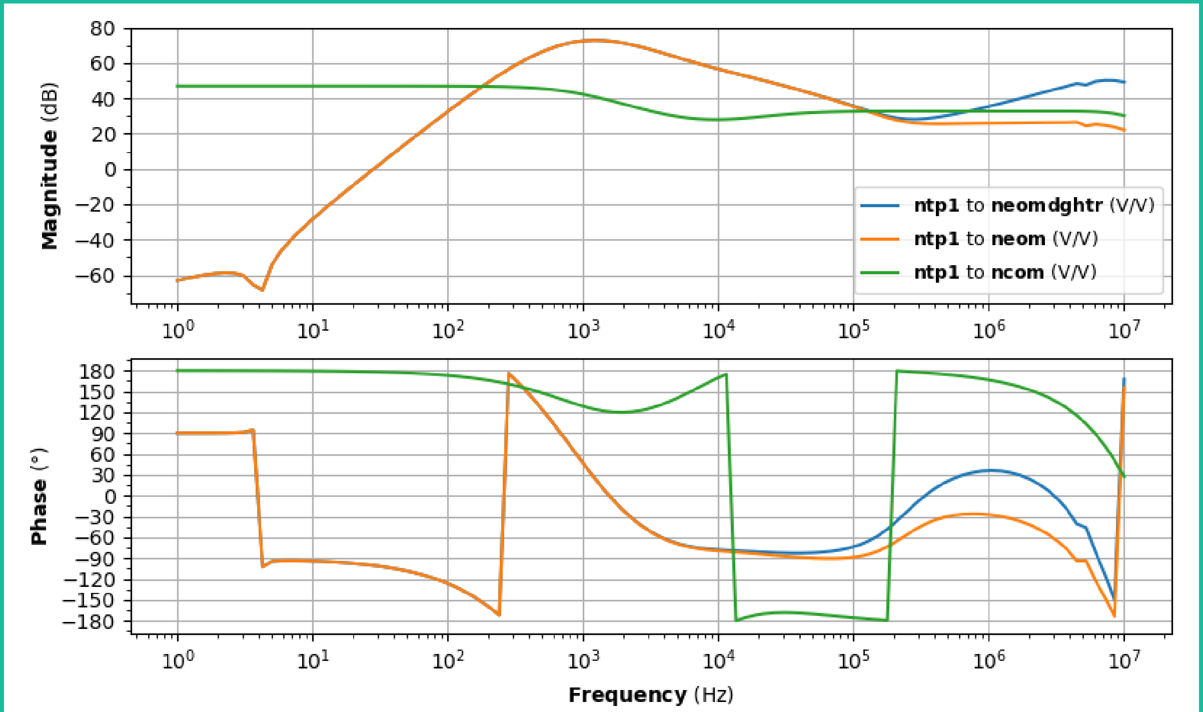


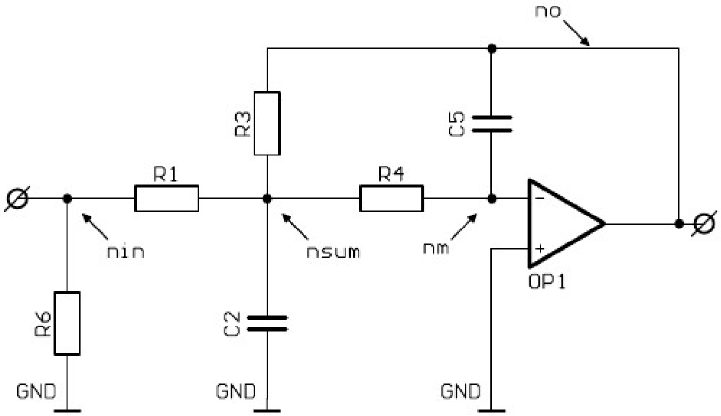
Zero: ~~yet another~~ a modern circuit simulation tool

