Getting the Most from LTspice (and a Quick Look at LTpowerCAD)

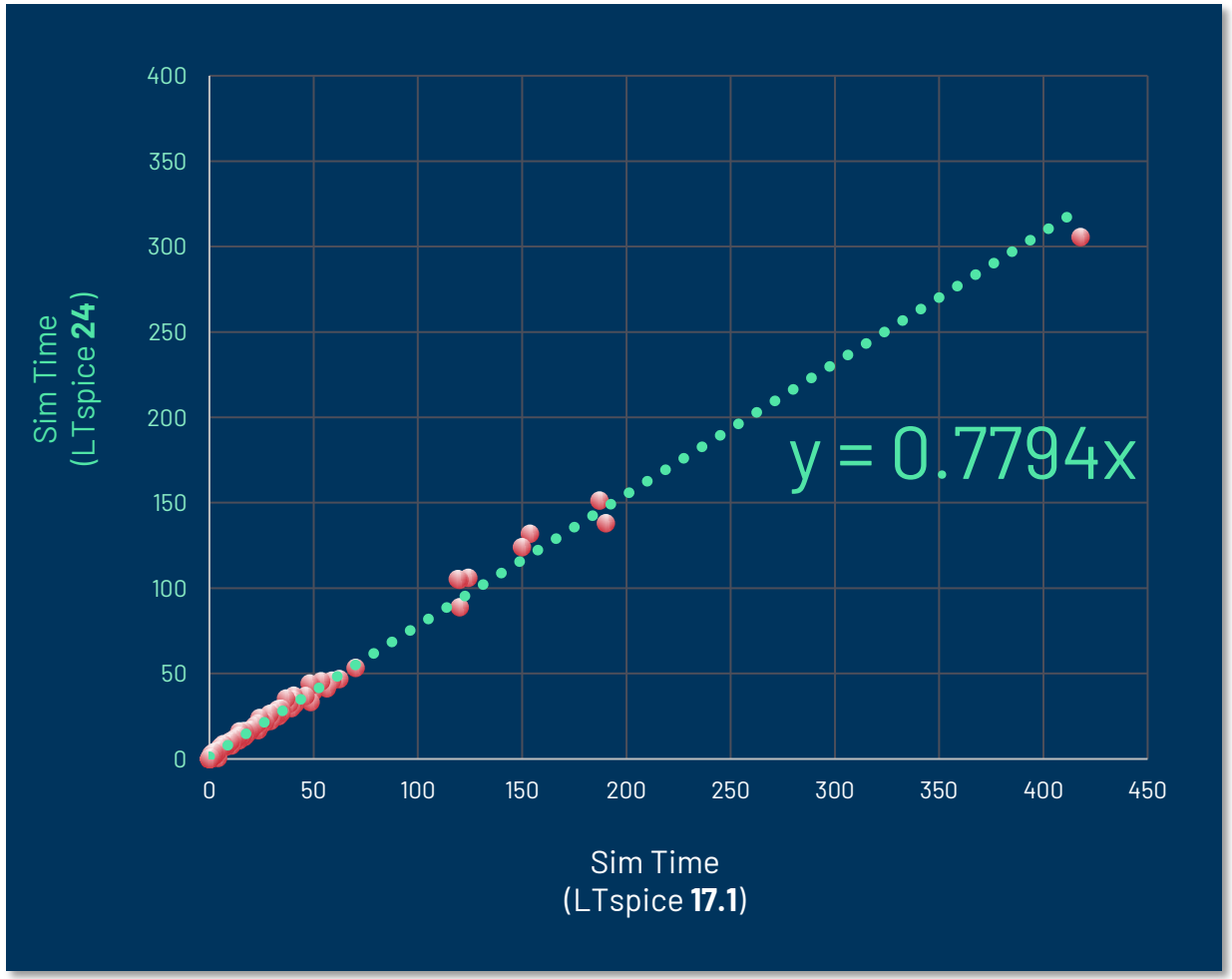
Bill Geosits
Principal Engineer, Field Applications
Analog Devices
November 6, 2025

Agenda

- What's new in LTspice24
- LTspice Basics Refresher
- Using new features in LTspice
  - String parameters
  - savestate & loadstate
  - Hierarchical Symbol Libraries
- Common challenges and how to solve them
  - Convergence and simulation speed
- Frequency Response Analysis (FRA)
- Quick Look at LTpowerCAD

What’s new in LTspice24

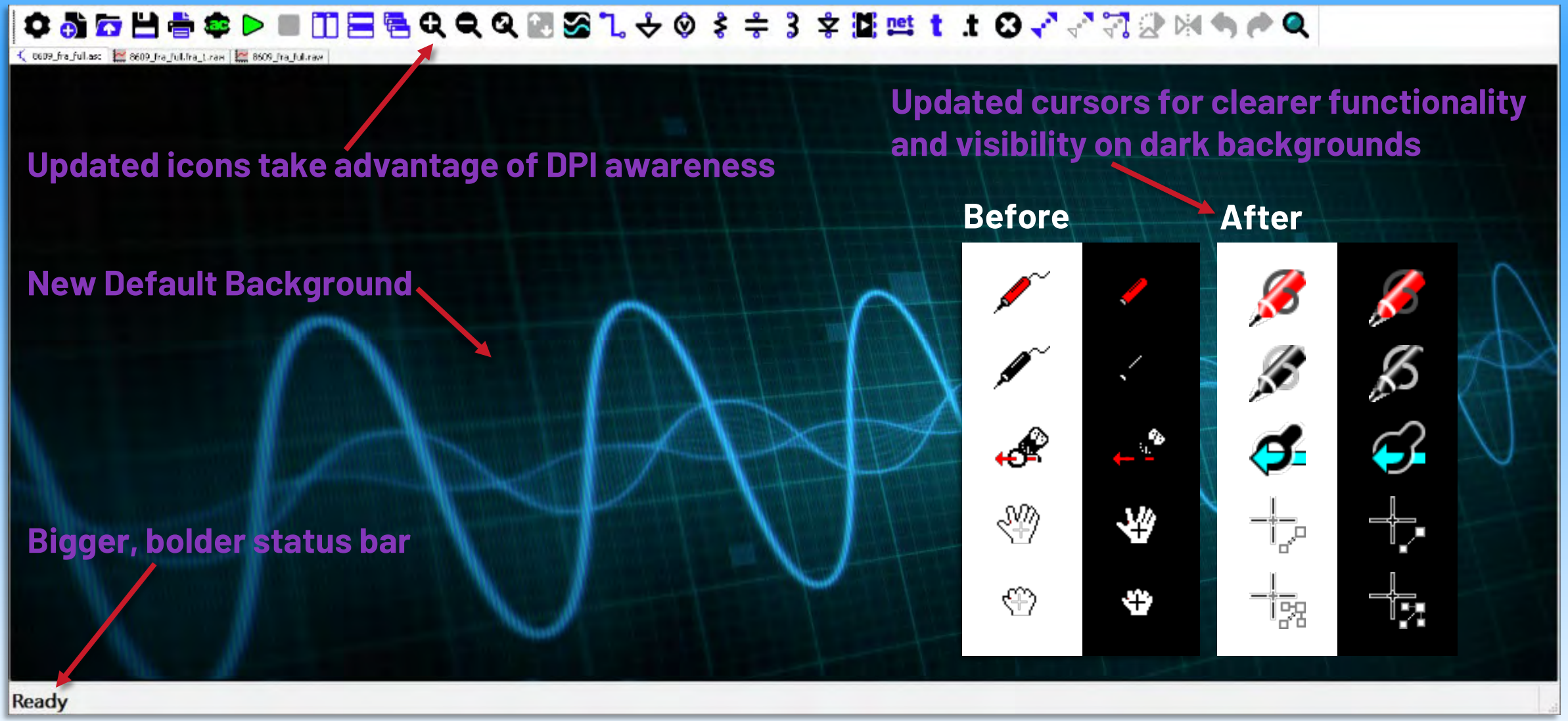
LTspice 24: Faster Simulations



- Improved simulation speed
  - Benchmarked ~200 popular MMP examples
  - ADI-standard Dell i7 Precision 5550 laptop
- Improved run-to-run consistency
- Changed default trtol to 2 for further improved performance

LTspice 24 UI: Refresh Overall Look and Feel

DPI awareness improves UI scaling on high-res monitors, avoiding too-smalls and jaggies

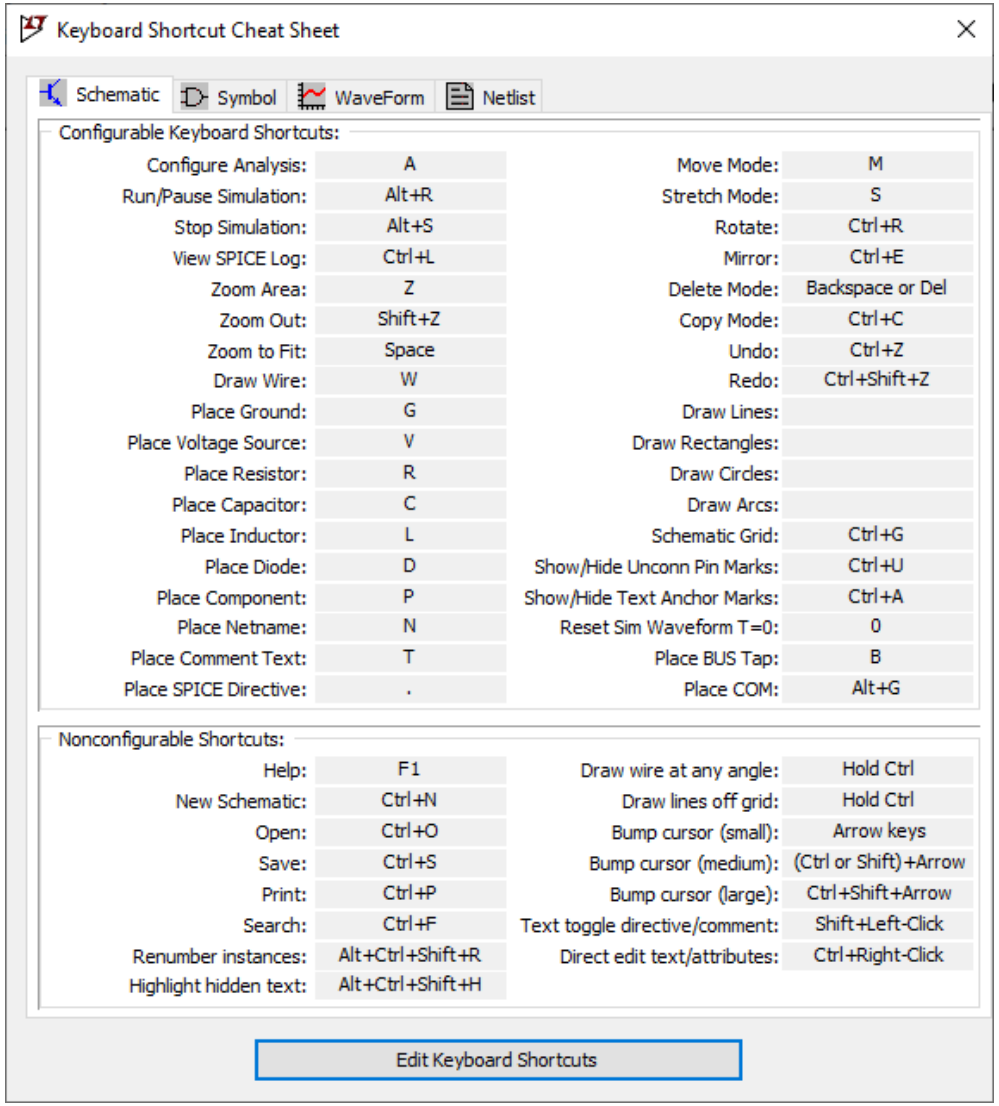


New Keyboard Shortcuts and Dynamic Cheat Sheet

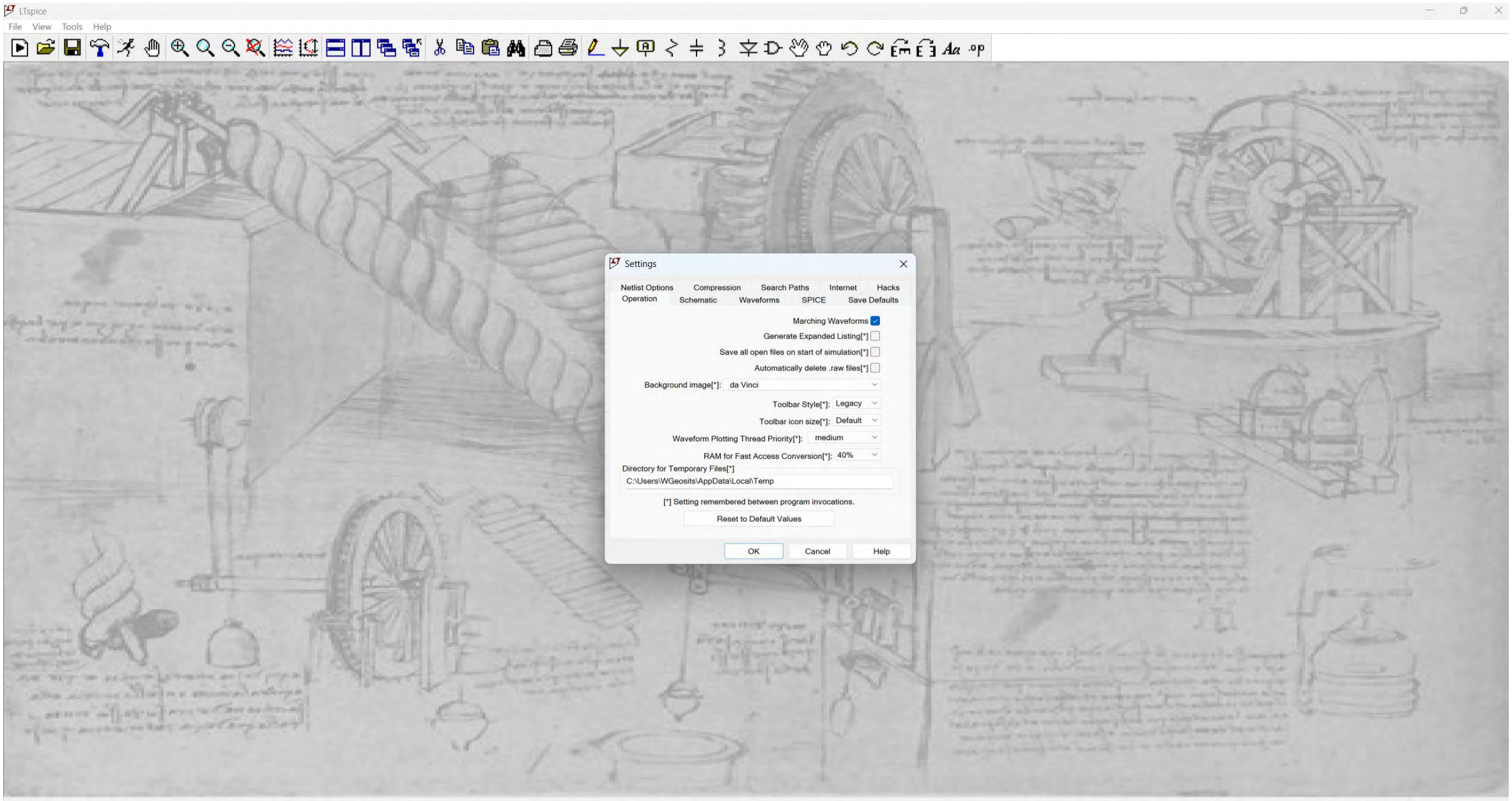
- Customization-safe
- Return to old shortcuts via

Restore LTspice Classic Values

New Non-Modal, Floating Cheat Sheet Available from Help Menu

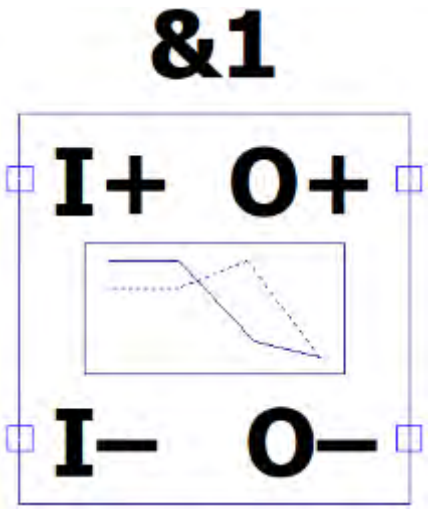


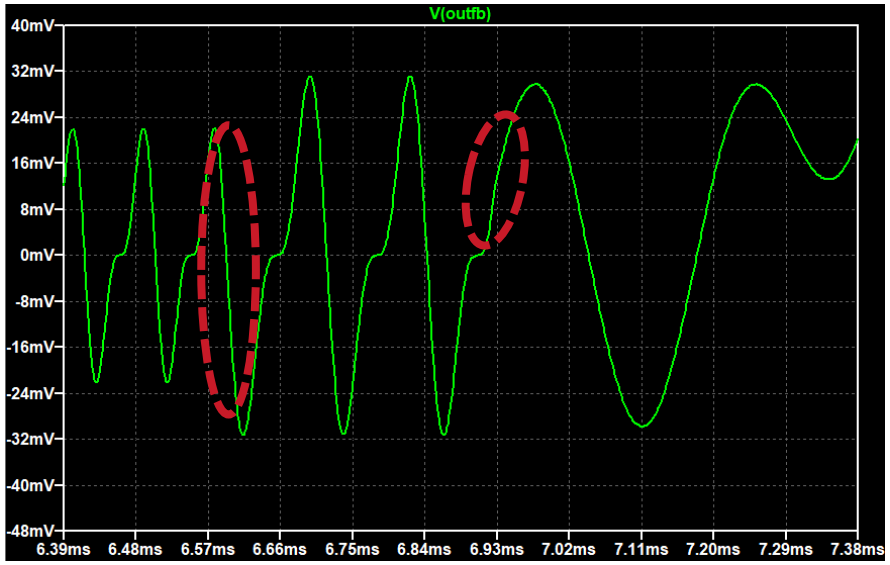
Yes, you can go back to the old toolbar!!



FRA Upgrades

- 4-terminal Frequency Response Analyzer Probe
  - Enables Bode plots of any part the loop
  - Simplifies analysis of µModules with integrated top feedback resistors; negative outputs; and current feedback
- Phase changed to represent phase margin (phase +180°)
- Smooth stimuli transitions between frequencies
  - Faster settling / improved accuracy



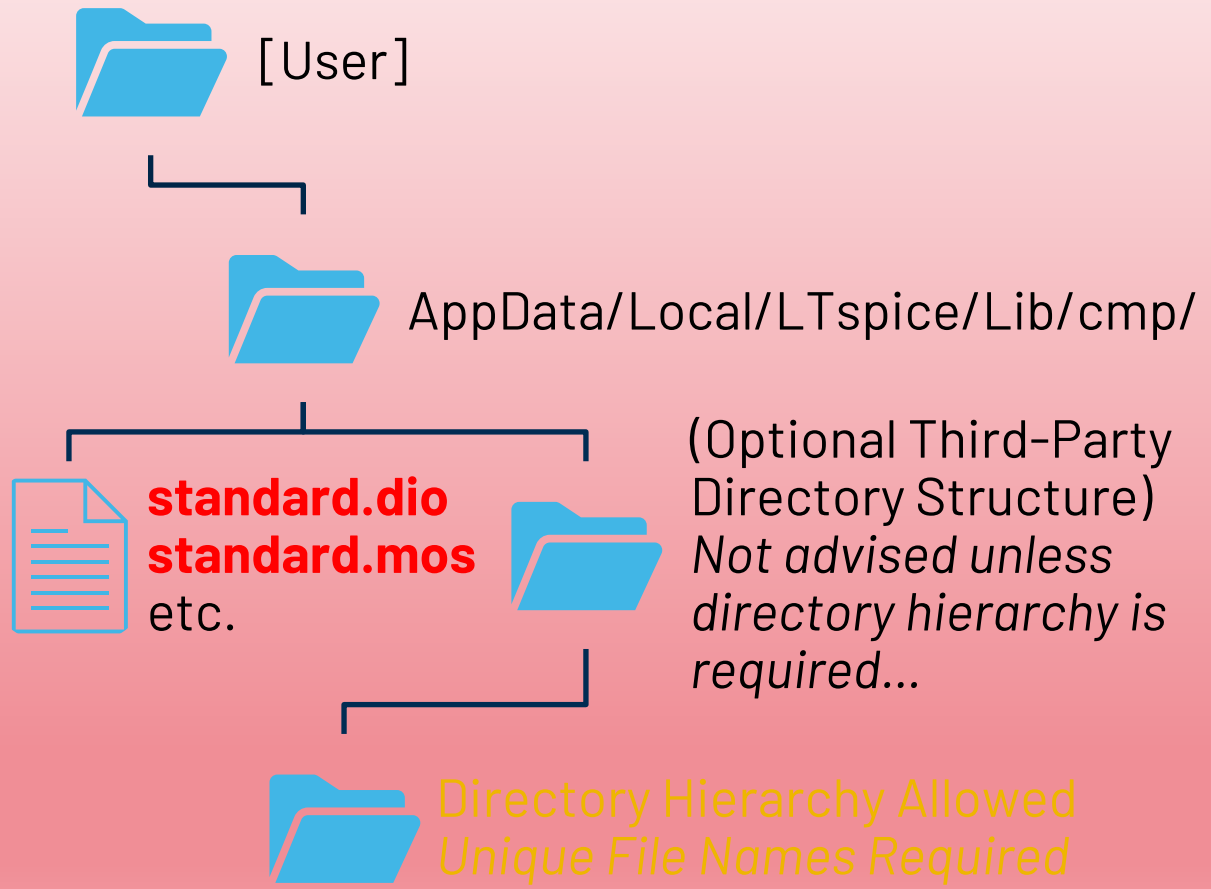


Component Libraries and AppData

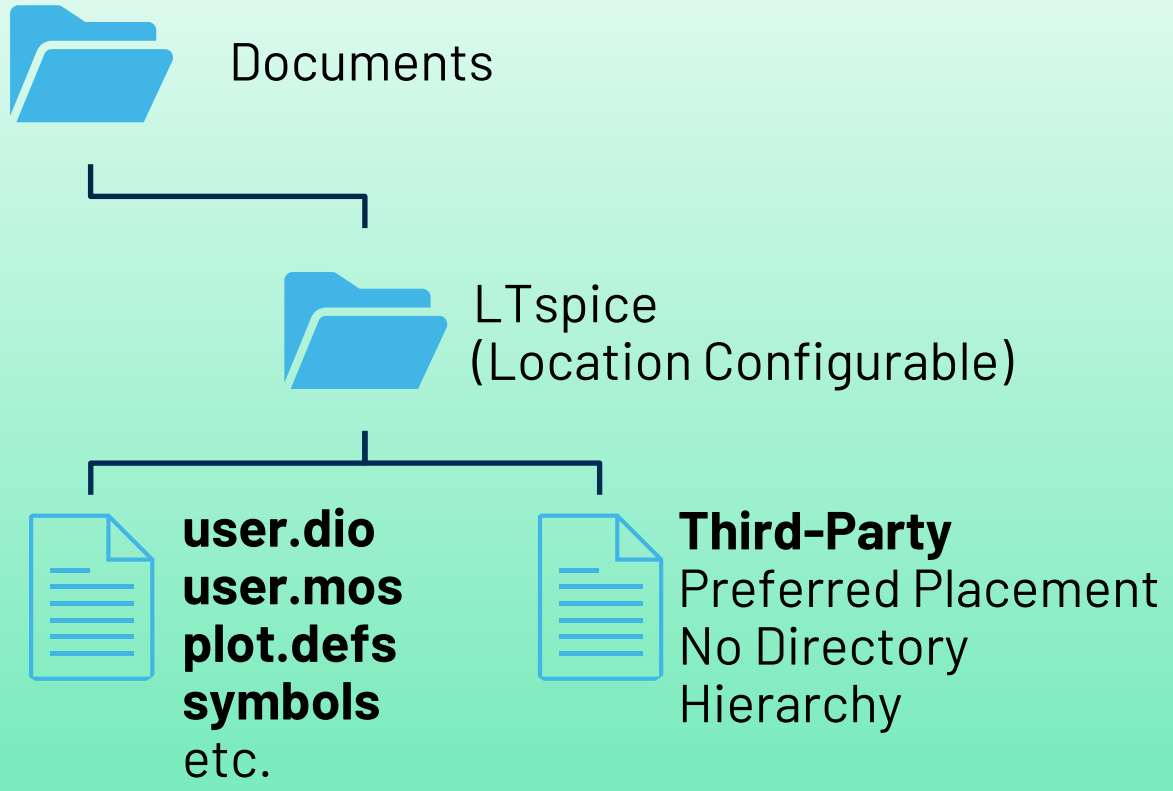
- No more “standard” library merge conflicts
  - Standard discrete component libraries are simply overwritten when updated…
- **Do not** edit ADI-installed files in **%localappdata%\LTspice**
  - Prompts discourage users from saving in AppData
  - To add third-party symbols, libraries, and sub-circuits, place *files* in **Documents\LTspice** (directory configurable in Settings)
    - Files only; this is not recursive — does not support directory hierarchy.
    - Hierarchical directories and files with unique names may be added to %localappdata%\LTspice
- Optional user-defined libraries (user.dio, mos, cap, ind, bjt, etc.)
  - May be added to define custom discrete devices in selection dialogs and simulations
  - Place in **Documents\LTspice** by default (configurable in Settings)

Component Libraries and AppData (Cont.)

