 1.1.7
Requires: setuptools | pip | numpy | matplotlib | urllib3

Downloads last day: 2
Downloads last week: 22
Downloads last month: 136



SignalIntegrity Summary

- It is free software under the GPLv3 license.
- It can be used as a Python library for scripted solutions and has a GUI.
- It can solve problems numerically and symbolically.
- It is inextricably linked with the book, "S-Parameters for Signal Integrity".
  - The book describes how the software works.
  - The software is listed and provides several examples in the book.