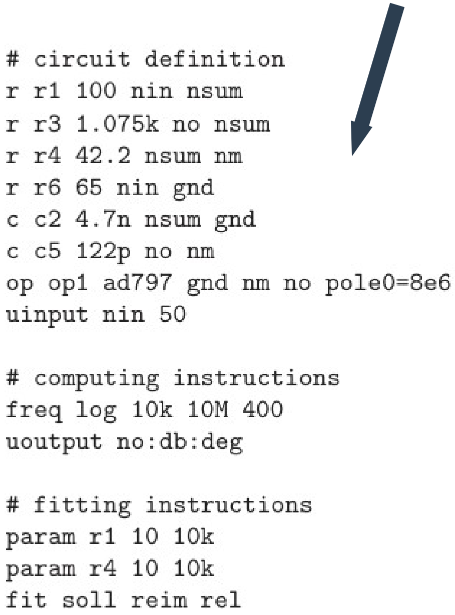
v0.6.3

Sean Leavey
AEI Hannover / Ifosim working group
GWADW, Elba, 2019-05-23
LIGO-G1901053

LISO

- “**Li**near **S**imulation and **O**ptimization of analog electronic circuits”
- **Developed by Gerhard Heinzel in the mid 90s**
- **Motivated by:**
  - Unrealistic op-amp behaviour found in SPICE
  - Need to determine stability criteria in circuits
  - Need for a fitting/optimisation tool
- **C based, closed source**
- **Performs *linear* analysis, a la Finesse, Optickle, etc.**
- **Circuit and analysis defined via input file**