LTspice Installs/Overwrites Files Here

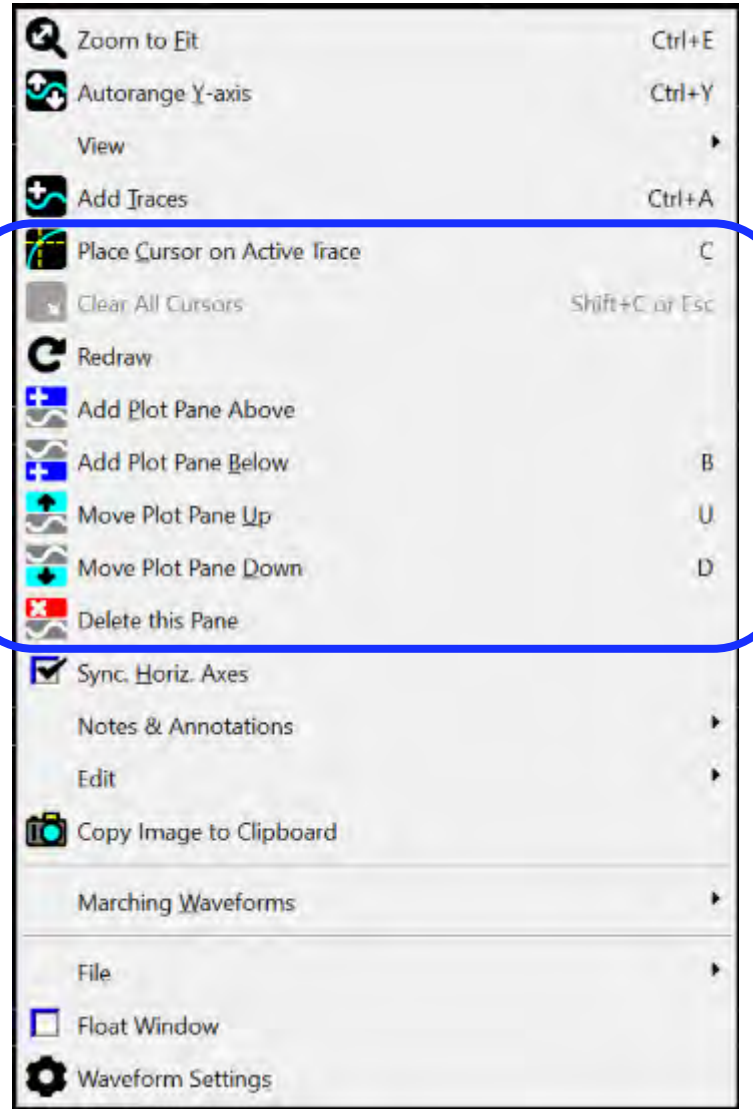


User Files Should be Placed Here



Waveform Viewer: Functionality Improvements

- Move plot panes up & down
- Add plot panes above or below
- Menu items and shortcuts to add/clear cursors, ESC key clears cursors
- Arrow keys to pan up, down, left and right
- Enable drag traces between panes during simulation
- Enable zooming while plotting

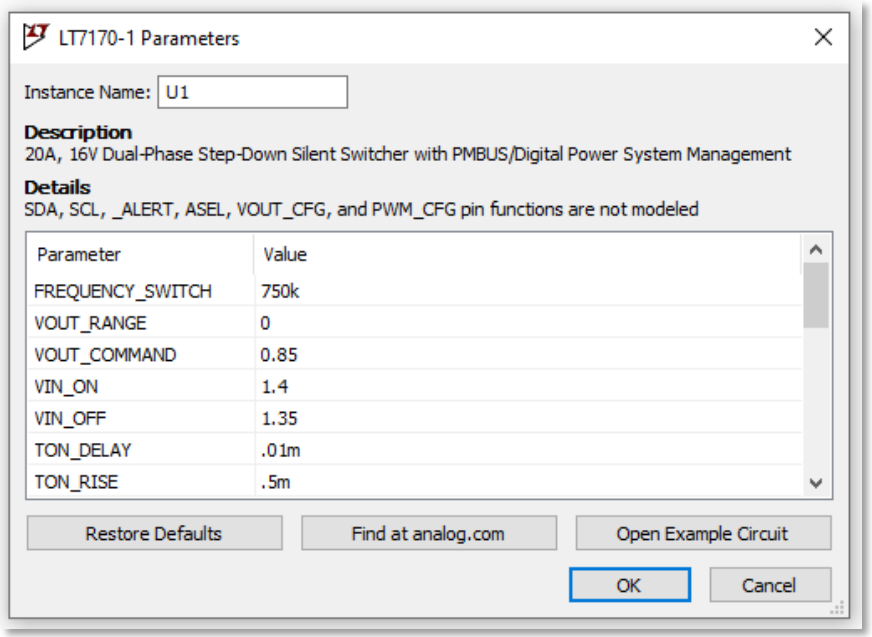


Simulation Control

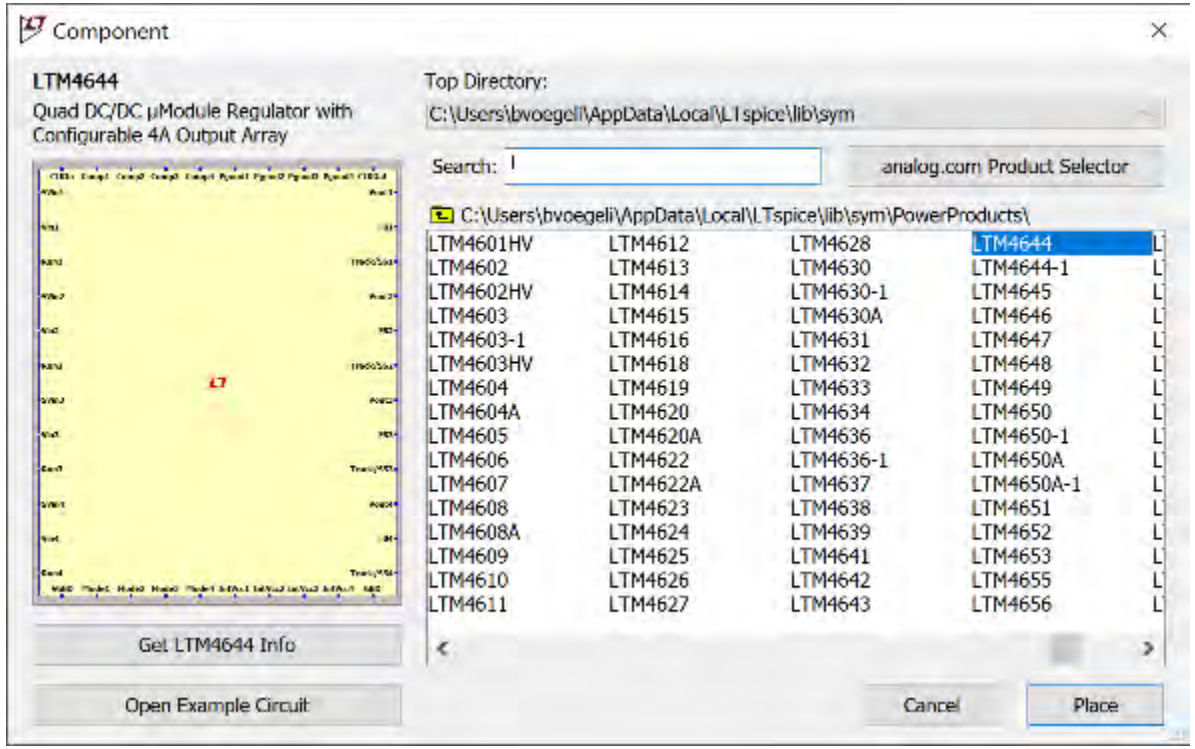
- New **Configure Analysis** Toolbar  ton and Shortcut ("A")
- Improved Configure Analysis Dialog Functionality
  - Captures all simulation commands on the schematic, *including comments*
  - Populates tabs accordingly
  - Automatically comment/uncomment schematic text
- **Shift + Left-Click** toggles text between **directive** and **comment**

Improved Dialogs

- New: Symbol Parameter Editor



- Better Component Selection
  - Cleaner, more efficient layout
  - Improved search in dialog
  - Quick links to analog.com



LTspice Basics

Simulation is controlled by text on the schematic

Black text = SPICE Directive = Active Simulation Control

Right-click directives and components to bring up helper dialog

Press ▶ to run simulation

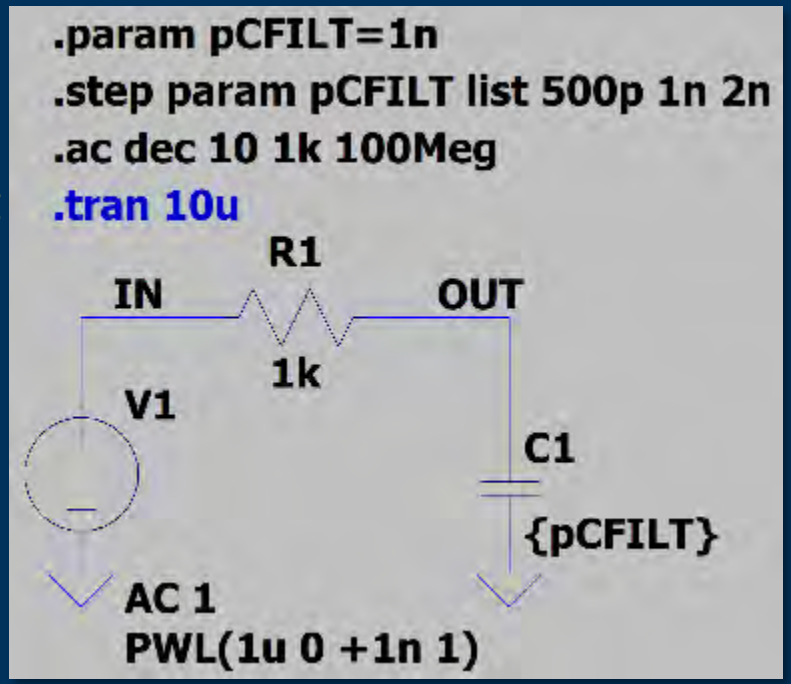
Click on node to plot

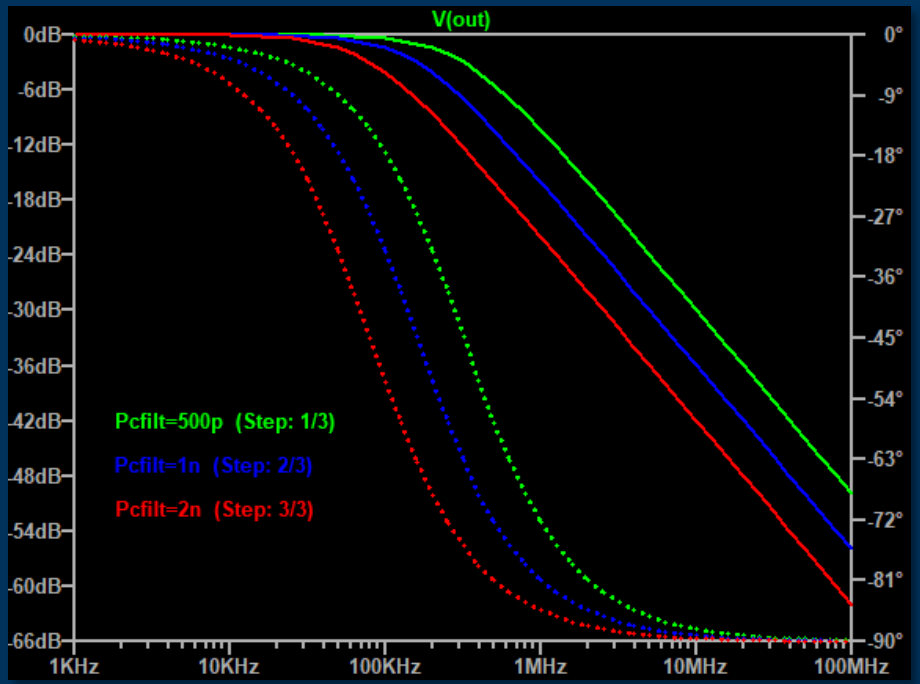
Right-click > Notes & Annot...> Annotate Steps

Blue text = Inactive Comment

Shift + Left-Click to toggle

directive • **comment**



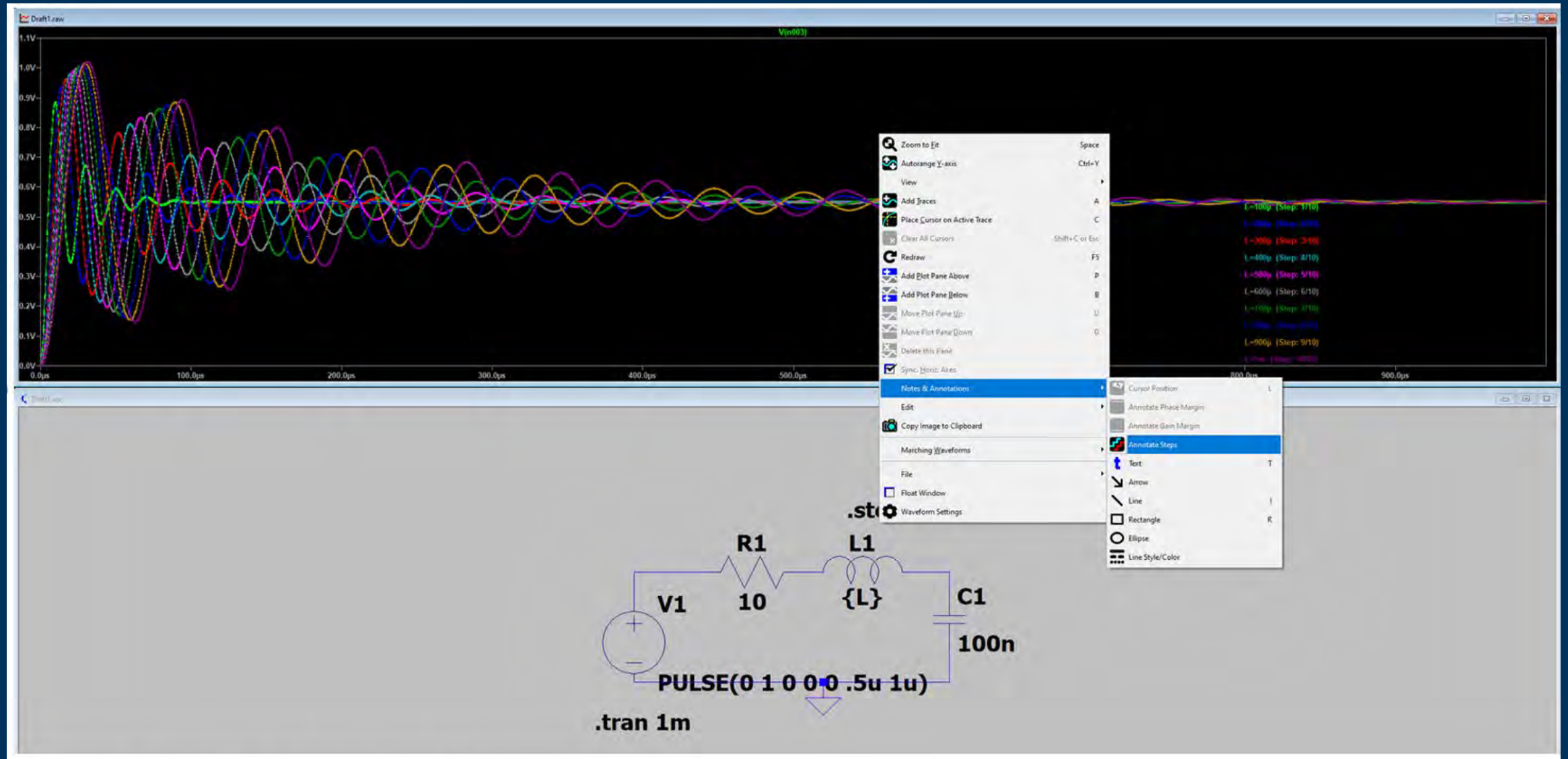


For .ac: 1V AC Voltage source

For .tran: PWL step 0V→1V at time 1µs, rise time 1ns

{parameterized capacitor value}

LTspice Basics – Identifying Traces in a Parameter Sweep



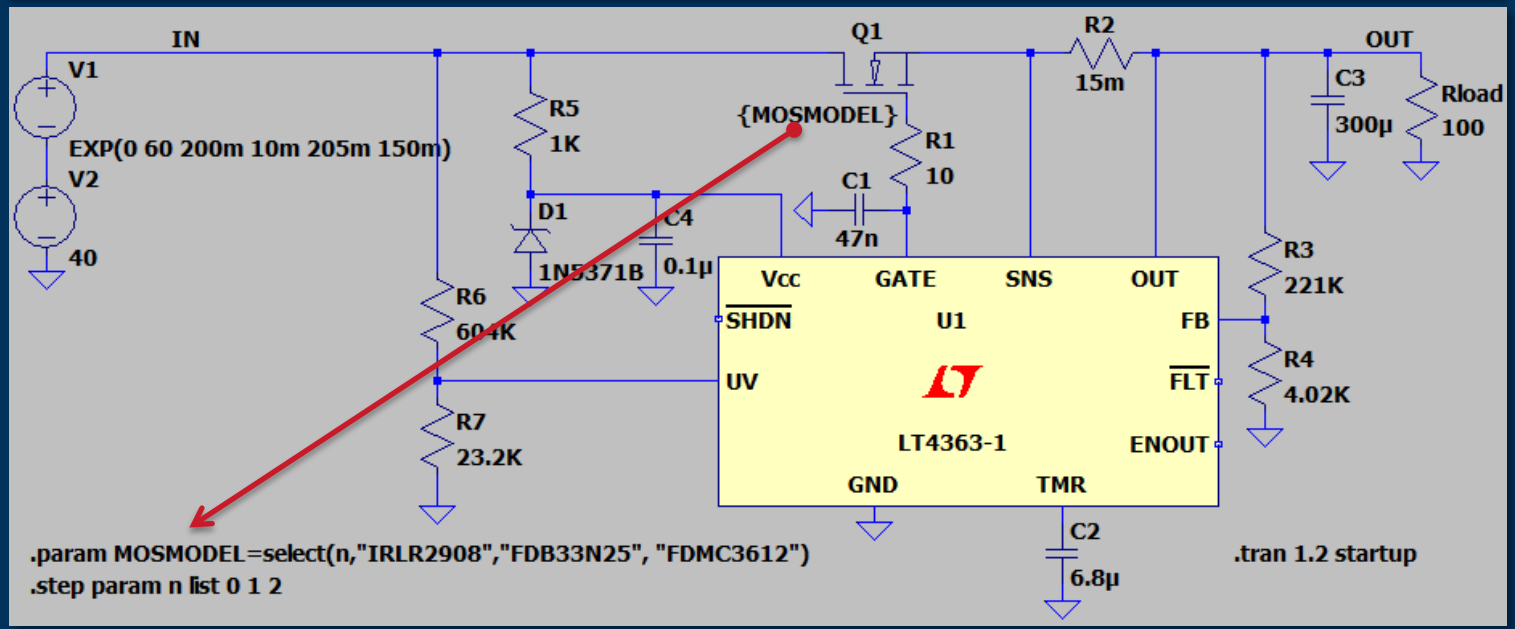
You can use view->select steps to pick the traces you want plotted.