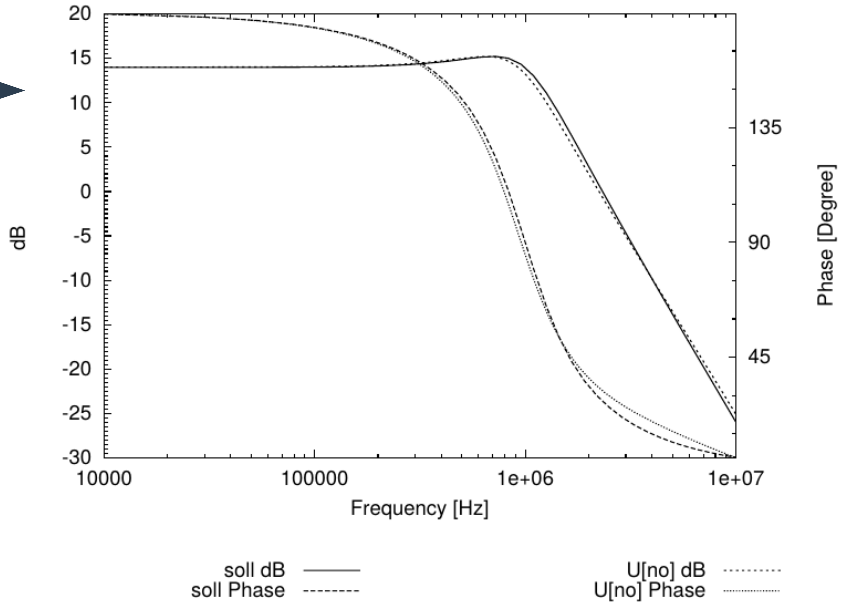


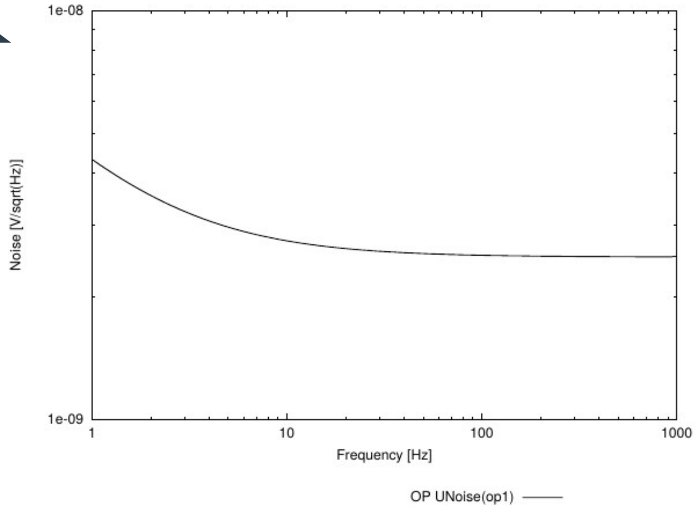


```
# circuit definition
r r1 100 nin nsum
r r3 1.075k no nsum
r r4 42.2 nsum nm
r r6 65 nin gnd
c c2 4.7n nsum gnd
c c5 122p no nm
op op1 ad797 gnd nm no pole0=8e6
uinput nin 50

# computing instructions
freq log 10k 10M 400
uoutput no:db:deg

# fitting instructions
param r1 10 10k
param r4 10 10k
fit soll reim rel
```

What people use LISO for

- **Response calculations**
  - Send signal in, read signal out at various nodes or components
- **Noise calculations**
  - Calculate noise at a node or component from other nodes/components
- **Optimisation and fitting**
  - Find appropriate component values that yield predetermined response functions
- **Stability analyses**



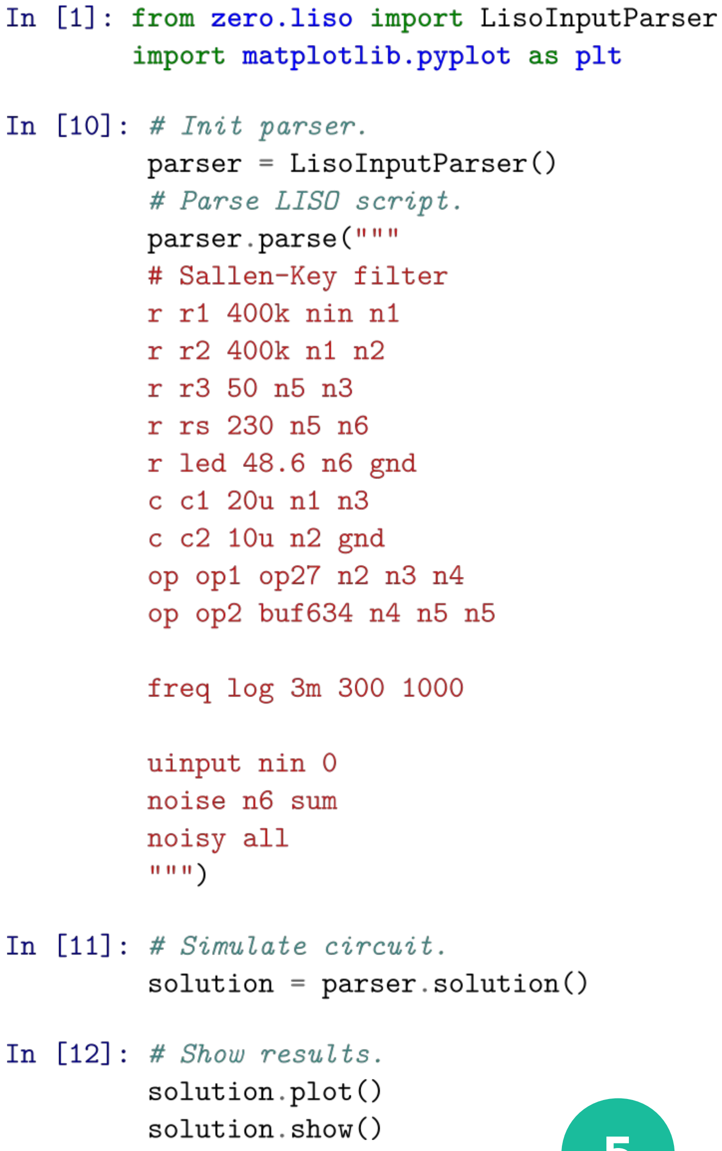


LISO limitations

- **LISO has worked pretty well for 20 years!**
- **But there are some limitations:**
  - Closed source; available only via email; officially Linux only
  - Everything goes through input and output files; harder to run parameter studies / batch jobs (same problems as Finesse ≤2)
  - Cannot fit noise, only responses
  - Everyone has their own version of the op-amp library

Zero

- **A clone of parts of LISO into a more accessible form to faciliate new features**
- **Python-based, open source**
  - Cross-platform
  - Can use in Jupyter notebooks
  - Access to large scientific Python ecosystem (NumPy, SciPy, control systems toolbox, Finesse 3, etc.)
  - Modular; can bolt-on new analyses or use different solvers in future
- **Supports most LISO input syntax (circuit mode) but also native Python**
- **Includes expanded LISO op-amp library (YAML formatted)**

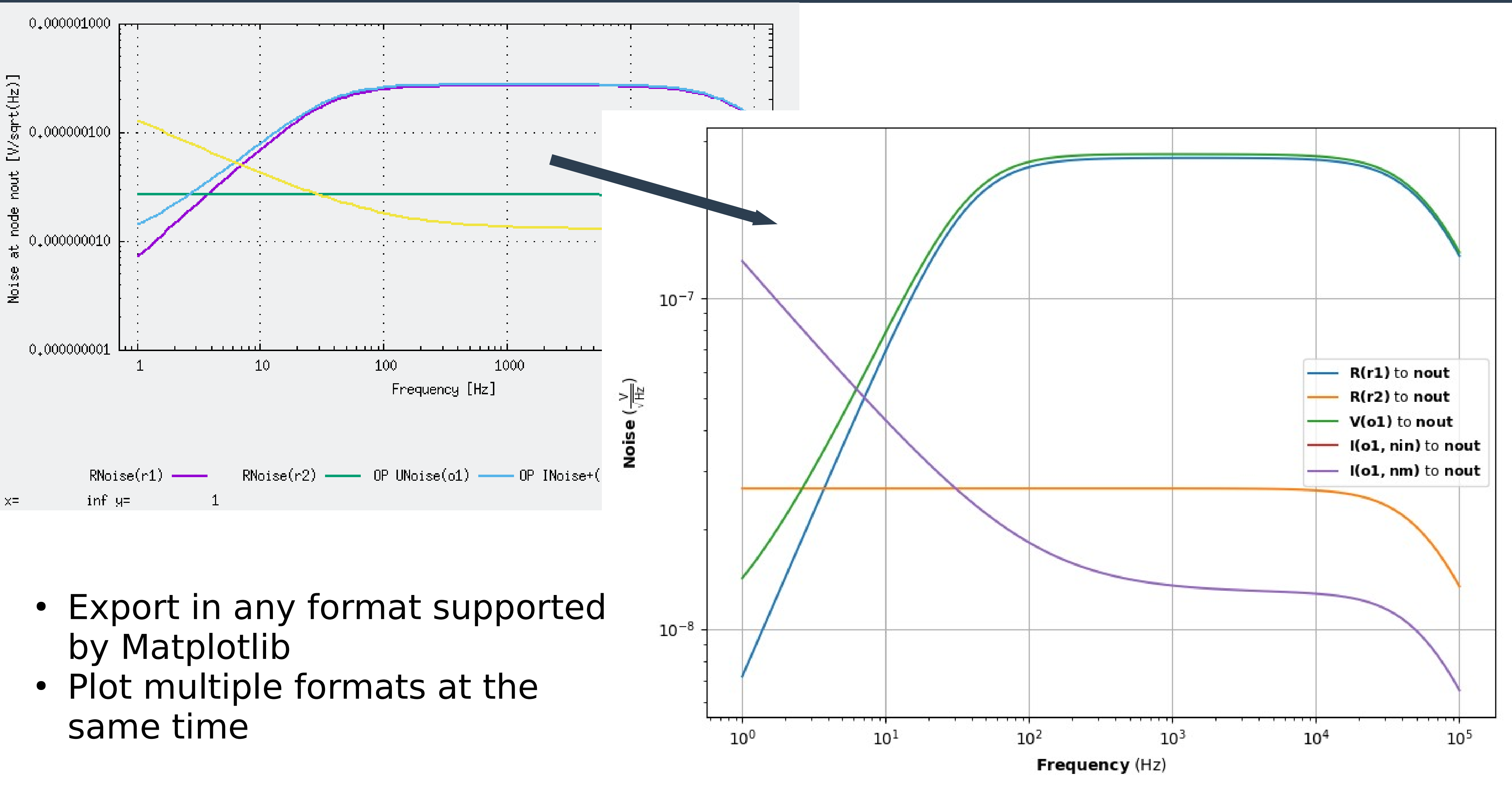


```
In [1]: from zero.liso import LisoInputParser
        import matplotlib.pyplot as plt

In [10]: # Init parser.
         parser = LisoInputParser()
         # Parse LISO script.
         parser.parse("""
         # Sallen-Key filter
         r r1 400k nin n1
         r r2 400k n1 n2
         r r3 50 n5 n3
         r rs 230 n5 n6
         r led 48.6 n6 gnd
         c c1 20u n1 n3
         c c2 10u n2 gnd
         op op1 op27 n2 n3 n4
         op op2 buf634 n4 n5 n5

         freq log 3m 300 1000

         uinput nin 0
         noise n6 sum
         noisy all
         """)

In [11]: # Simulate circuit.
         solution = parser.solution()

In [12]: # Show results.
         solution.plot()
         solution.show()
```