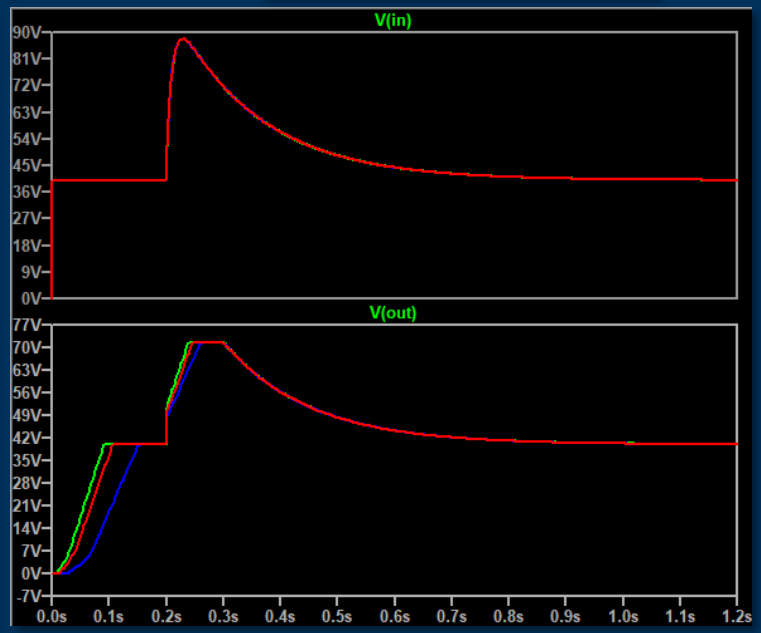
String Parameters (LTspice 24.1+)

Example: Stepping *intrinsic* MOSFET .models

```
.param MOSMODEL=select(n,"IRLR2908","FDB33N25","FDMC3612")
.step param n list 0 1 2
```

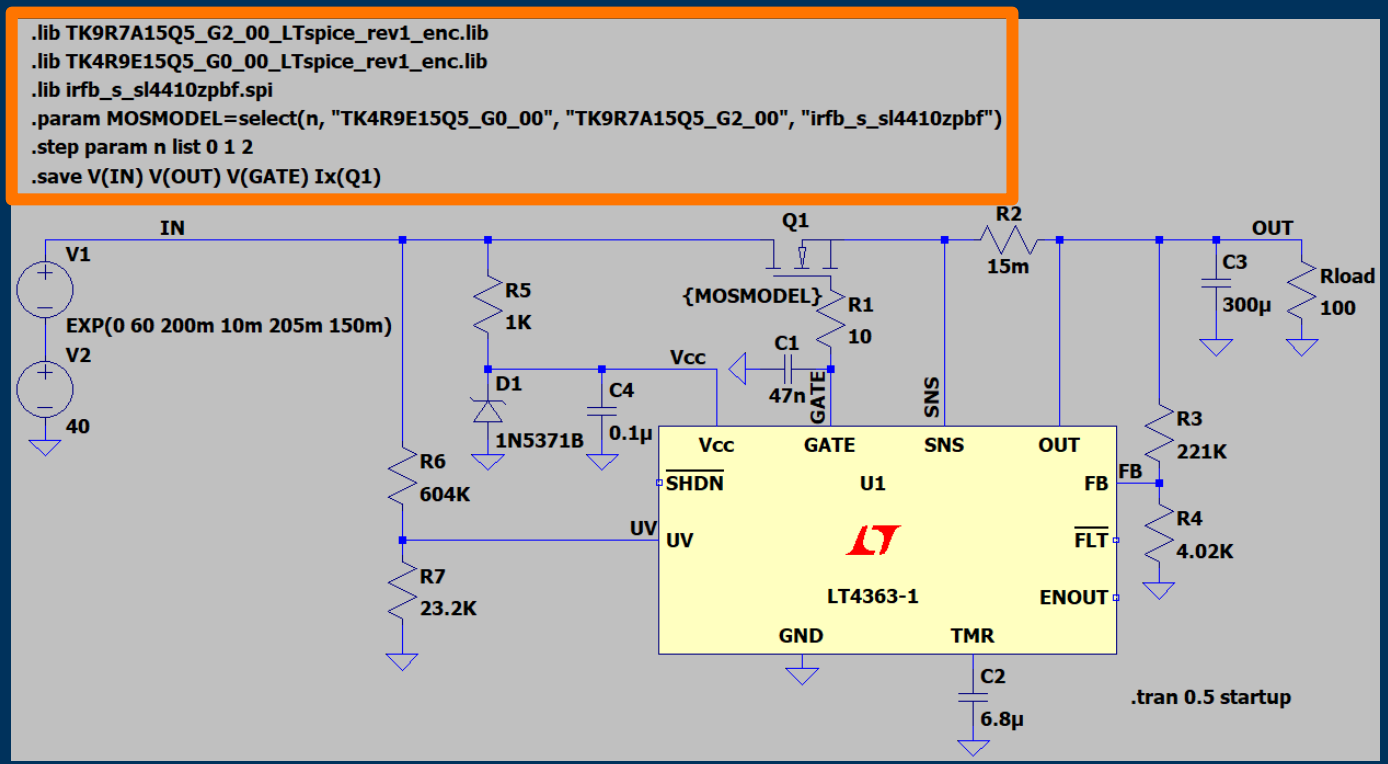
n	.model
0	IRLR2908
1	FDB33N25
2	FDMC3612

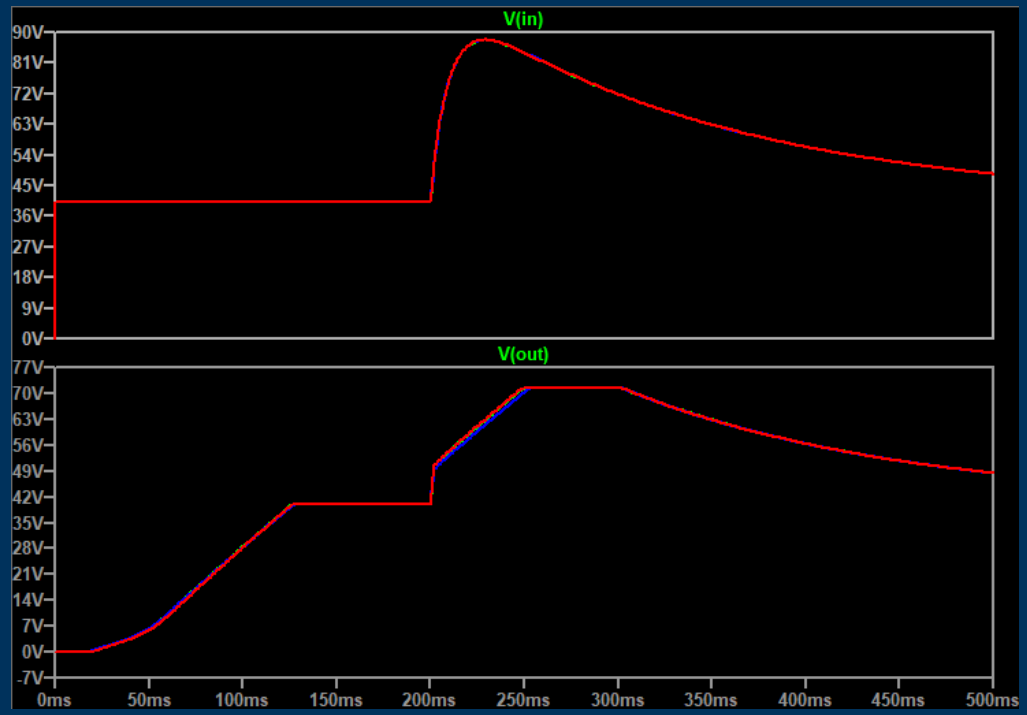




String Parameters (LTspice 24.1+)

Example: Stepping .subckt models





Important:

- The list of **signals saved in the raw file must match among all steps**
  - If subckt models have different structures, **use .save** to limit saved signals
- Ctrl+Right-click on Q1 nmos symbol to **set prefix to "X" for subckt**

Importing Third Party Spice Models

https://www.analog.com/en/resources/technical-articles/ltspice-how-to-import-third-party-models.html

ANALOG DEVICES

TECHNICAL ARTICLE

LTspice How to: Importing Third-Party Models

Anne Mahaffey, Principal Applications Engineer, and
Michael Potts, Staff Field Applications Engineer

Abstract

This article presents how to import third-party SPICE models into LTspice, step by step. The process of importing two different model types is covered: models implemented with **.MODEL** directives and models implemented with **.SUBCKT** blocks. The steps provided are intended to ensure maximum portability when sharing schematics with others.

Introduction

LTspice® makes it easy to create and simulate schematics quickly—sometimes the best starting point for hashing out a design is using ideal circuit elements. However, a circuit designer will need to improve the initial simple schematic with more realistic component models.

LTspice ships with an extensive collection of third-party manufacturer models. To use one of these models, simply right-click the component, then click the **Pick...** or **Select** button in the properties window and select one of the models listed. See Figure 1.

directive. This article will provide guidance on importing both model types.

Note: If the imported model file is encrypted, it may be difficult to determine if the model was implemented with **.MODEL** or **.SUBCKT** directives. Contact the model vendor for support with encrypted models or post your issue on the LTspice EngineerZone® forum, someone in the EZ community might be able to help.

Each of the examples below is included in the LTspice-importing-third-party-models.zip file available for download here.

Importing a .MODEL Directive

For a device that is modeled with a **.MODEL** directive, import-

Coming in LTspice 26

- Improved convergence
- Loadstate, savestate, and convergence report are available in .tran/.fra configuration dialog
  - No longer require separate directives
- Alleviate hierarchy pain points
  - Auto-generated symbols from .subckt no longer include directory path in ModelFile attribute (improves portability)
  - Symbols that have simulation library file dependencies are indicated in netlist comments (easier debugging)
  - Generate Subckt Netlist from a schematic (faster modeling)
    - Optionally use a symbol to define the port order
- Add unicode support for file paths in settings
- Left-click any unconnected pin to start a wire in a schematic
- Improve display of recent files in File menu
  - Show full path in status bar when hovered

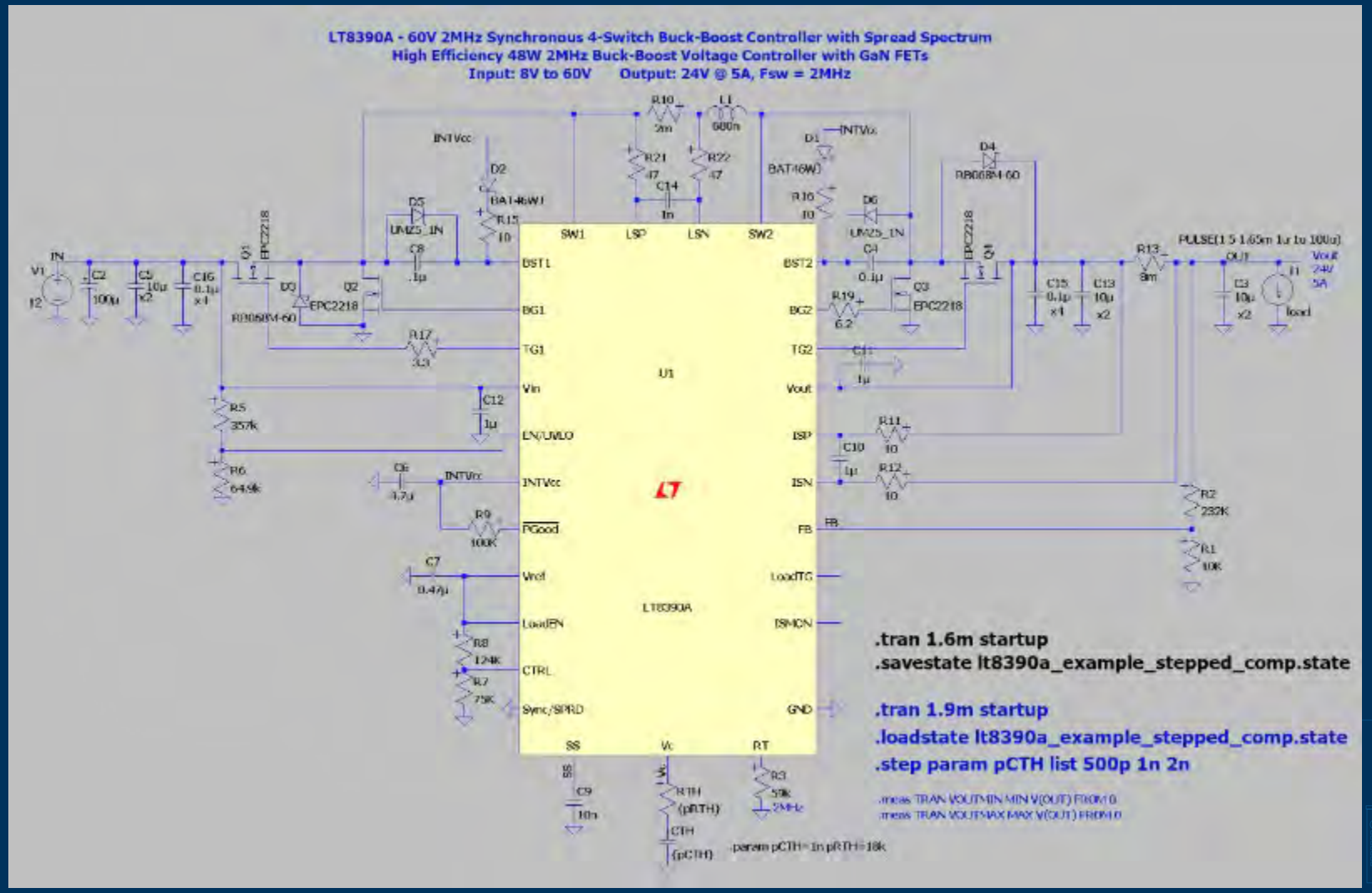
Save and Load Transient Simulation State

About the .savestate & .loadstate directives

- Capture the *complete* simulation state:
- `.savestate <statefilename> [time=<value>]`
- If no time is specified, the state is captured at the end of the simulation
- Reload the state:
- `.loadstate <statefilename> [reset]`
  - Simulation time jumps to the saved point, or can be reset to zero
  - Components cannot be added, removed, or rewired
  - *Component values can be changed* and LTspice will attempt to make it work
- Supports .tran and .fra

Save and Load Transient Simulation State

Example: Transient response with varying compensation cap values



Save and Load Transient Simulation State

Example: Transient response with varying compensation cap values

```
.tran 1.6m startup
.savestate lt8390a_example_stepped_comp.state
```

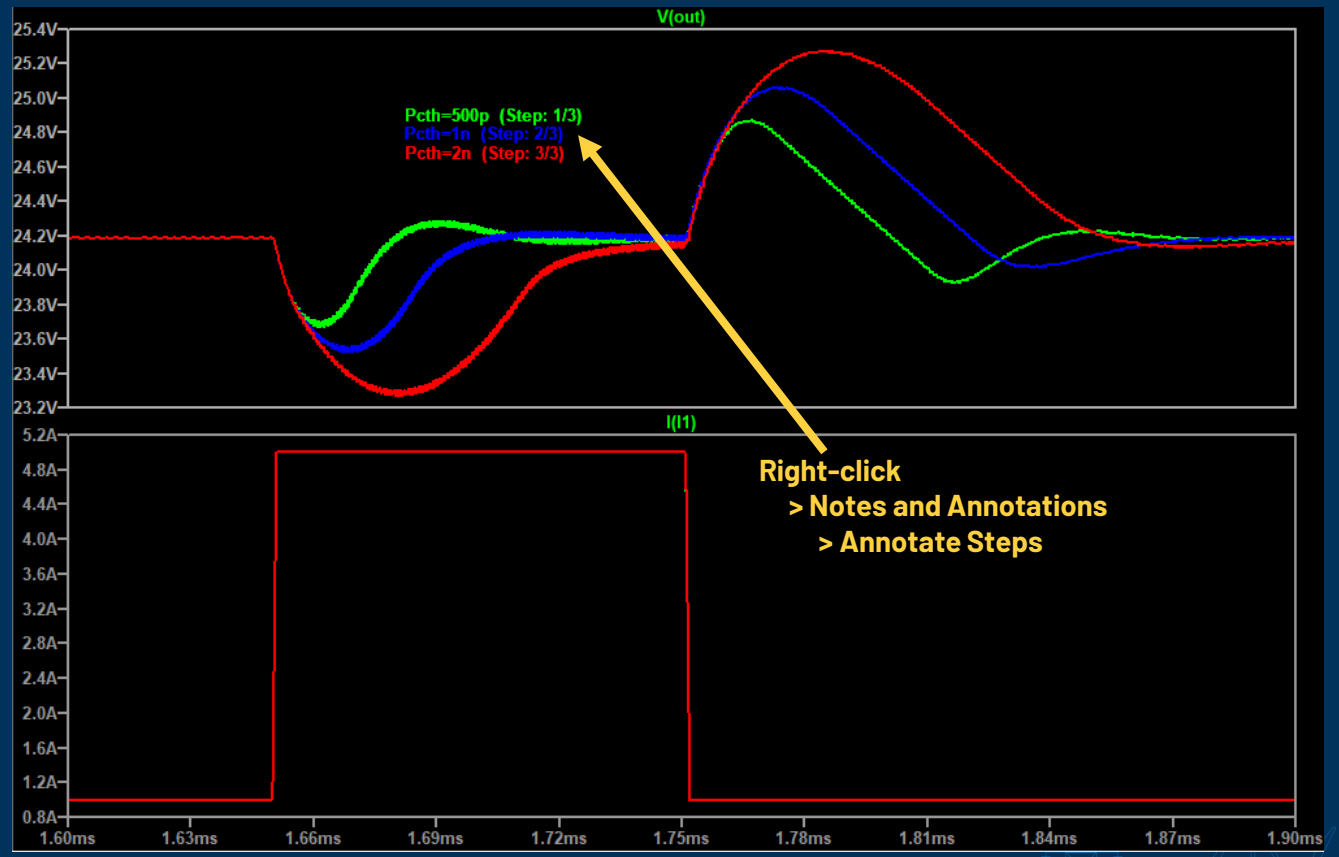
```
.tran 1.9m startup
.loadstate lt8390a_example_stepped_comp.state
.step param pCTH list 500p 1n 2n
```

Initial sim for .savestate after soft-start

Can optionally be stopped manually

Subsequent sim uses .loadstate to skip soft-start, with .step directive to vary pCTH

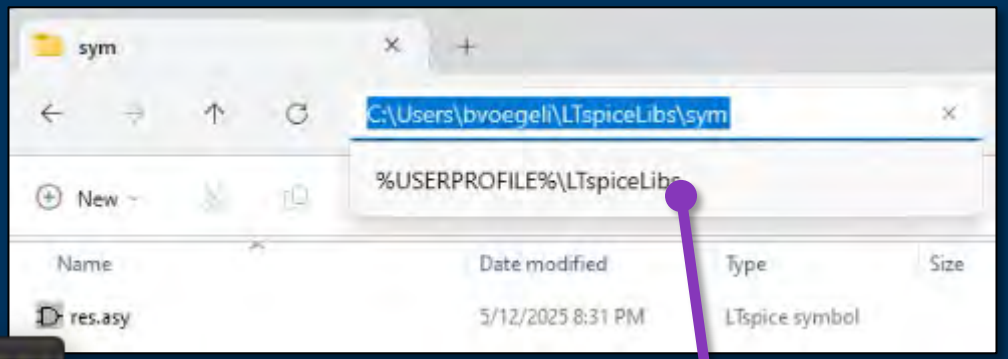
Clock time reduced by ~9 minutes

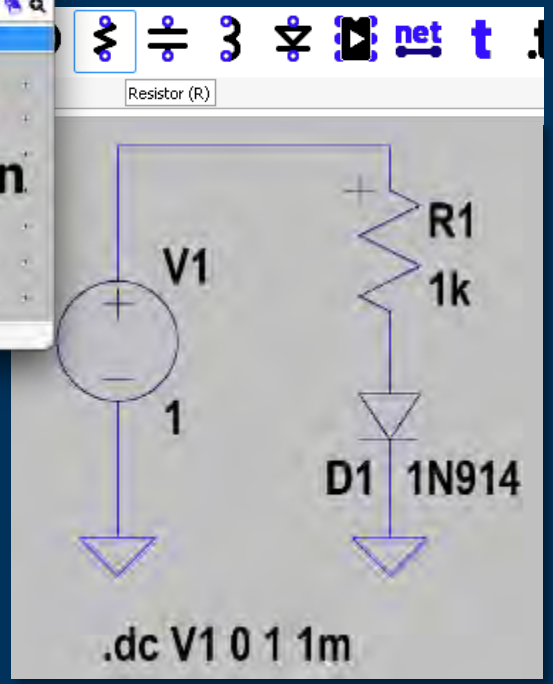


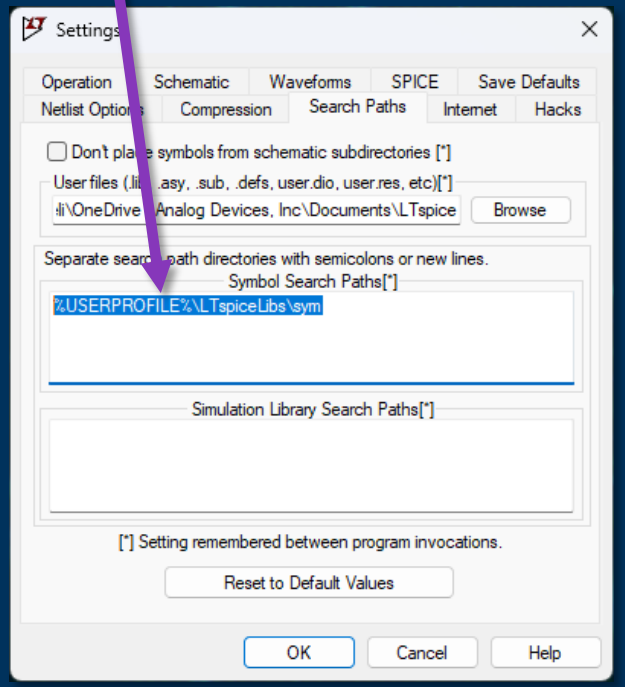
Overriding an Intrinsic Symbol

Example: Use a custom `res` symbol that indicates current flow direction

Custom Resistor







Set Path to Symbols in Settings

The replacement symbol *must exist at the* ***root*** *of a defined path*

Symbol Priority	
1	Schematic Directory
2	Settings > User Files
3	*Settings > Search Paths > Symbol Directories*
4	ADI Library

Do not place or edit files in %LOCALAPPDATA%\LTspice

Common Problems & Solutions

Convergence Problems — Symptoms

Simulation stops with an error

- Failure to find initial operating point
- Time step too small
- Singular matrix

Simulation runs extremely slowly

- Small time steps / slow transient simulation speed
- Very slow Pseudo Transient while finding initial operation point

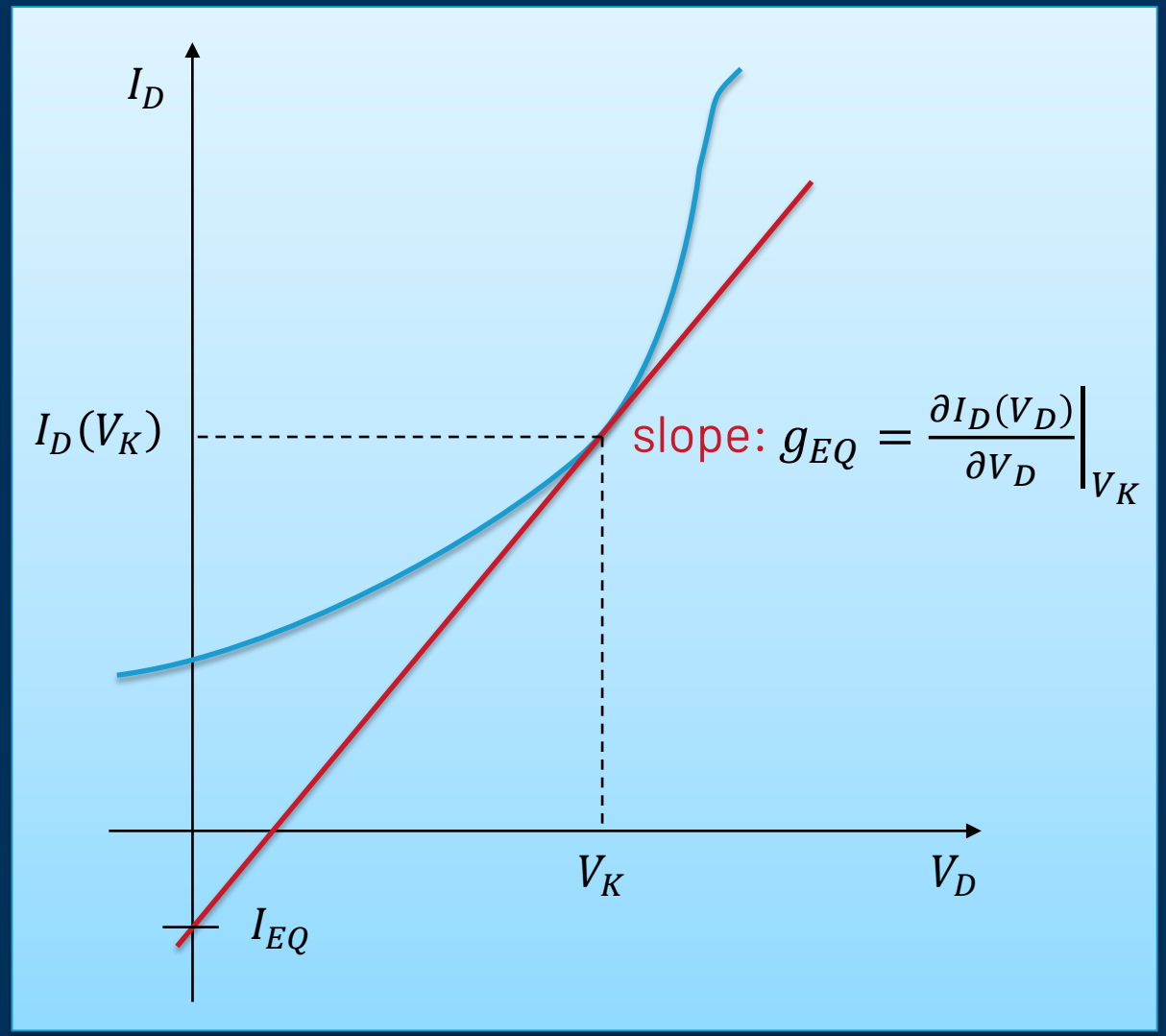
It is common for a borderline circuit to converge or fail due to a minor and/or unrelated change.

Nonlinear Elements

$$I_D(V_D) = I_S\left(\exp\left(\frac{V_D}{V_T}\right) - 1\right)$$

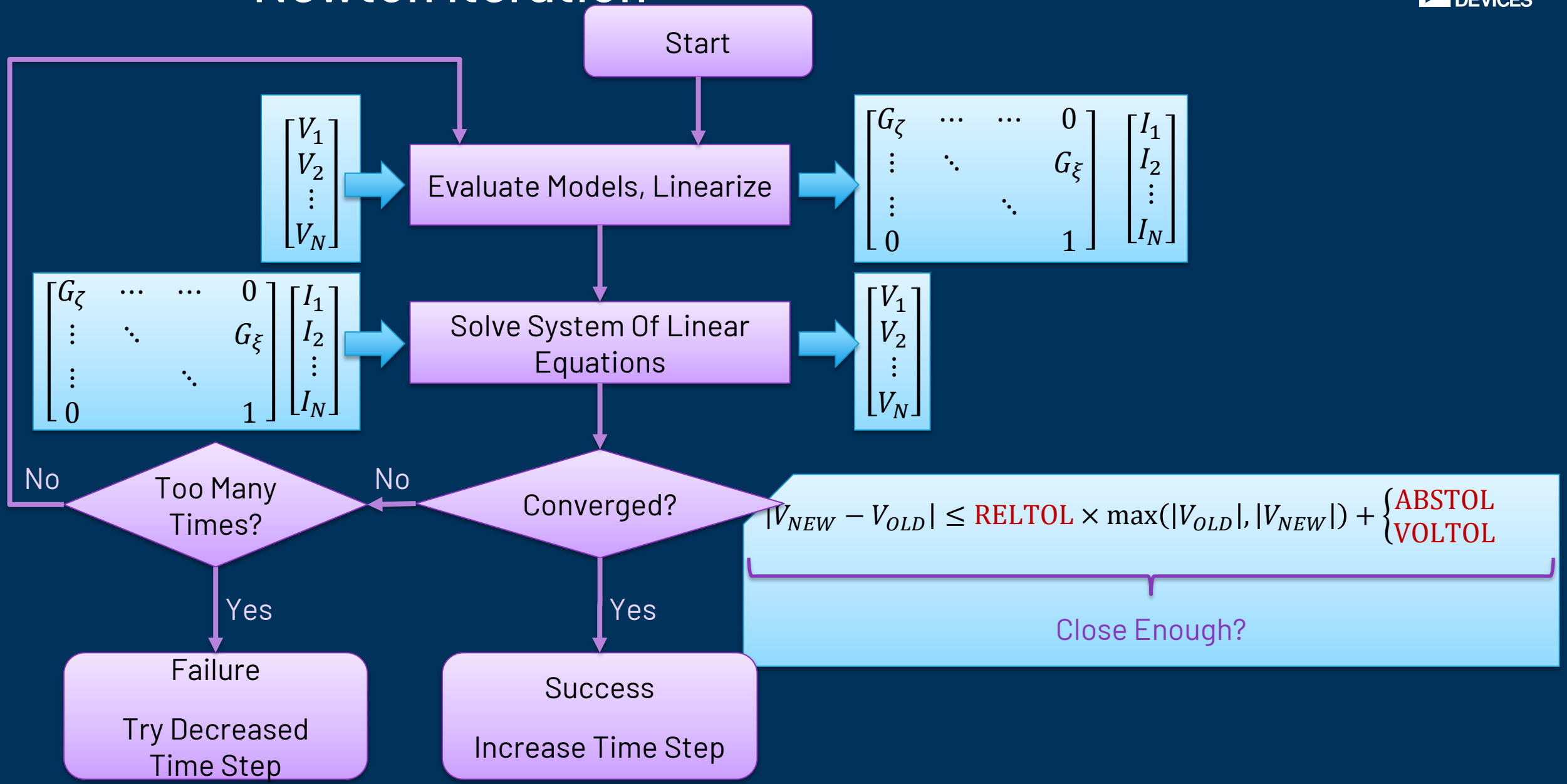
linearized:

$$I_D(V_D) = g_{EQ} \times V_D + I_{EQ}$$



Also used for .ac + .noise

Newton Iteration



Common Sources of Convergence Problems

- Discontinuities in component I-V curve
- Instantaneous changes
- Discontinuities in first derivative of component behavior
- Non-physical extremely high-impedance nodes (e.g. nodes with zero capacitance)
- Non-physical immediate local feedback

The preferred solution is always to fix the problematic circuitry.

Sometimes the problem may be in circuitry you can't change (such as an encrypted model)…

… but there are some things that may still help.

Improve Convergence — Debug the Circuit

Add `.options debugtran` to the schematic *(new in LTspice 24.1)*

- This adds a report to the log file

```
Device Convergence Difficulty Score:
m5:gcapgd    : 55.276382
m3:gcapgd    : 54.522613
…
Node Convergence Difficulty Score:
s3#current   : 76.884422
s4#current   : 61.557789
…
```

- Debugtran slows down the simulation
- Scores below 1 are usually negligible
- Scores above ~50 are a sure sign of a problem

Improve Convergence — Fix the Circuit

If you can change the problematic circuitry, here are some things to try:

- Add capacitance, especially for high impedance nodes. Even 1e-15 or 1e-18 can help.
- Avoid discontinuities
  - For ideal diodes, add (or increase) the `epsilon` parameter. Increase series resistance `ron`.
  - Define `trise` for all logic gates (as large as possible).
  - Avoid extreme non-physical values
    - For example, don't set emission coefficient n=0.001 for a silicon diode (should be in the range 0.5 ~ 2)
- Avoid instantaneous or extremely tight feedback
  - For example, don't use a behavioral current source to model a capacitor, don't tie logic gate output back to its input
- Add series resistance to signal voltage sources using the `rser` parameter
  - Also add a parallel capacitance using `cpar`
  - Not recommended for power supply input sources
  - Not recommended if current monitoring is important

Improve *DC Operating Point* Convergence — Simulator

Increasing values compromises accuracy for the sake of convergence.

- Add `startup` keyword to `.tran` directive. Ramps top-level independent sources up from zero over first 20µs
- `.options solver=alt` : Use alternate solver. Higher precision, slower.
  Not available for ARM processors. Impractical for switching regulators due to slow speed. Generally, *not* more accurate.
- Add `.nodeset` directive to suggest initial operating points
- `.options abstol=1e-10` : Increase absolute current convergence tolerance.
  Default=1e-12, try values from 1e-11 to 1e-9.
- `.options gshunt=1e-15` : Add a conductor from every node to ground, including internal subcircuits.
  Default=0, try values from 1e-21 to 1e-9.
- `.options reltol=0.005` : Increase relative convergence tolerance.
  Default=0.001, try values from 0.002 to 0.05.
- `.options gmin=1e-10` : Add a conductor in parallel with all PN diodes.
  Default=1e-12, try values from 1e-21 to 1e-9.

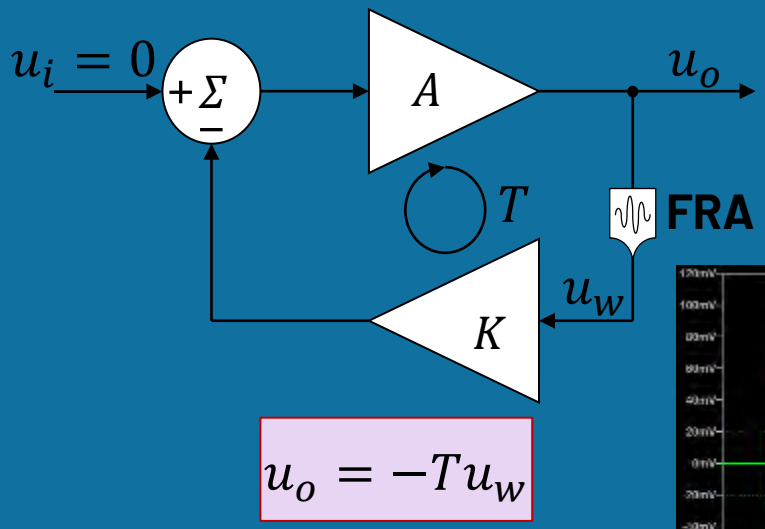
Improve *Transient* Convergence — Simulator

With the exception of `itl4` increasing values compromises accuracy.

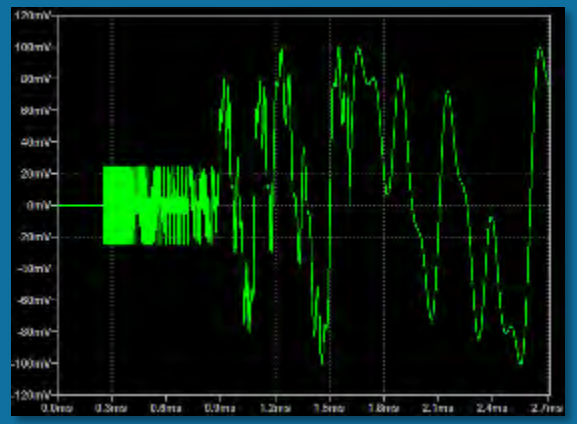
- `.options solver=alt` : Use alternate solver. Higher precision, slower.
  Not available for ARM processors. Impractical for switching regulators due to slow speed. Generally, *not* more accurate.
- `.options method=gear` : Use Gear integration. Larger truncation error, tends to dampen oscillations, usually slower.
- `.options itl4=50` : Transient iteration limit. Default=10, try values from 20 to 500. This only works properly in version 24.1 and beyond. No accuracy penalty, but may slow down simulations.
- `.options maxstep=1n` : Maximum time step. Can also set this value in `.tran` directive. Default=tstop/1024.
- `.options cshunt=1e-15` : Add a capacitor from every node to ground, including internal subcircuits.
  Default=0, try values from 1e-21 to 1e-12.
- `.options abstol=1e-10` : Absolute current convergence tolerance. Default=1e-12, try values from 1e-11 to 1e-9.
- `.options gshunt=1e-15` : Add a conductor from every node to ground, including internal subcircuits.
  Default=0, try values from 1e-21 to 1e-9.
- `.options reltol=0.005` : Relative convergence tolerance. Default=0.001, try values from 0.002 to 0.05.
- `.options gmin=1e-12` : Add a conductor in parallel with all PN diodes. Default=0, try values from 1e-21 to 1e-9.
- Use `.savestate` and `.loadstate` to temporarily modify tolerances for only a portion of a sim.

Frequency Response Analysis (FRA)

- Transient Simulation
- The FRA component injects sinusoidal stimulus

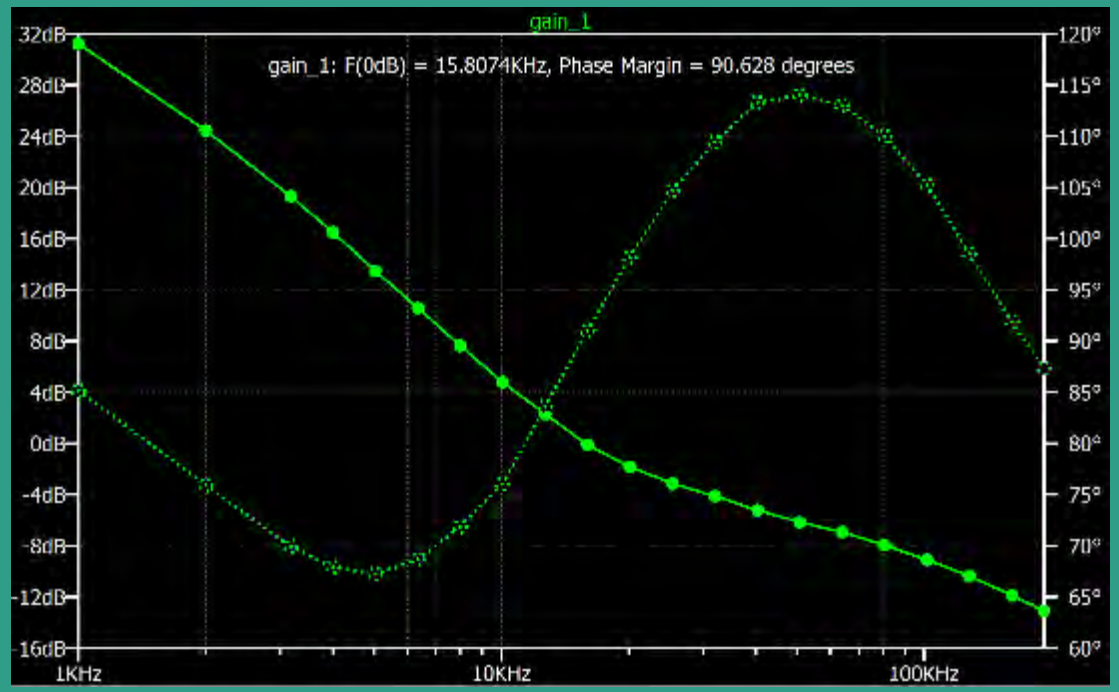


$$u_o = -Tu_w$$



time

- Perturbation is measured at FRA nodes
- Bode plot is extracted by Fourier analysis