Command line interface

- **Show and save simulation results/plots**
  - `zero liso my-circuit.fil –-save-figure circ.pdf`
- **Search op-amp library with binary operators**
  - `zero library search “vnoise < 3n & inoise < 1p”`
- **Automatically download op-amp datasheets using search query**
  - `zero datasheet -f “OP27”`
- **Print circuit matrix / equations**
  - `zero liso my-circuit.fil --print-matrix --print-equations`
- **Open library and config in text editor**
  - `zero config --help`

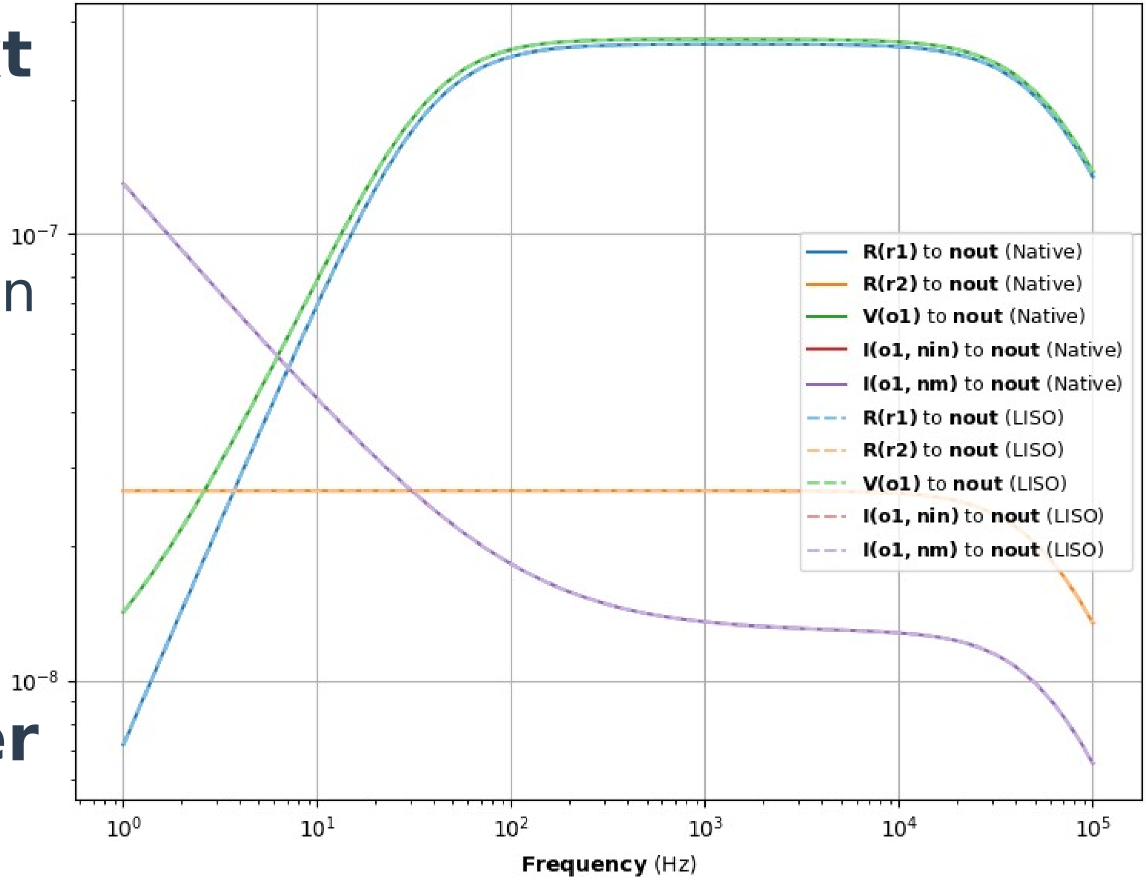
Prettier plotting