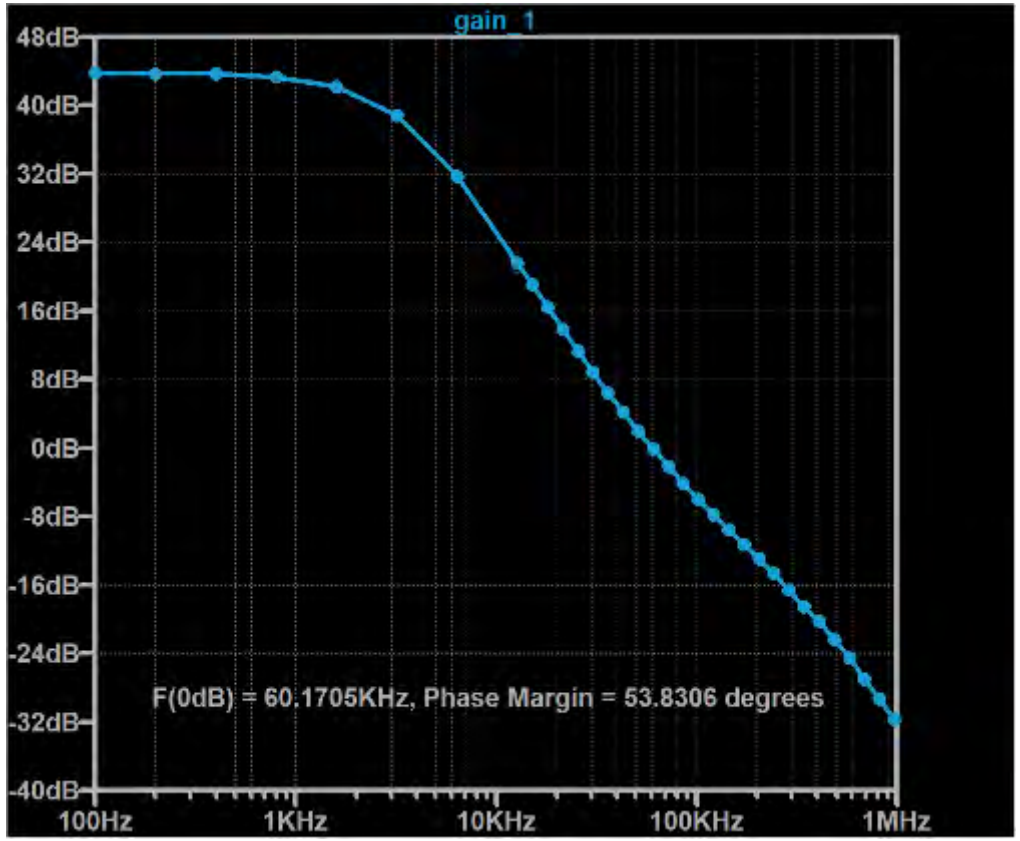


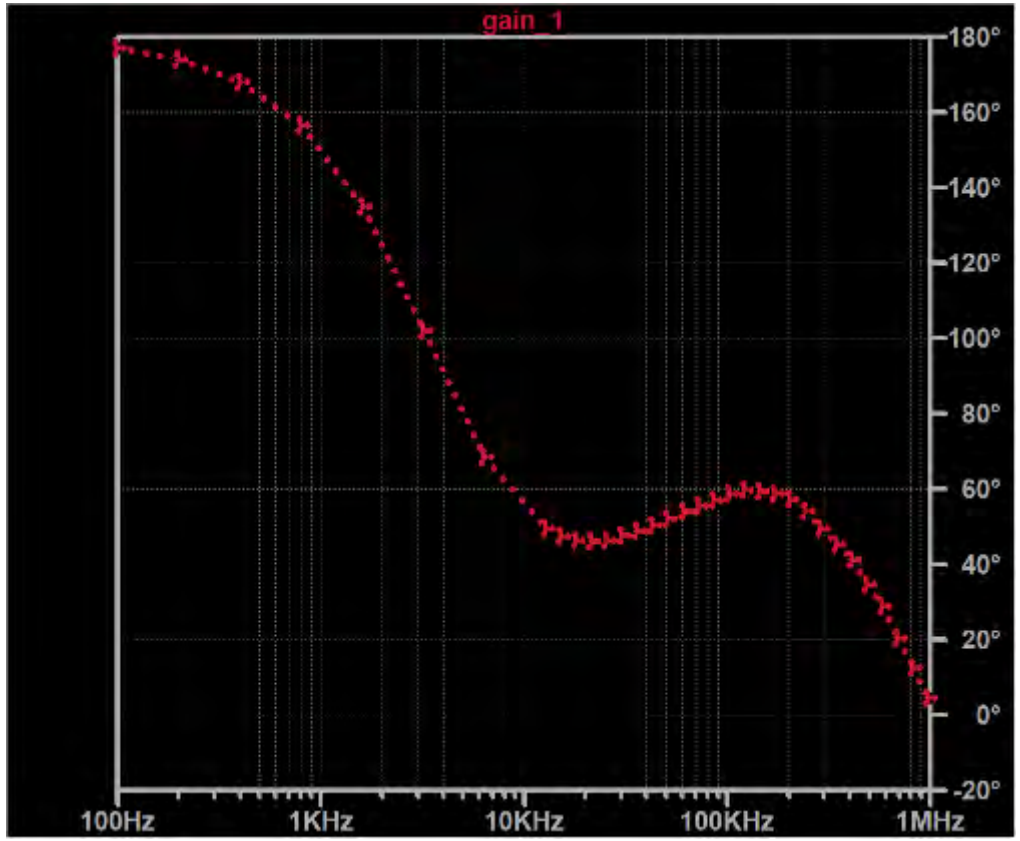
frequency

see LTspice 24 and FRA How-To Webinar

Generating SMPS Bode Plots in LTspice

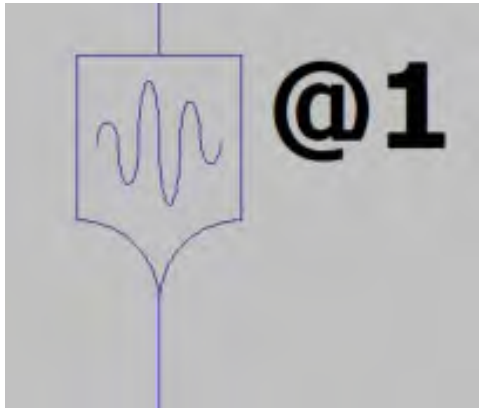
Bode Plots in LTspice



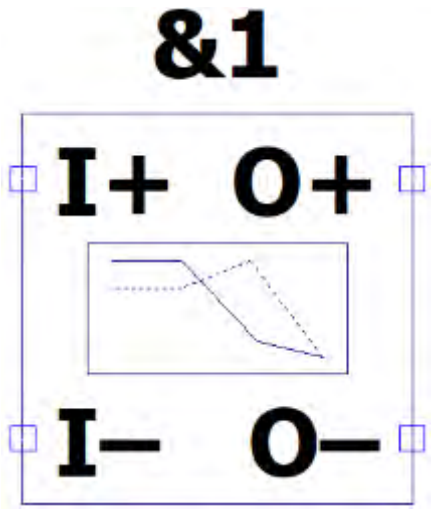


Frequency Response Analysis (FRA) Components in LTspice

FRA Device

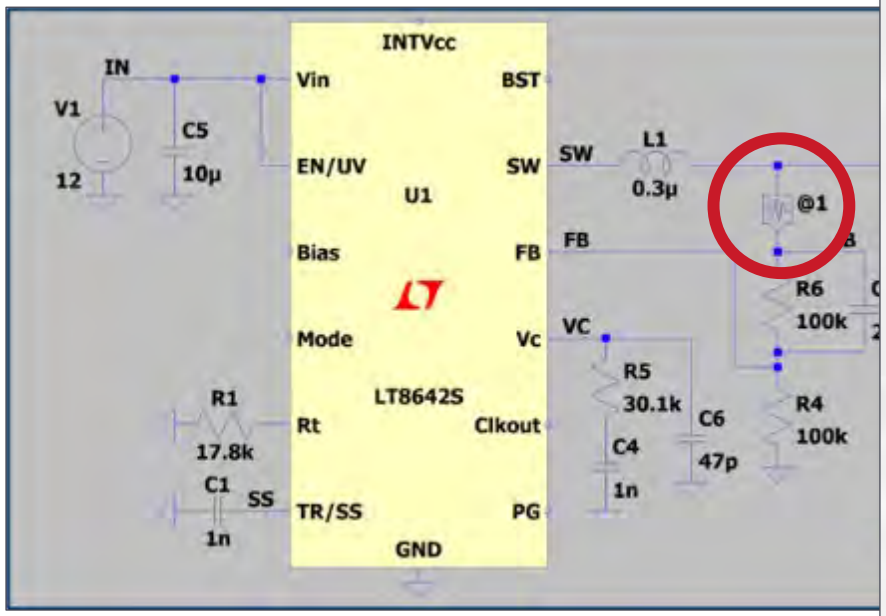


FRA Probe

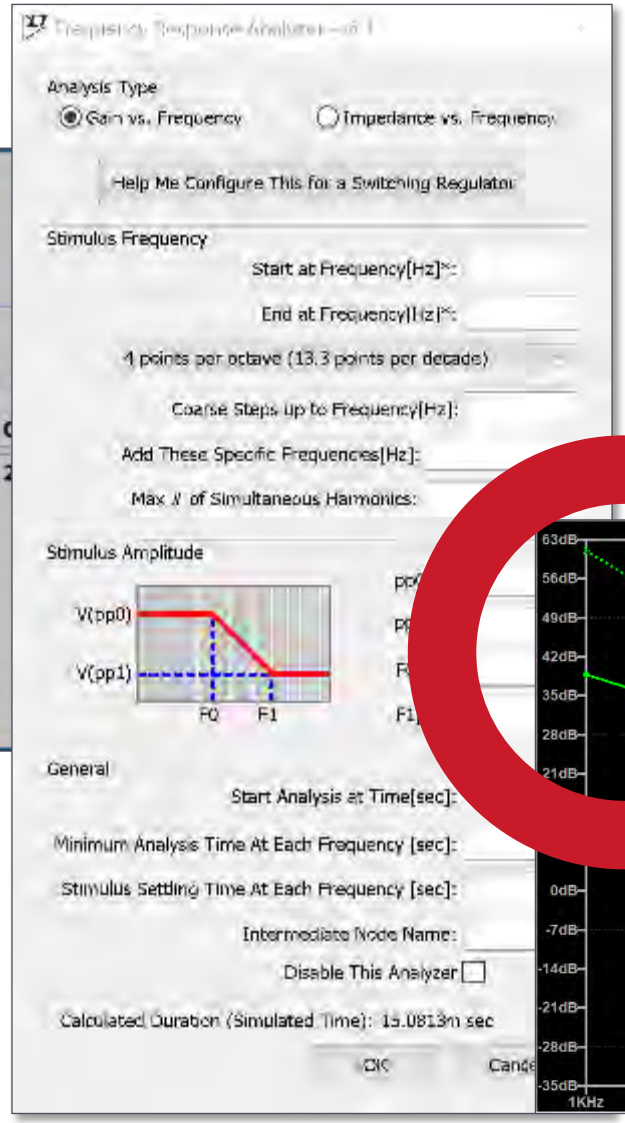


How to use the Frequency Response Analyzer Tool

How To FRA

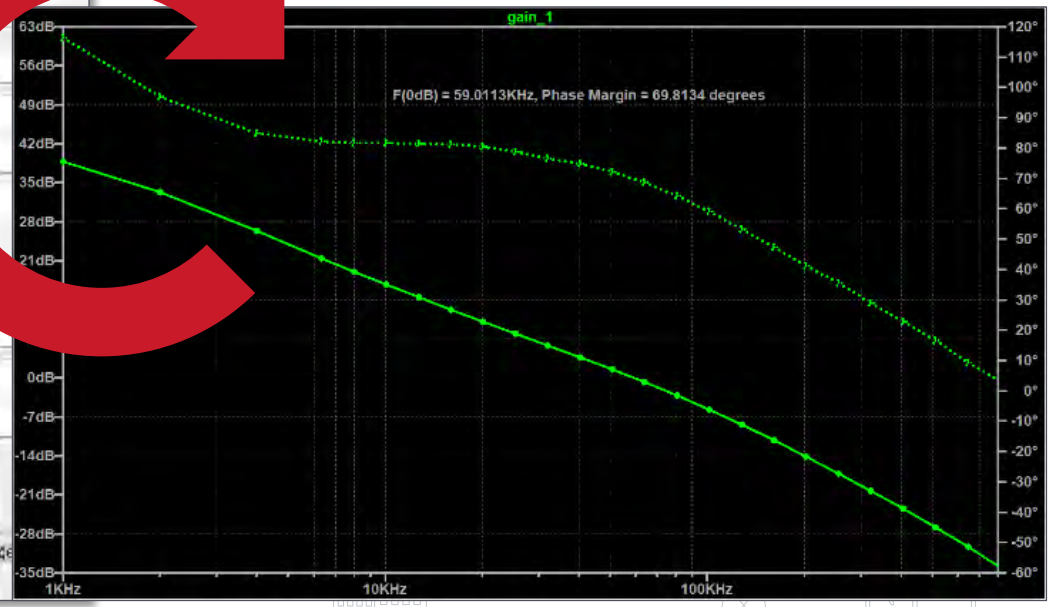


Place FRA Device



Configure FRA

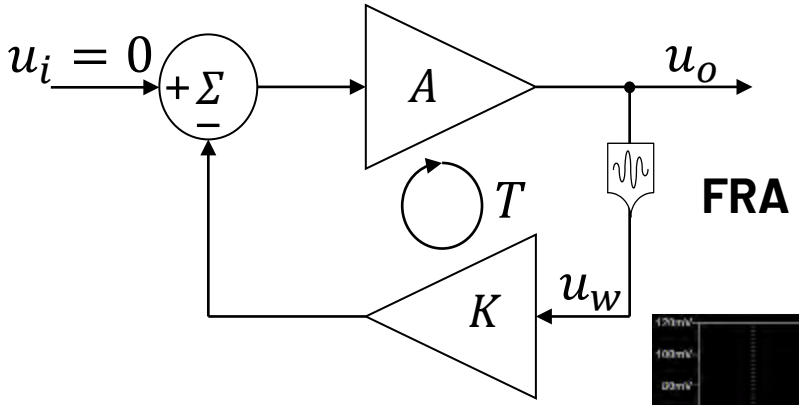
Iterate



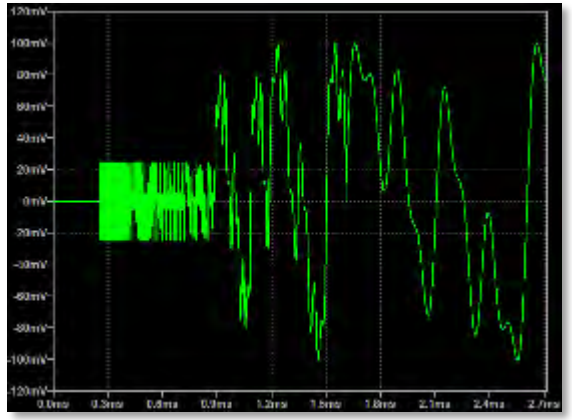
Simulate

What does the FRA Device do?

- Transient Simulation
- The FRA component injects sinusoidal stimulus

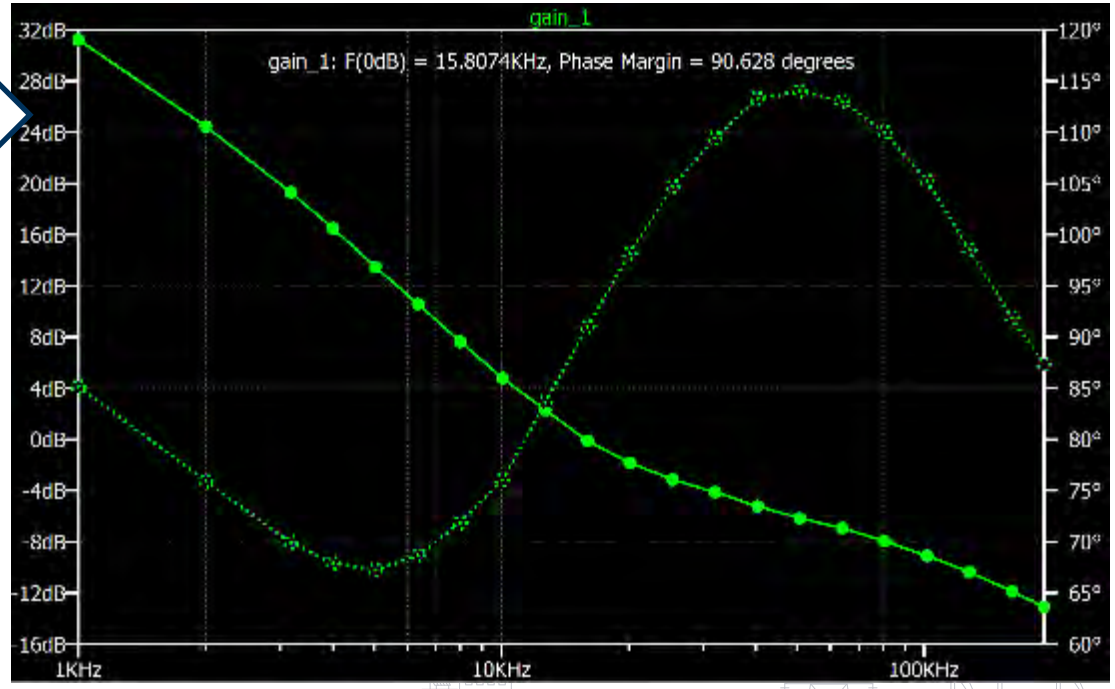


$$u_o = -Tu_w$$



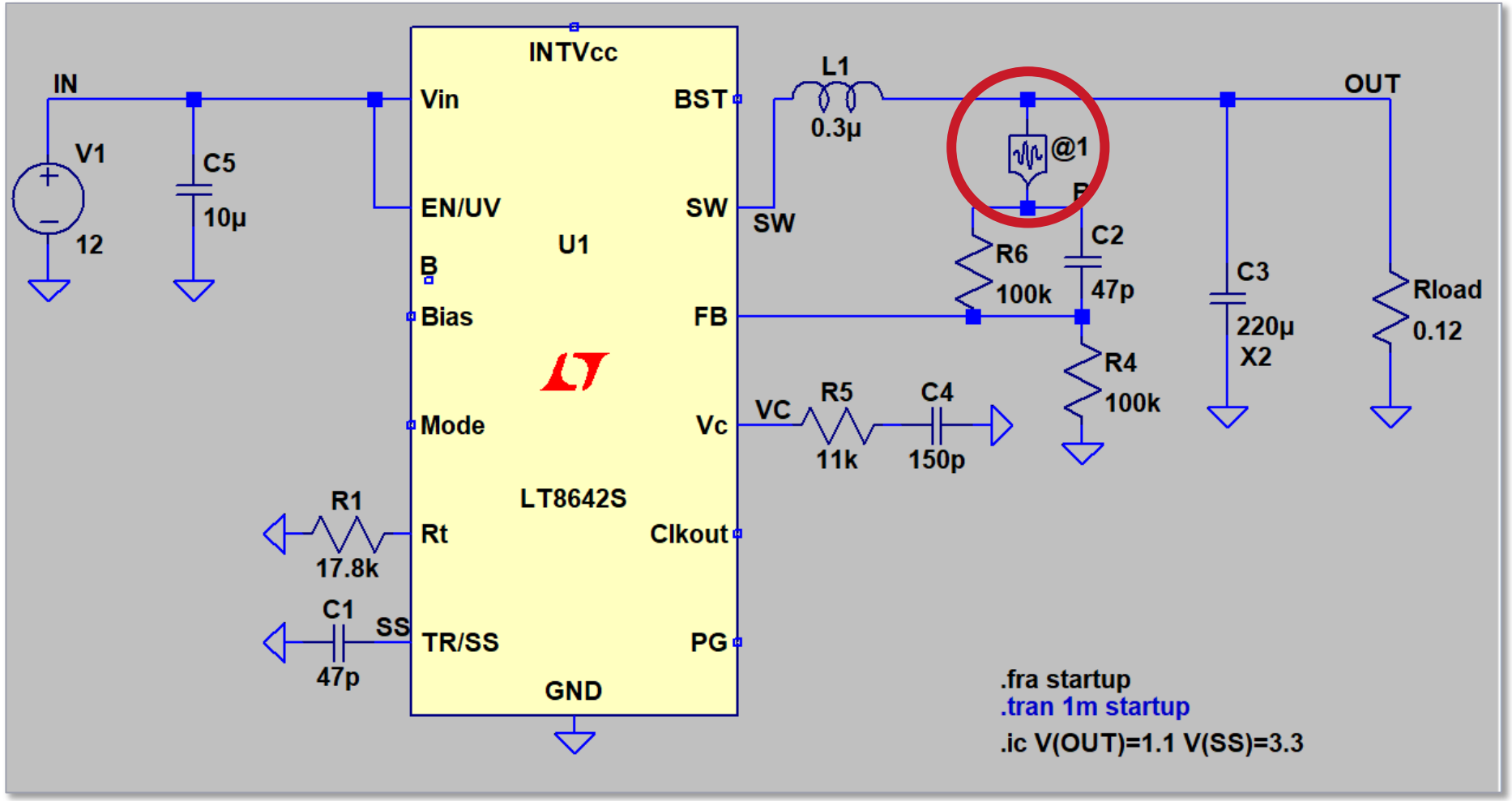
Time

- Perturbation is measured at FRA nodes
- Bode plot is extracted by Fourier analysis



Frequency

For many circuits, FRA Device is all that is needed

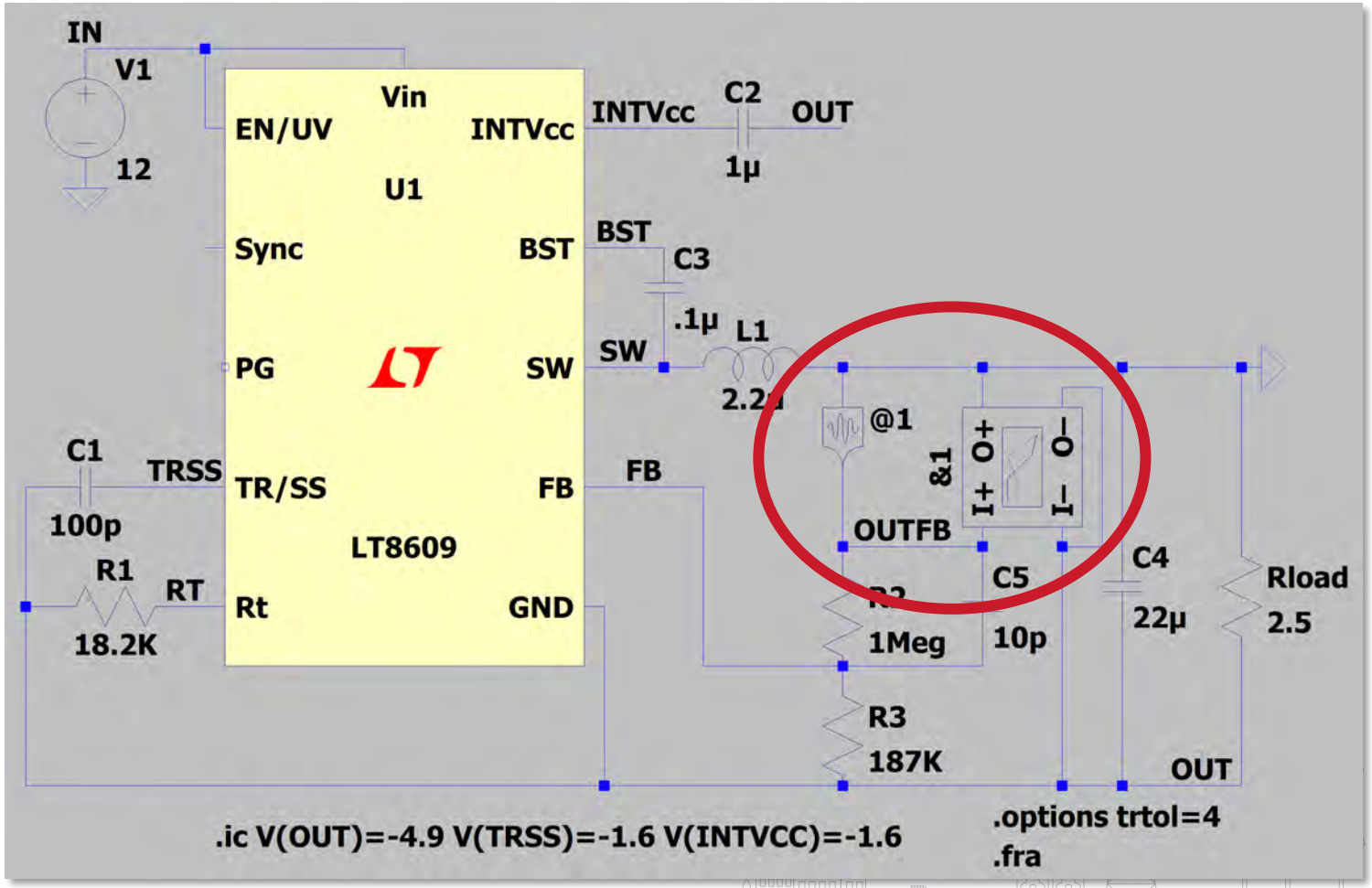


Typical Buck Example – including FRA Device

What does the FRA Probe do?