- Export in any format supported by Matplotlib
- Plot multiple formats at the same time

Simulation results

- **Results as queryable objects and not just text files**
    - Functions know their units, can check if units clash when performing mathematical operations*
- **Easily combine results from multiple simulations in single plots, then plot together**
    - solution_c = solution_a + solution_b



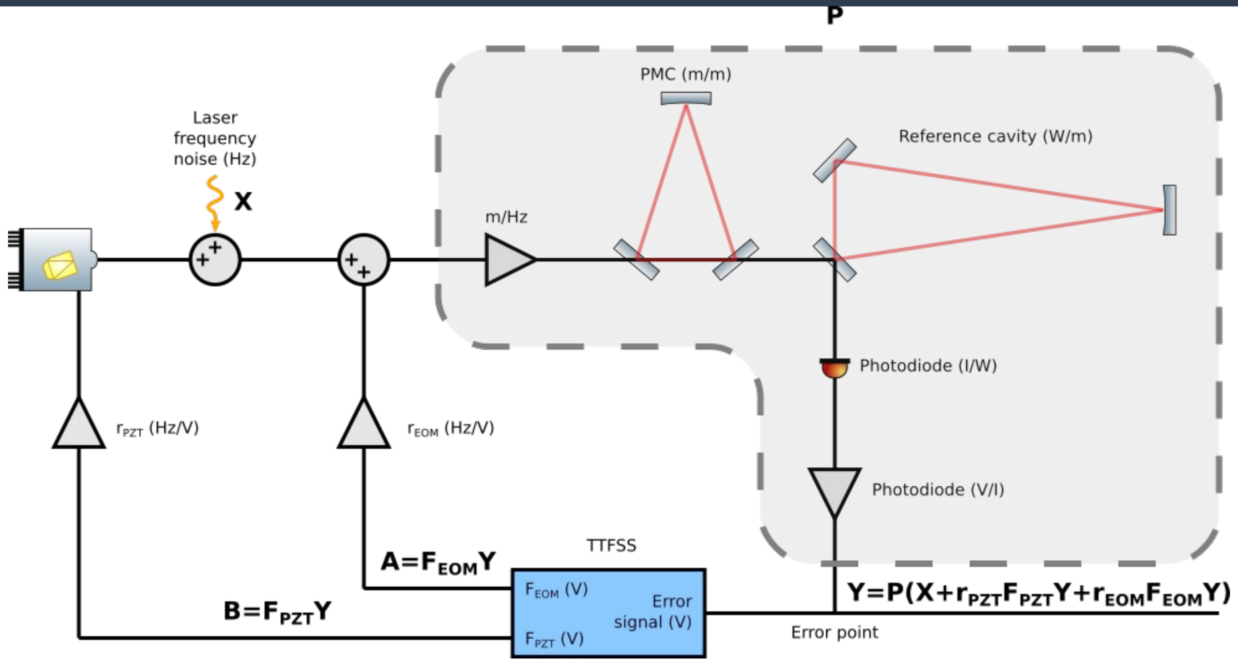
*Currently separate project (https://git.ligo.org/sean-leavey/freqstab/), will be merged soon