FRA Probe used along with FRA Stimulus

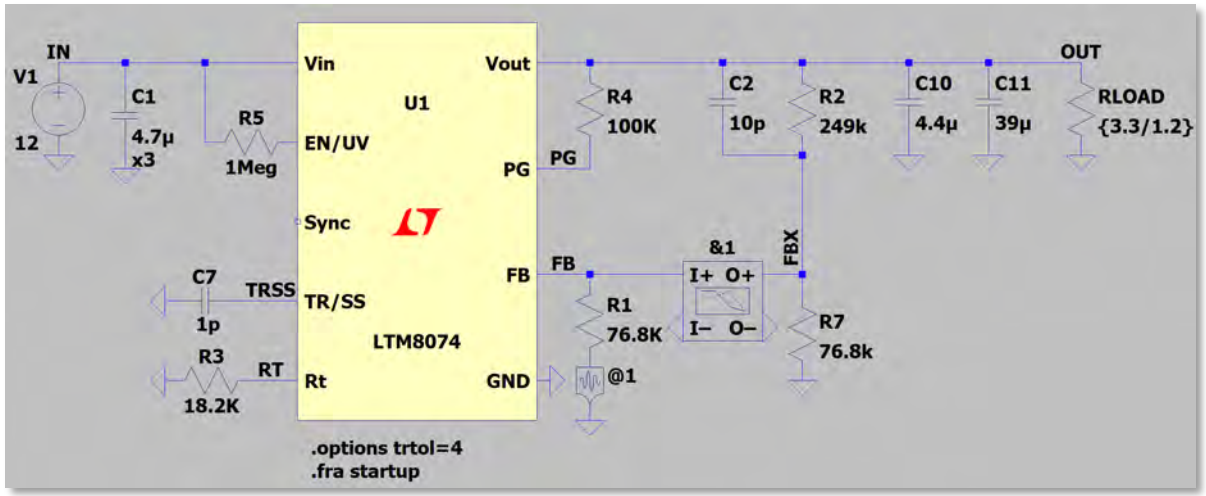
- Enables Bode Plots of any portion of the loop
- Enables analysis of uModules, Inverting Outputs, Current Feedback topologies



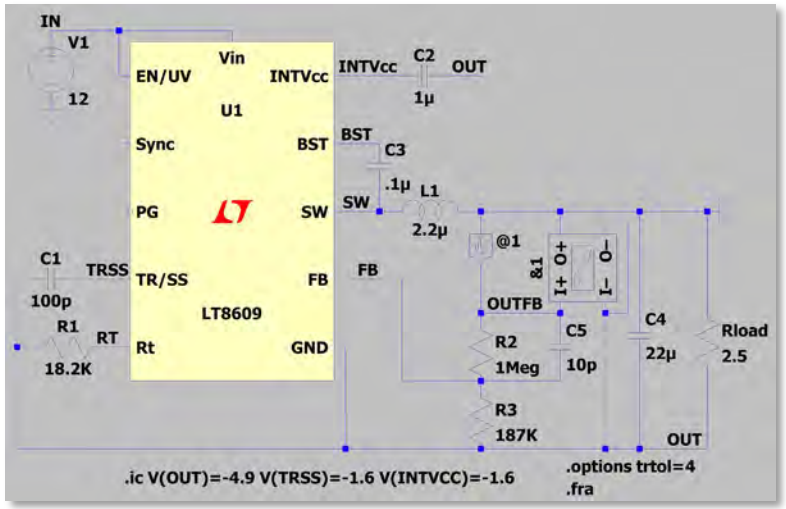
Inverting Output Example – requires FRA Probe

What does the FRA Probe do?

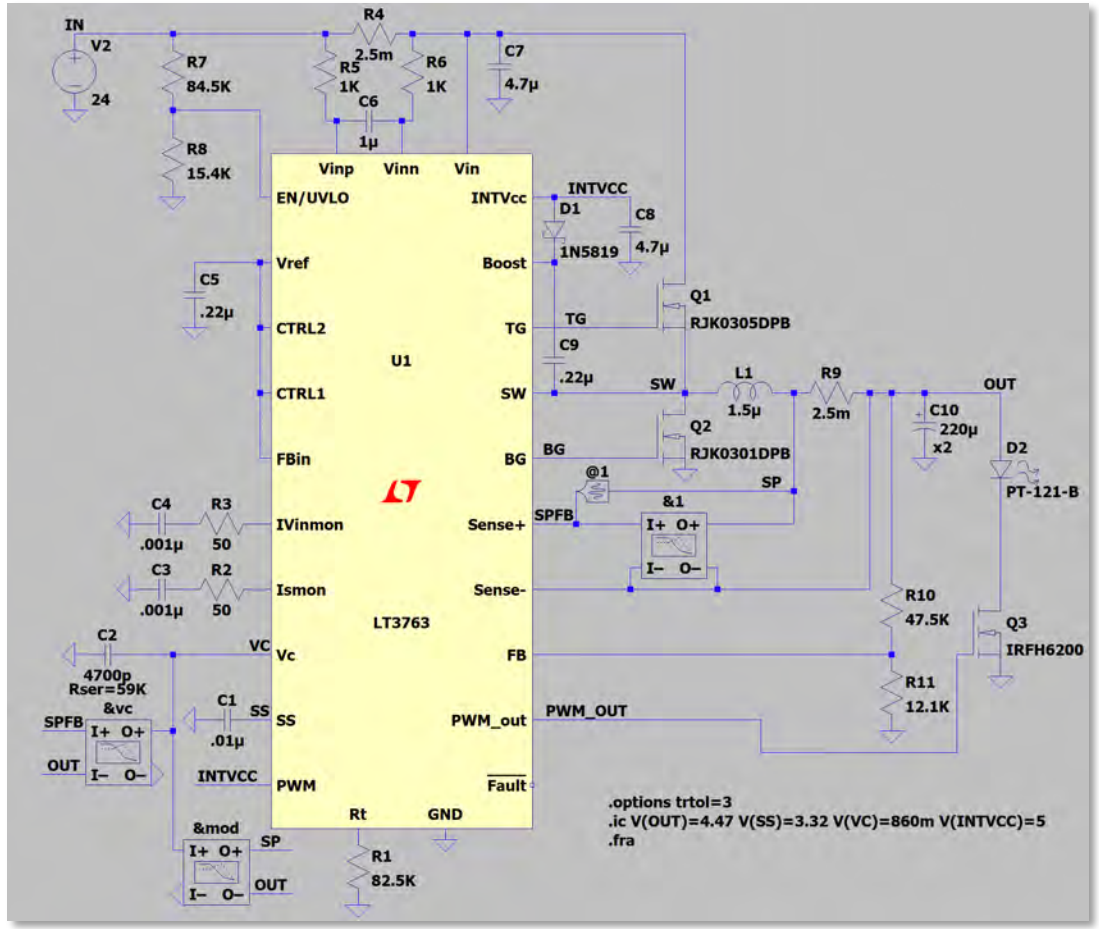
µModules with integrated top feedback resistors



Negative Outputs

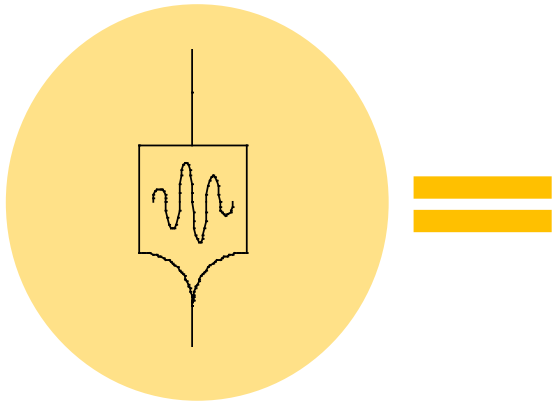


Current Feedback

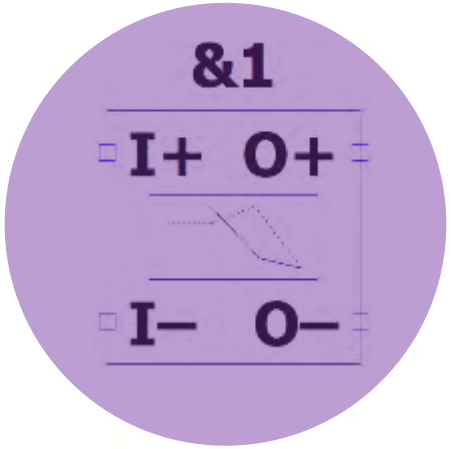


These examples are in
Open Examples -> Educational -> FRA

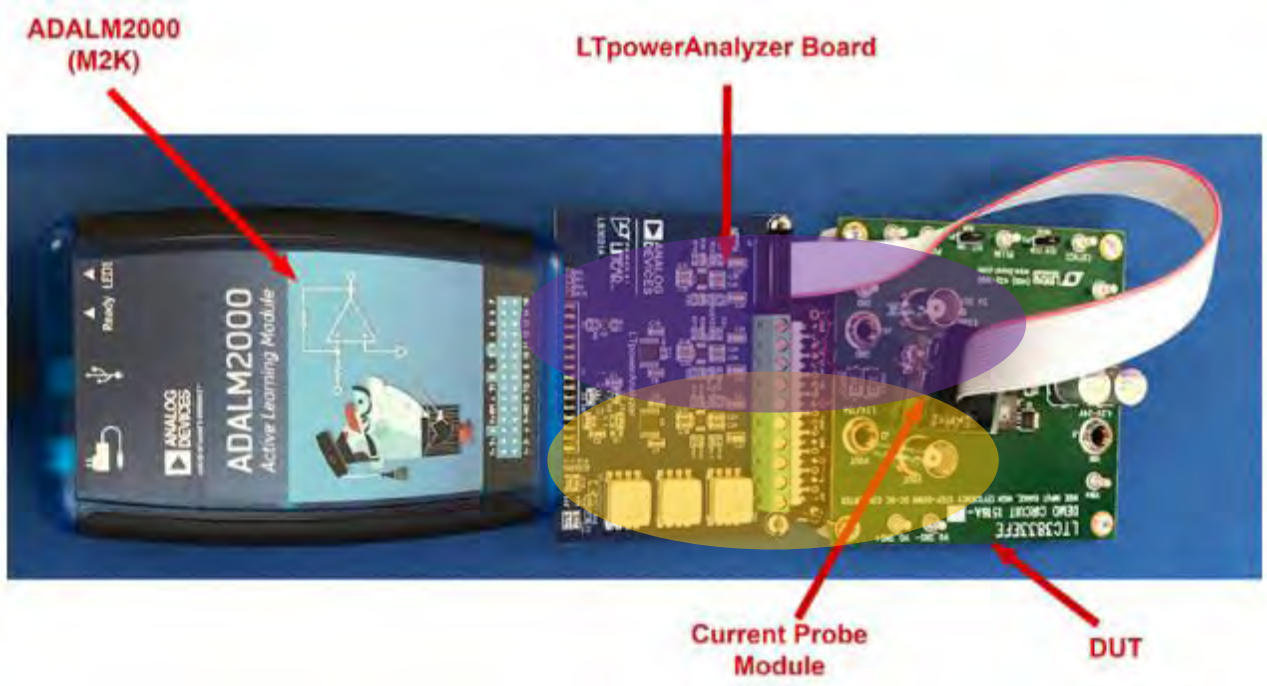
Functional FRA “Equivalencies”



=

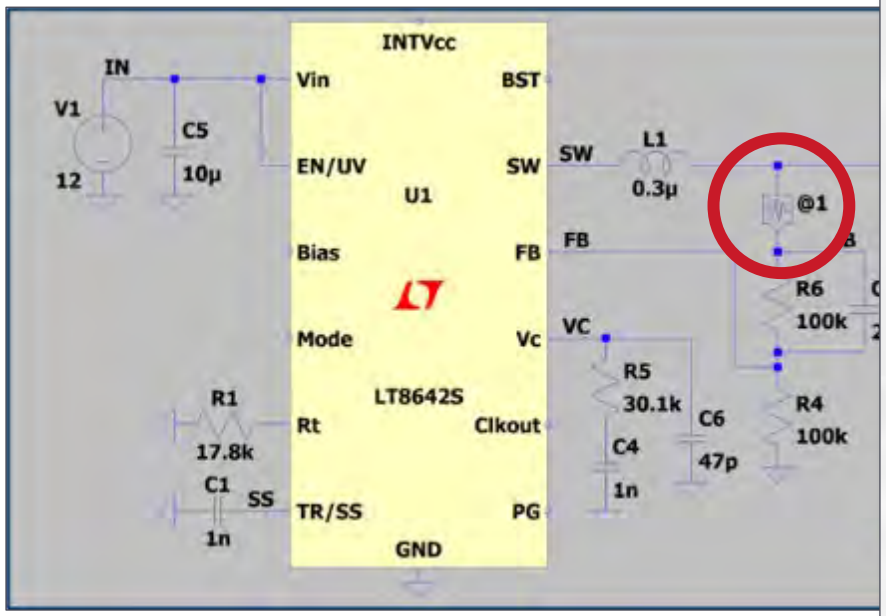


=

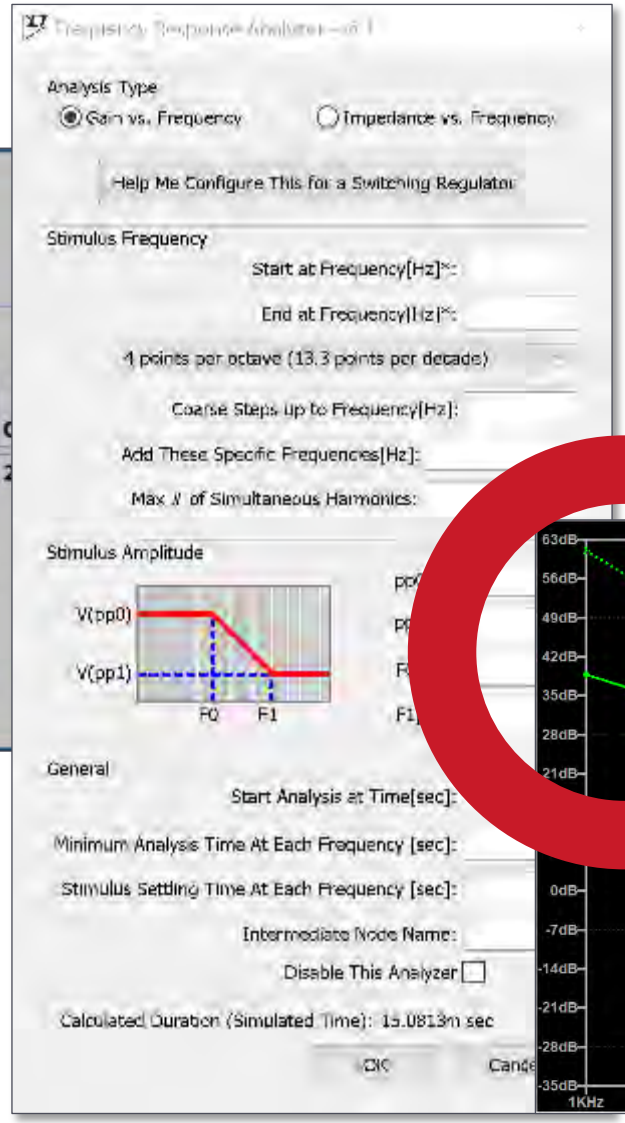


FRA Settings Overview

How To FRA

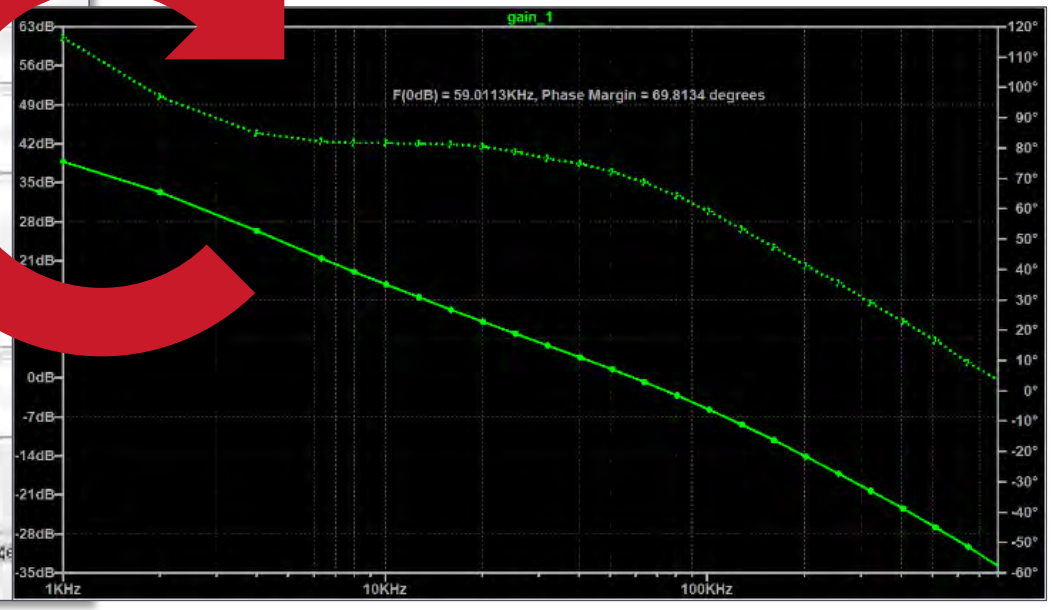


Place FRA Device



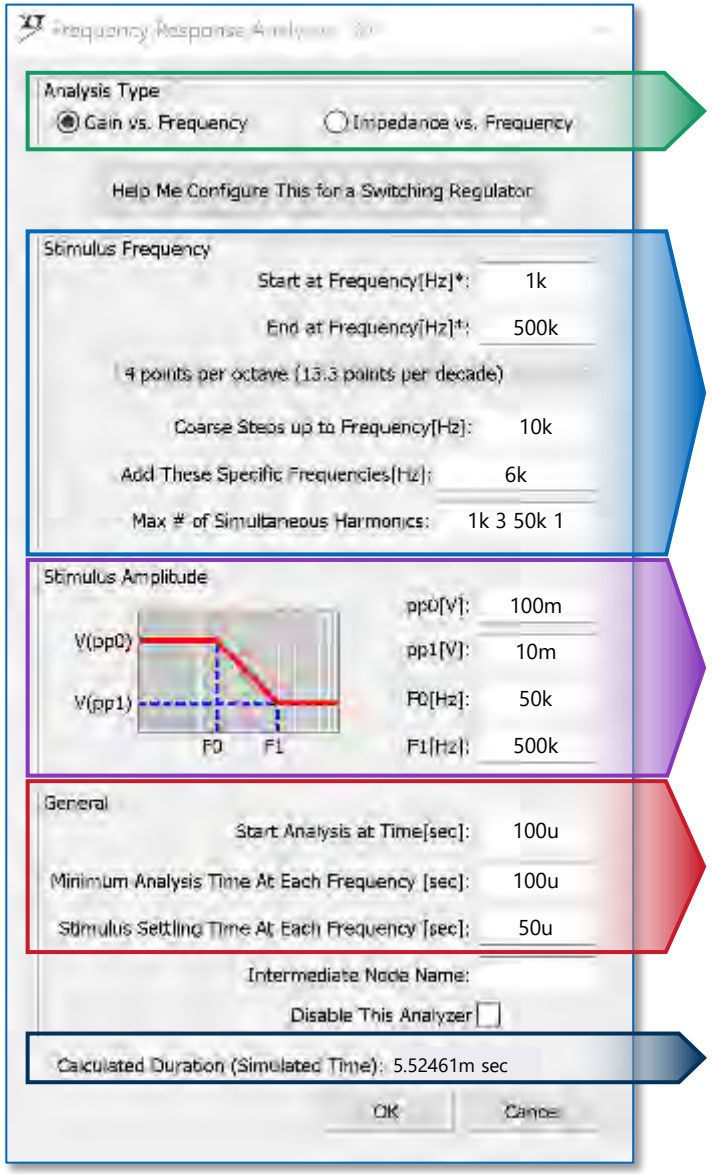
Configure FRA

Iterate



Simulate

FRA Device Settings: Overview



Bode Plot of **Gain** or **Output Impedance** Sweep

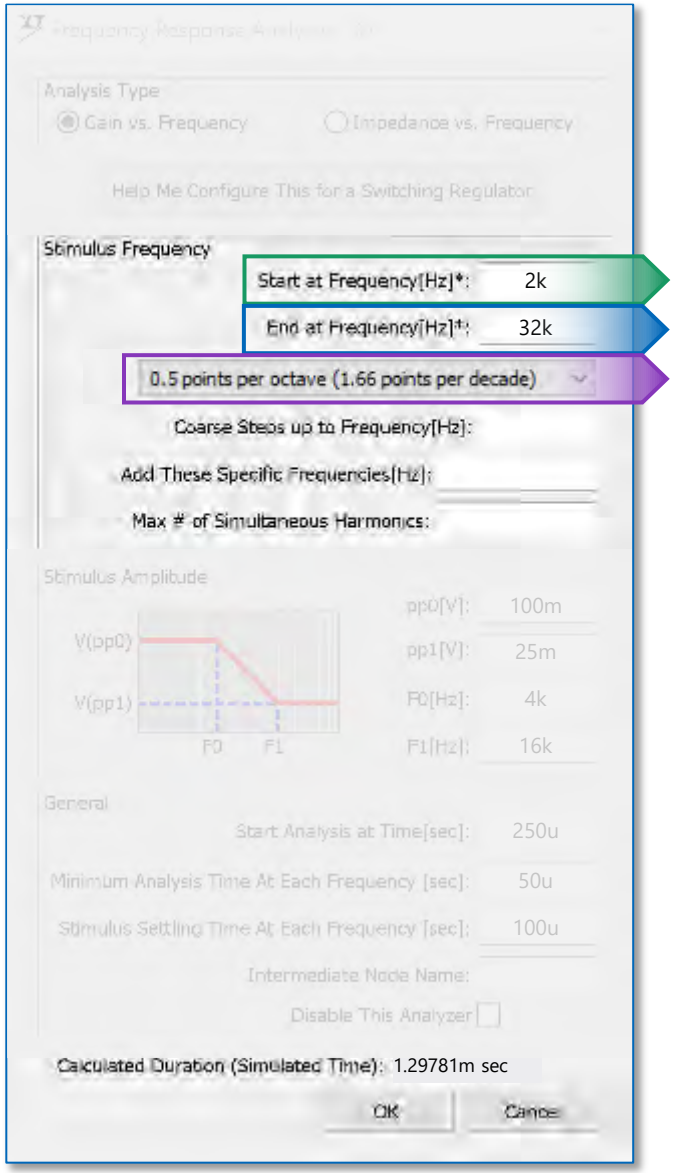
Set the **frequencies** applied in the analysis

Set stimulus **amplitude** as a function of frequency

Set the **timing** of analysis and stimulus transitions

Indicates simulated time based on settings above

FRA Device Settings: Required Frequency Settings

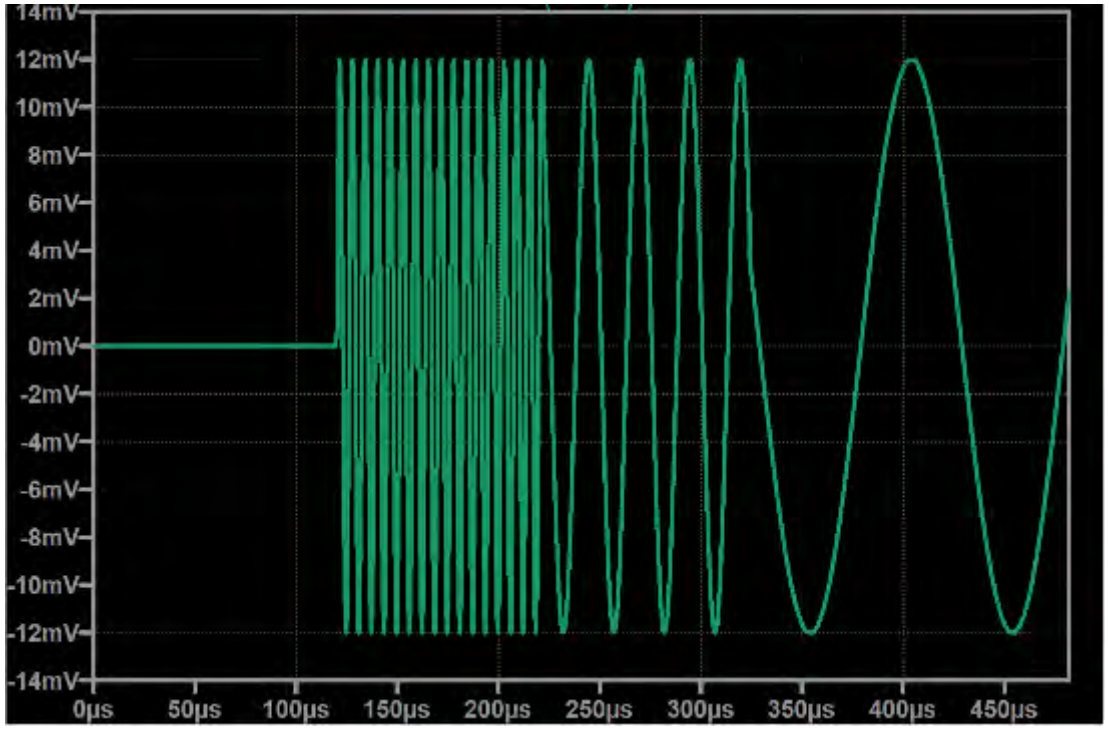


Lowest Frequency

Highest Frequency

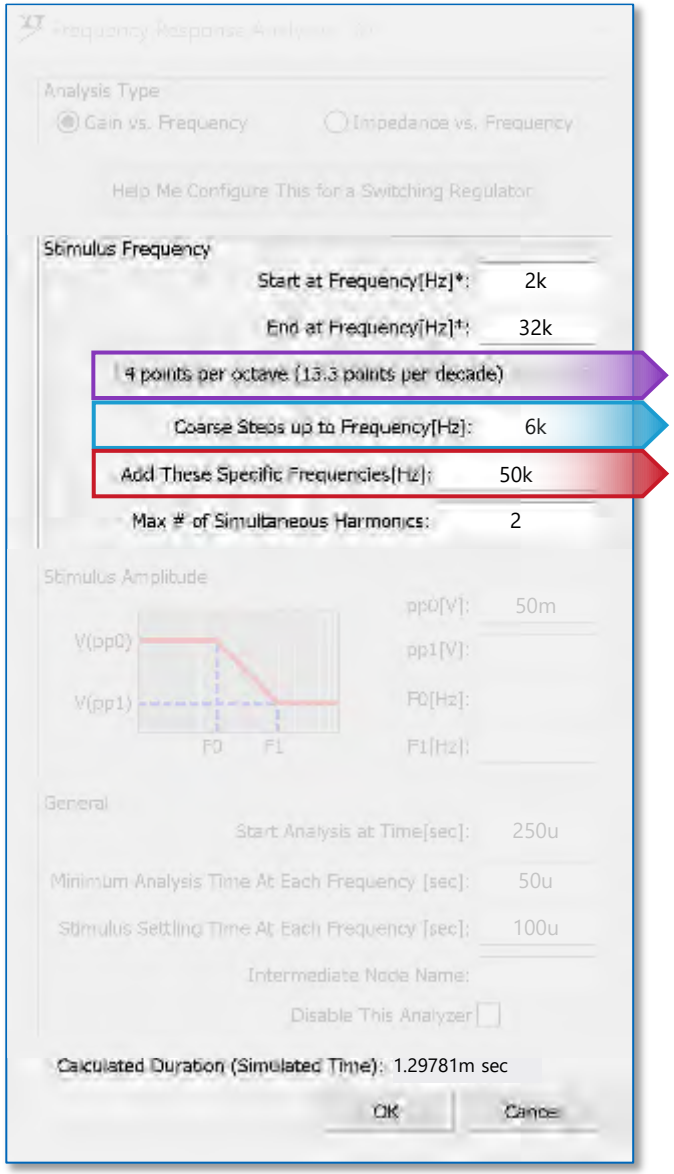
# of Frequencies

Specified in octaves so LTspice can apply simultaneous harmonics to speed up the simulation

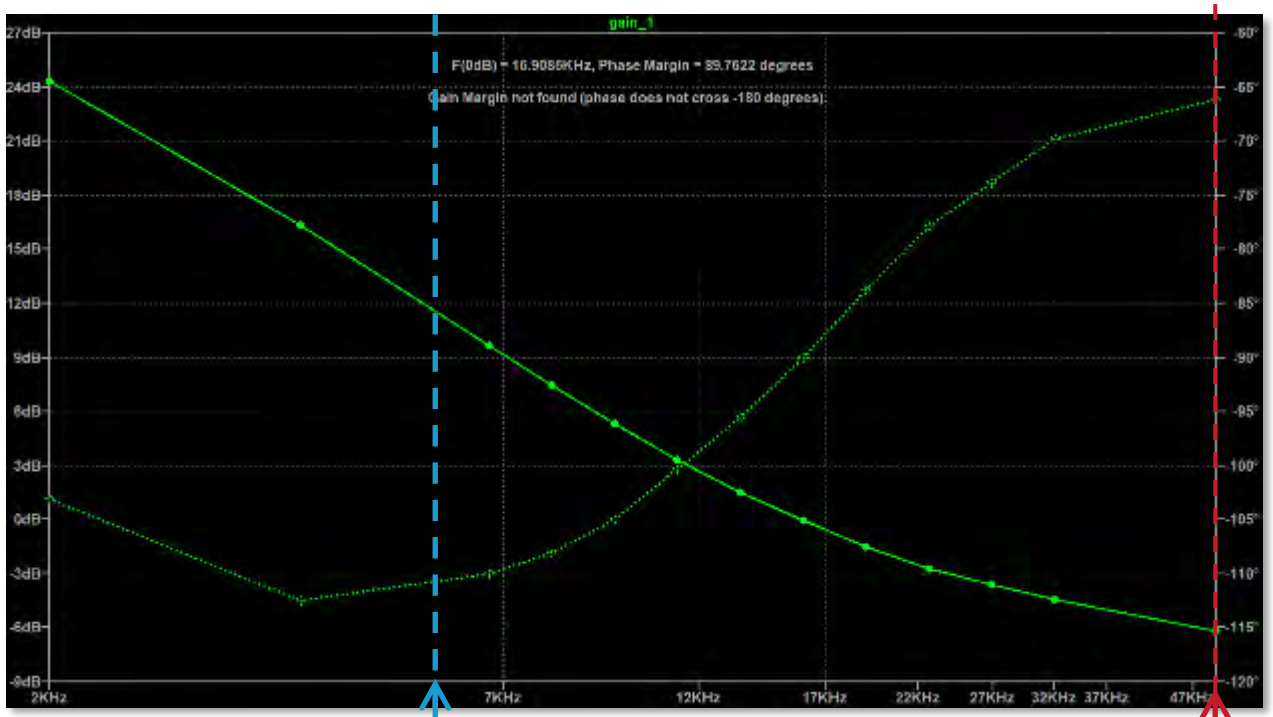


Example with 3 frequency points

FRA Device Settings: Optional Frequency Settings



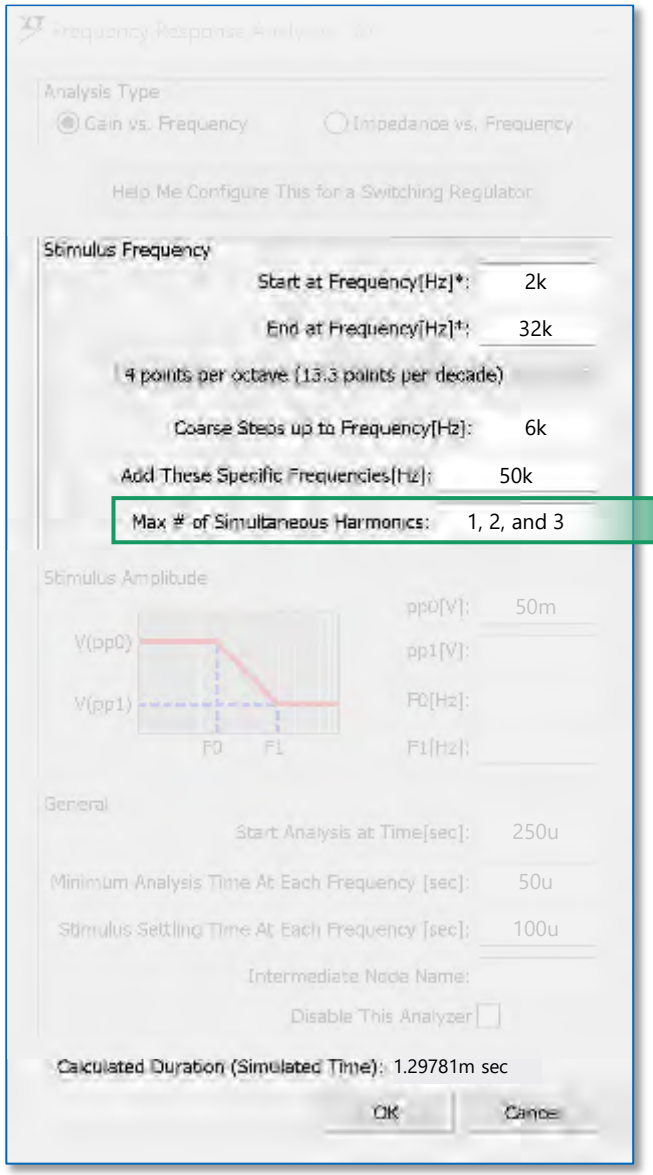
Increase Resolution
Reduce Low Freqs
Specific Freqs

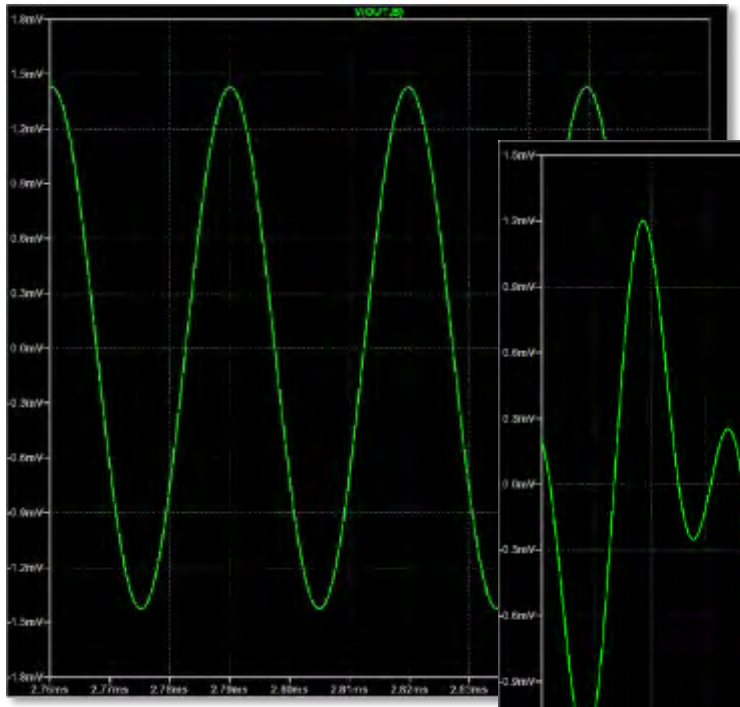


Limit resolution at low frequency (Fcoarse)
LTspice applies just one point per octave below this frequency, to speed up simulations. Recommend 2x~10x Fstart.

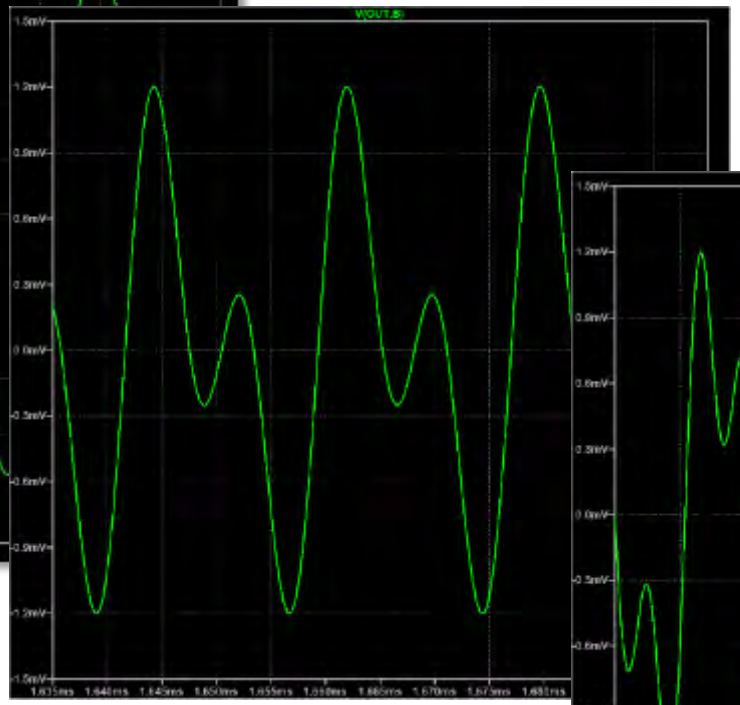
Specify individual frequencies
To be applied in addition to the sweep specified above

FRA Device Settings: Simultaneous Harmonics

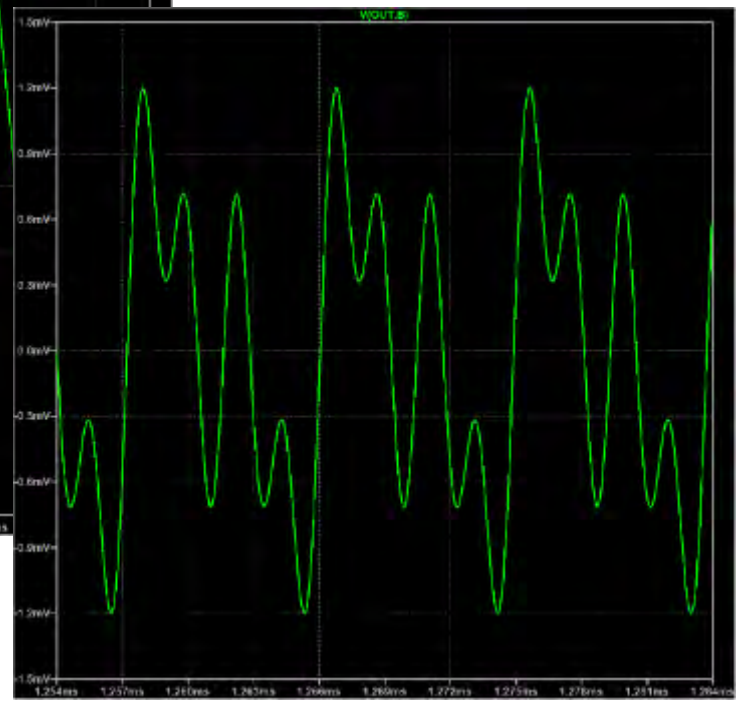




1 harmonic

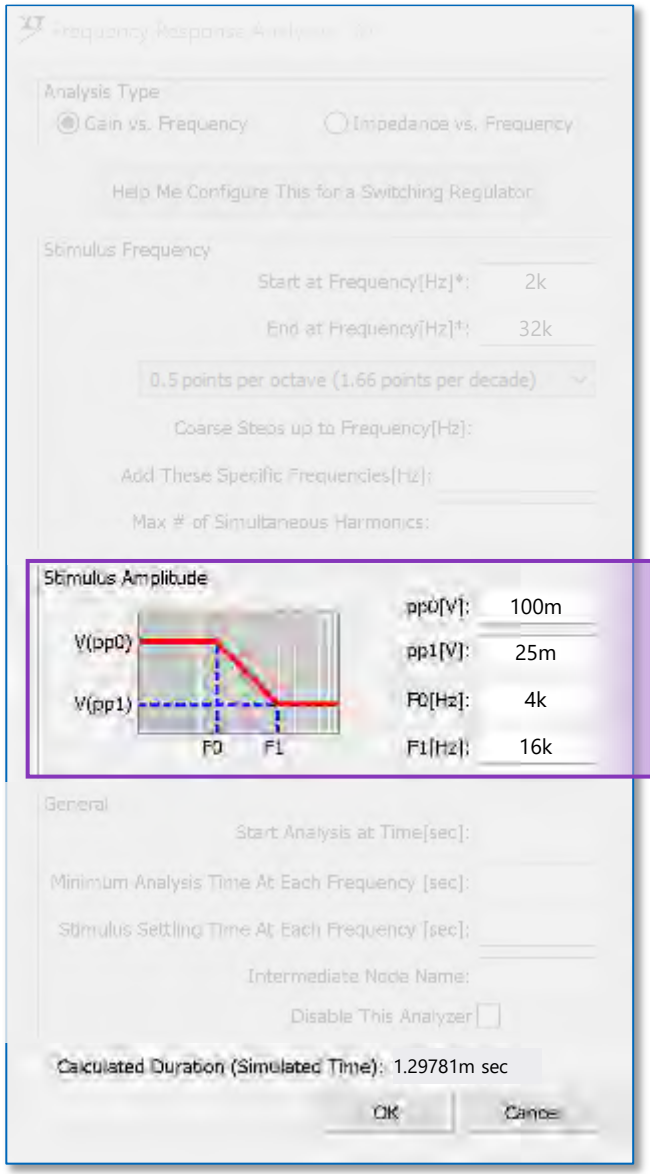


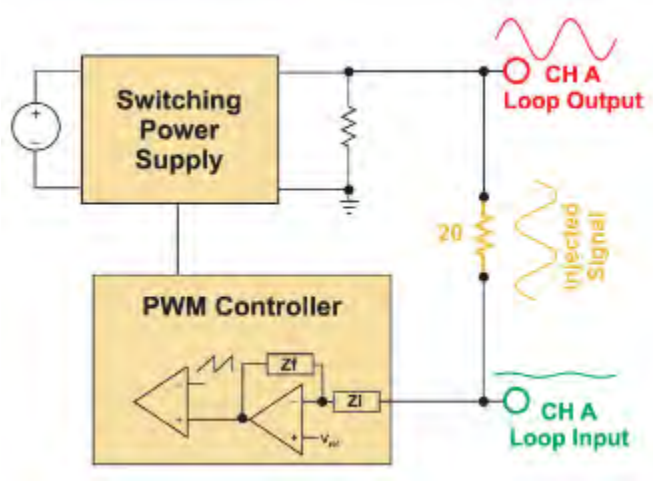
2 harmonics

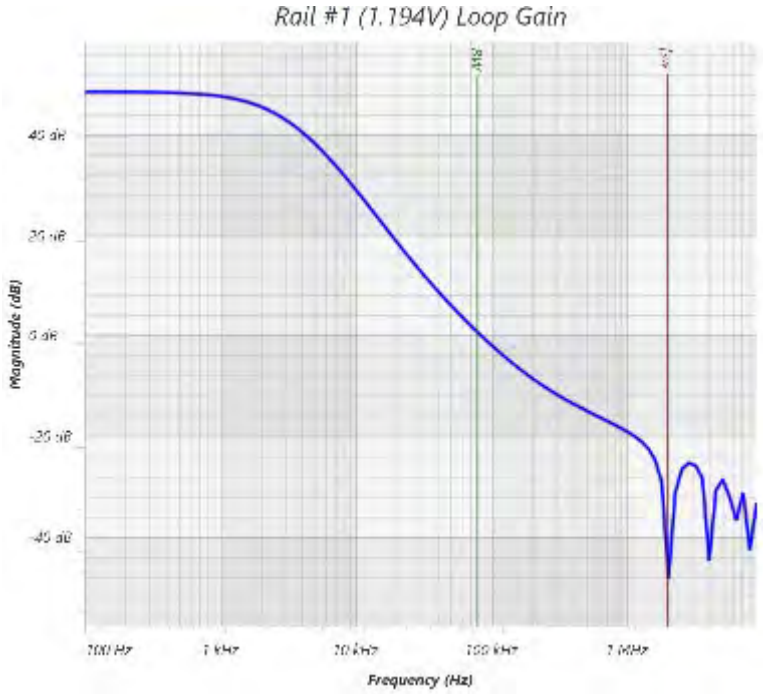


3 harmonics

FRA Device Settings: Stimulus Amplitude

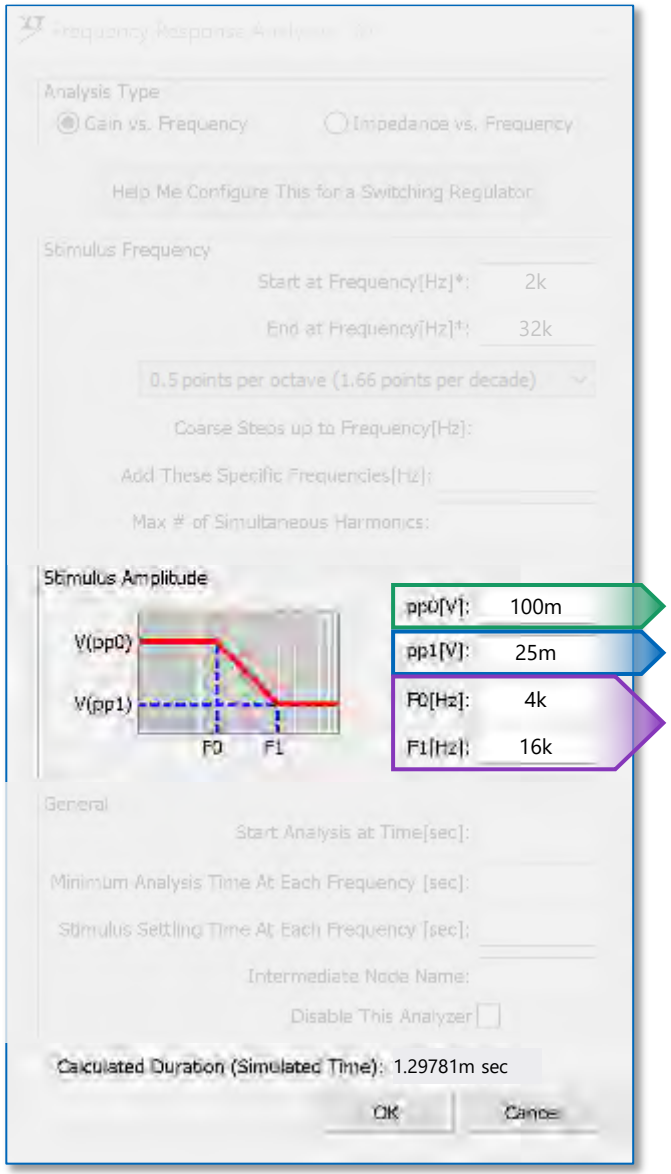


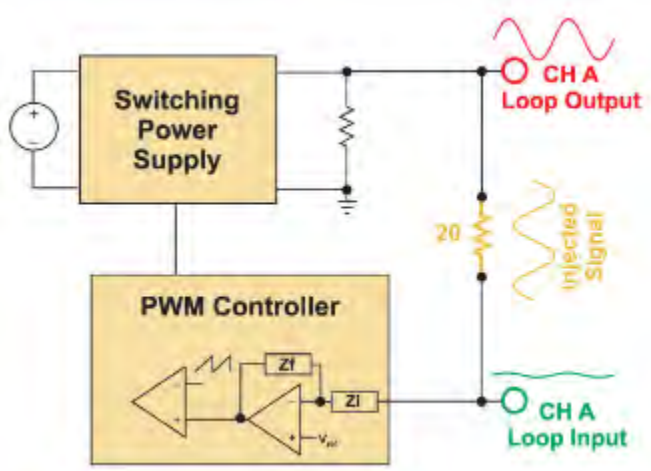




Signal needs to be large enough to resolve when divided by gain, yet small enough to avoid non-linear operation.