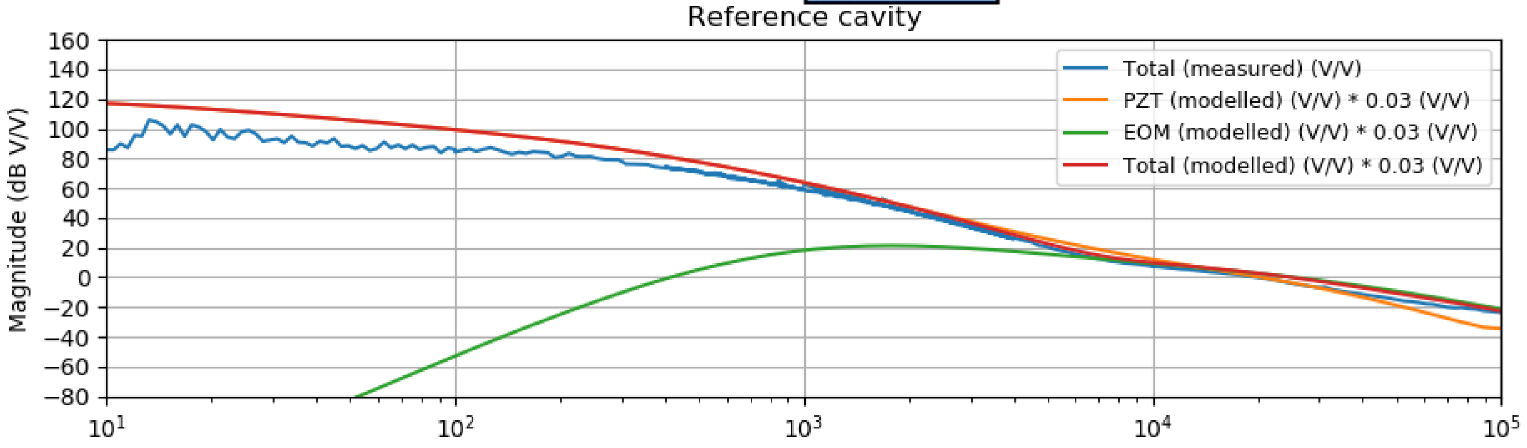
Comparing to LISO

- **LISO files in Zero can be compared to LISO automatically**
  - `zero liso my-circuit.fil --compare`
  - Runs LISO directly and overlays results to Zero
- **Automatic tests against hundreds of LISO files identical within 10^-5 relative/absolute tolerance**

Zero as part of wider analyses

- **AEI 10 m reference cavity open loop gain model**
  - Zero
  - Finesse
  - Python control library





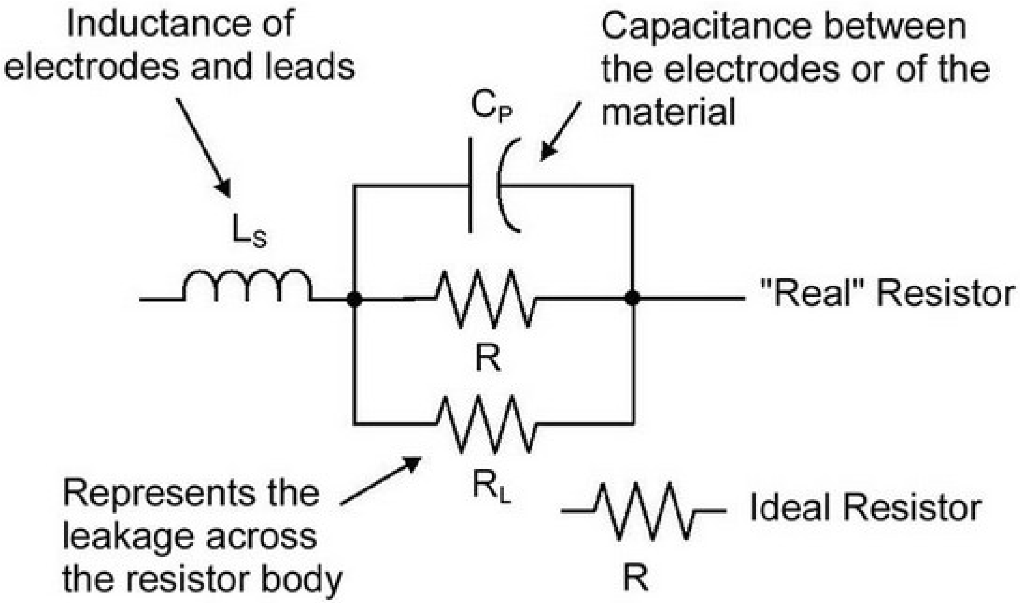
https://git.ligo.org/sean-leavey/freqstab/

Limitations of Zero

- **No direct fitting and optimisation functionality**
  - But Zero can be included as a library within other fitting and optimisation tools
- **Stability, max input, etc. analyses not yet available (but can be added if there is a desire)**
- **Slower (currently)**
  - But could be improved with C-based sparse matrix solver a la Finesse 3

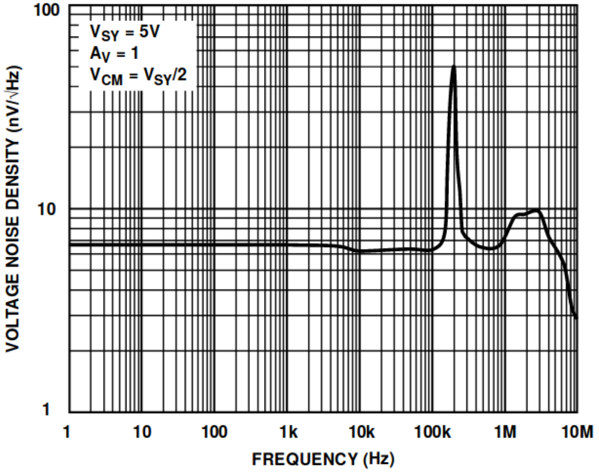
Future

- **Continue to expand op-amp library**
- **Support realistic passive components (resistors with stray C, L, excess noise, etc.)**
- **Support arbitrary op-amp noise, not just 1/f**
- **Future direction depends on feedback from you!**



Inductance of electrodes and leads
Capacitance between the electrodes or of the material
C_P
L_S
"Real" Resistor
R
R_L
Represents the leakage across the resistor body
Ideal Resistor
R

https://www.nutsvolts.com/magazine/article/resistors_types_and_applications



ADA4528

Figure 3. Voltage Noise Density vs. Frequency

Where to get it

- **On your computer**
  - `pip install zero`
- **On GitLab**
  - https://git.ligo.org/sean-leavey/zero
- **Help and support**
  - Examples on GitLab
  - Documentation at https://docs.ligo.org/sean-leavey/zero/
  - GitLab issue tracker for help/requests/complaints
  - Circuit simulation Mattermost channel: https://chat.ligo.org/ligo/channels/circuit-simulation