FRA Device Settings: Stimulus Amplitude

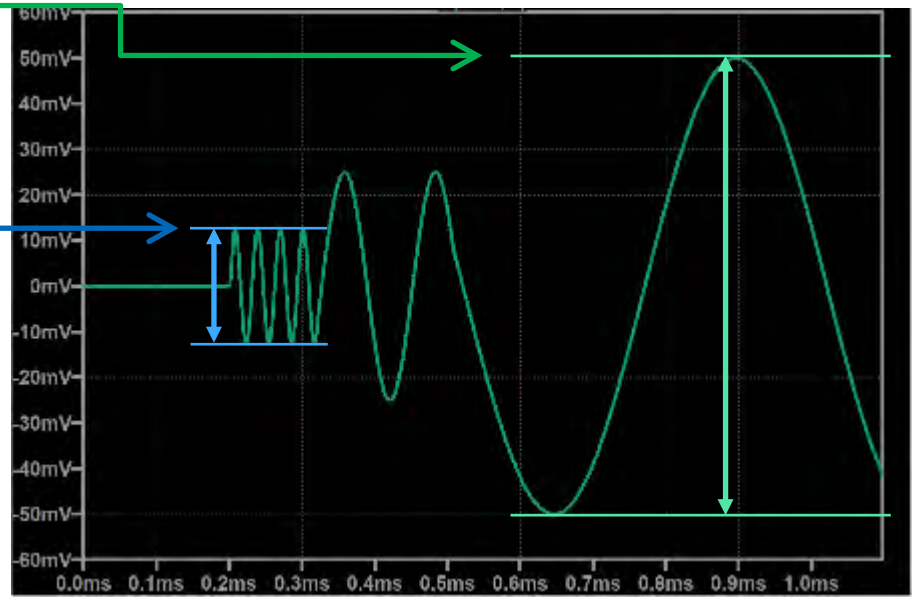




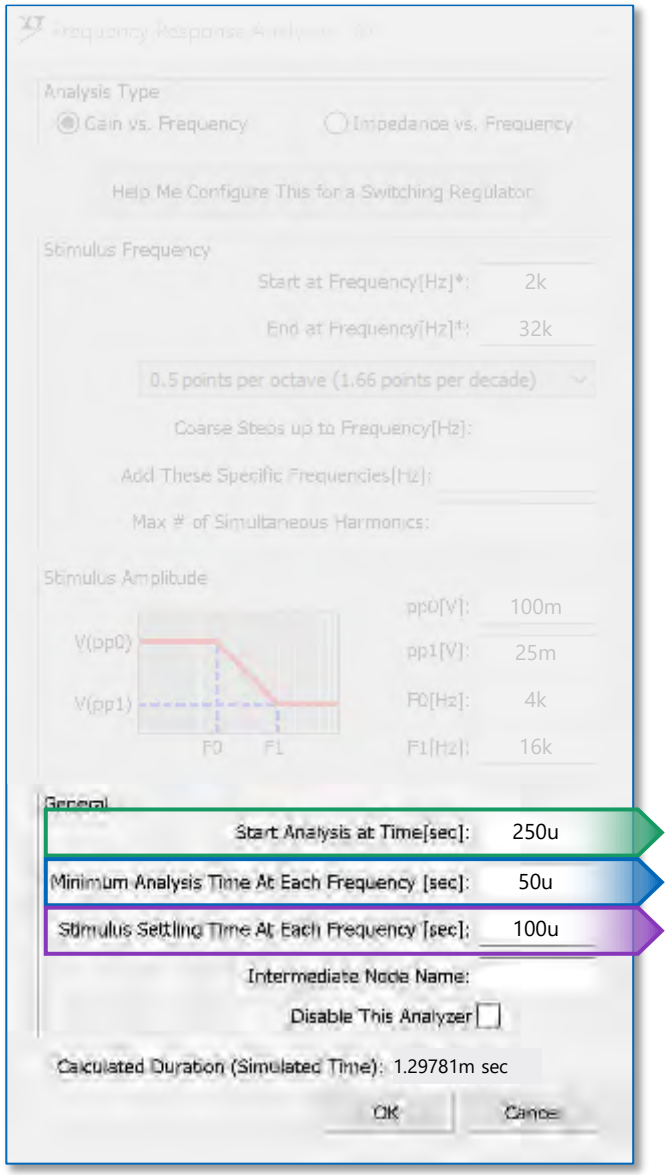
Amplitude at Low Freqs

Amplitude at High Freqs

Frequency Corners



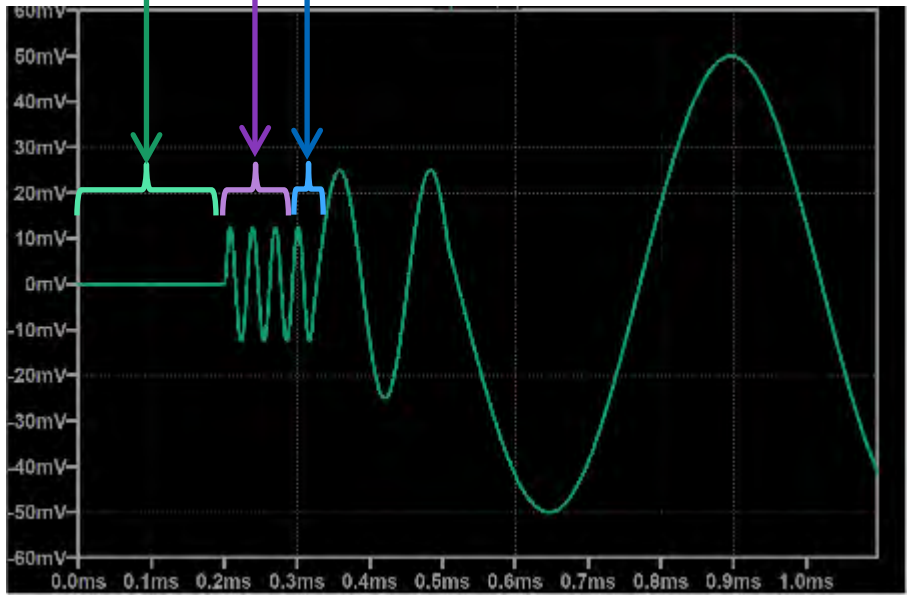
FRA Device Settings: General



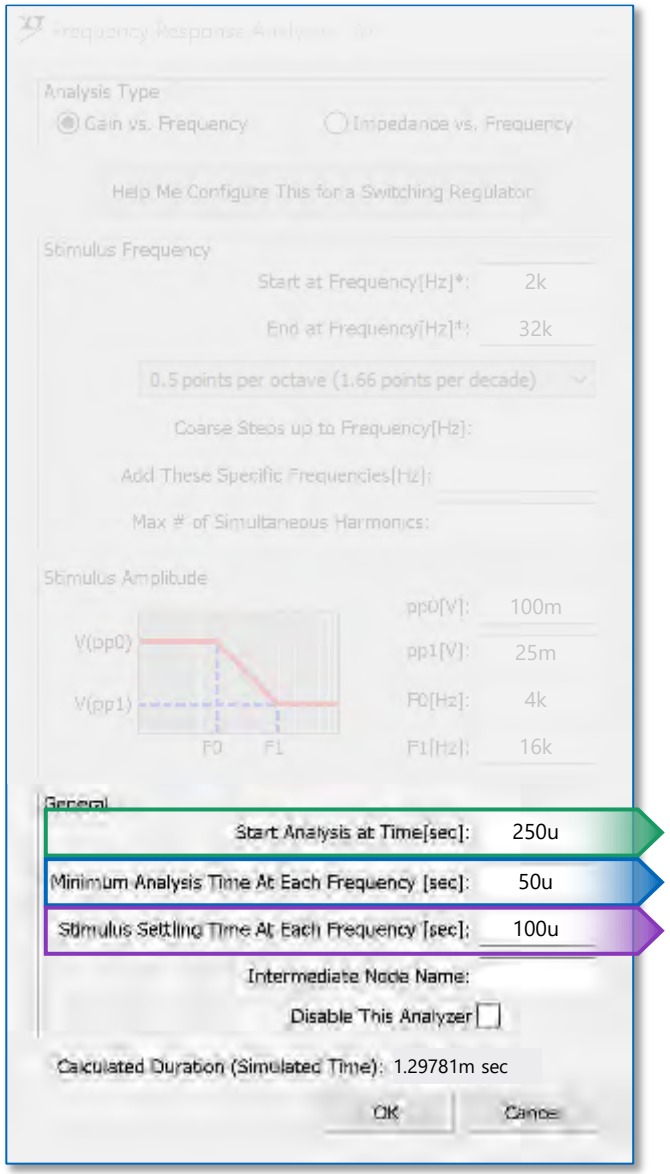
Averaging (Tavgmin)
Must be sufficient to average out switching noise
($100 \div f_{SW}$ is a good starting point)

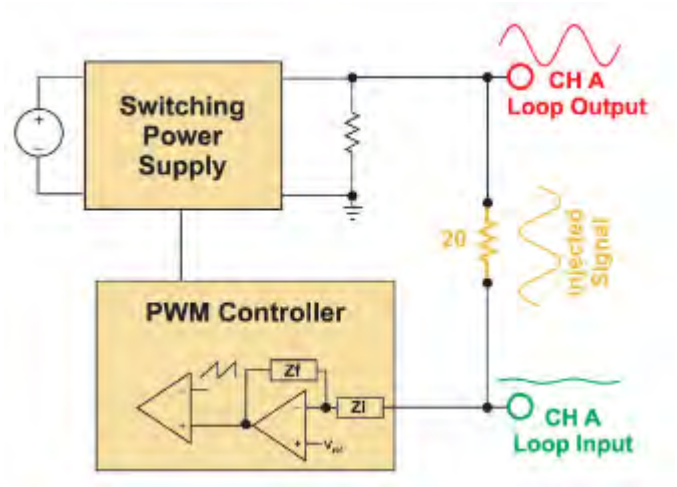
Settling (Tsettle)
Must be sufficient to allow system to adjust to new stimulus pattern, generally at least $2 \div f_{0dB}$

Start Analysis at Time Delay
Must be long enough to allow circuit to reach steady-state

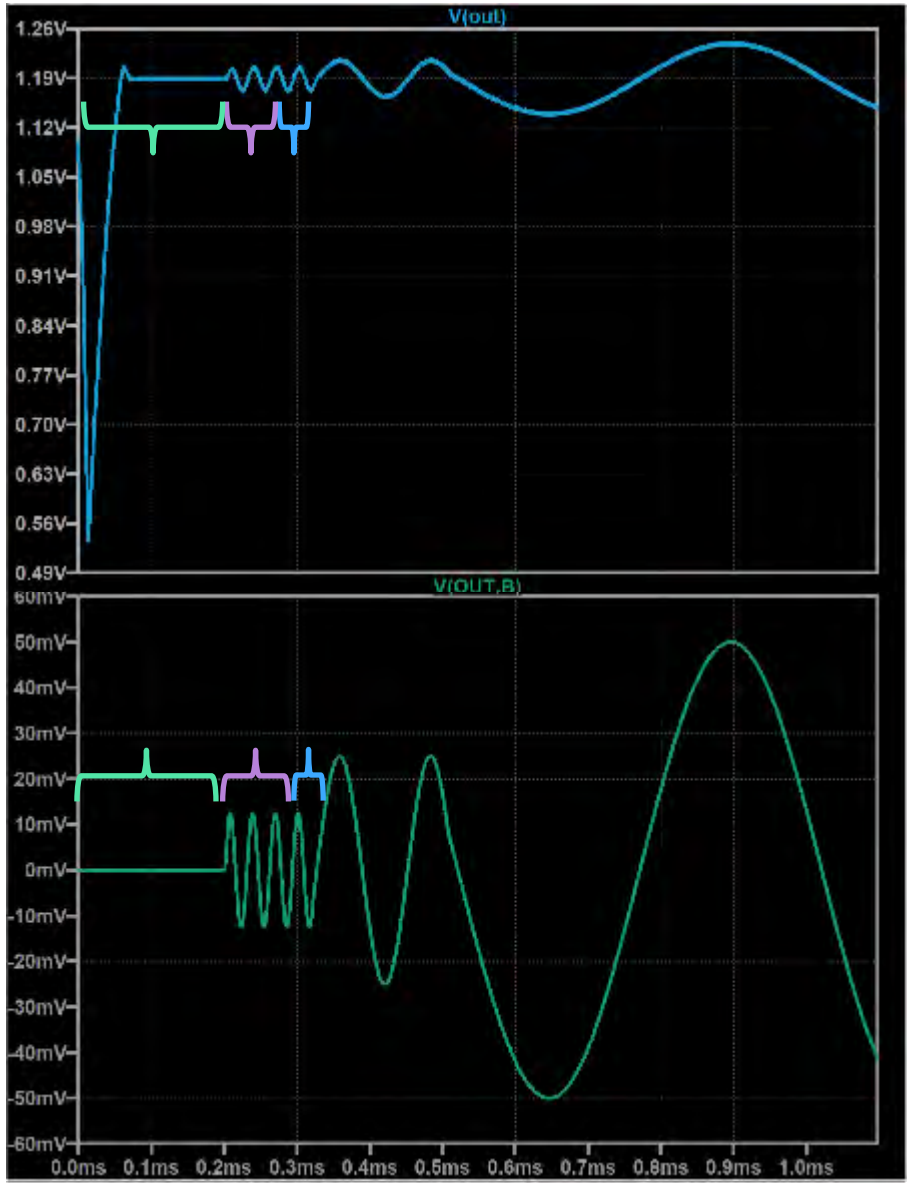


FRA Device Settings: General



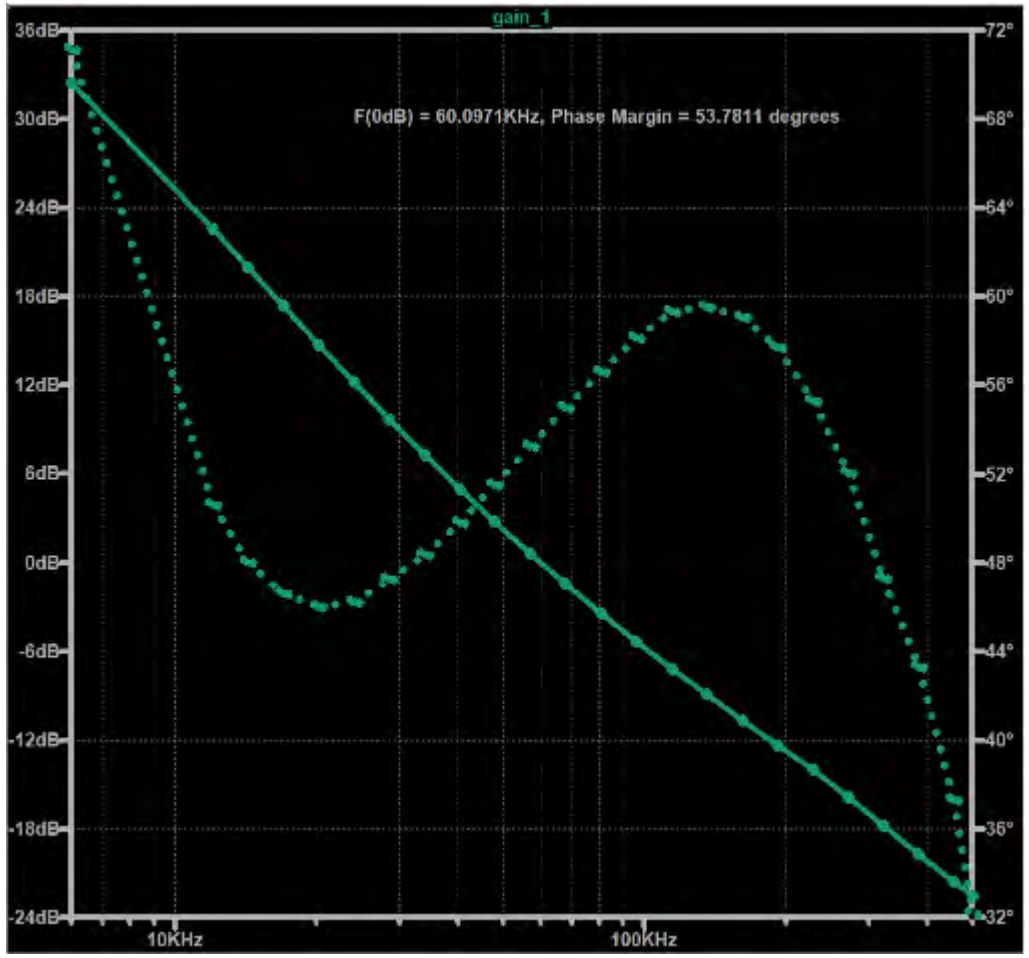


Inspect the resulting waveform in the transient simulation, and adjust these values accordingly

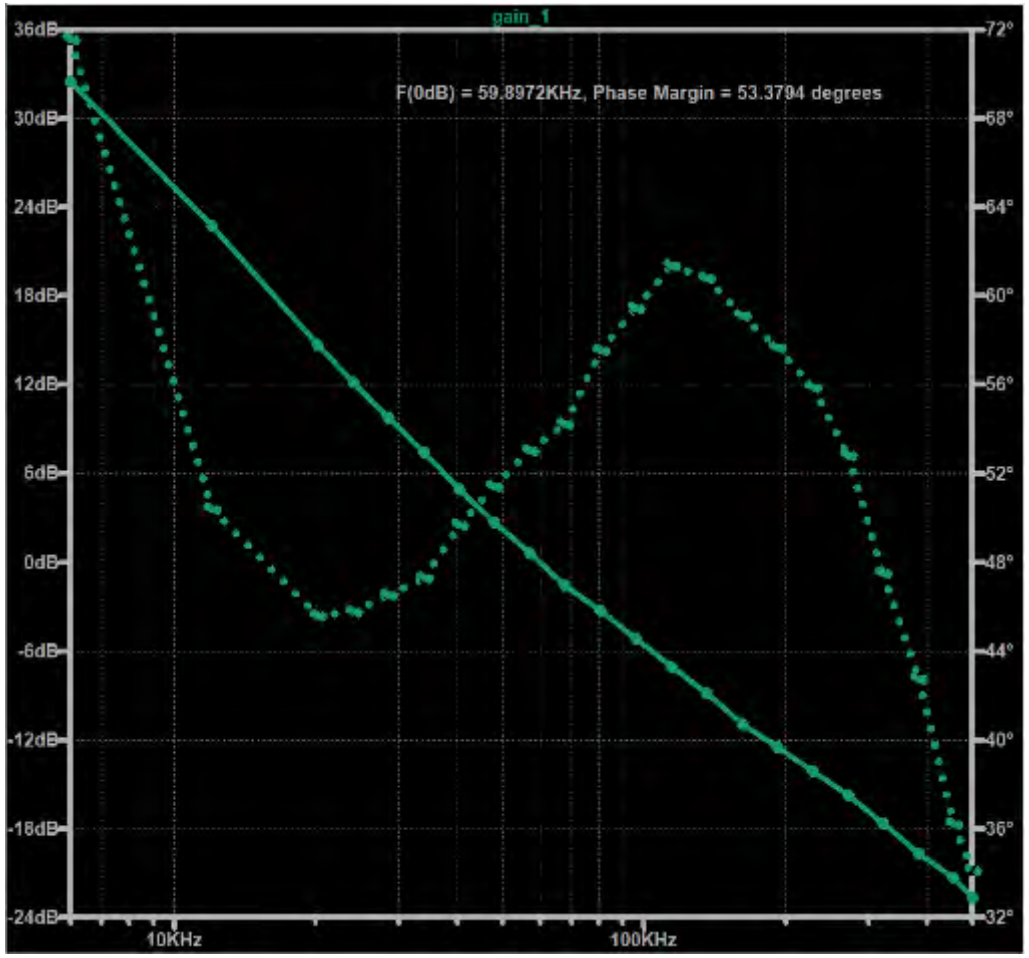


Presto!

Completed in 75 seconds

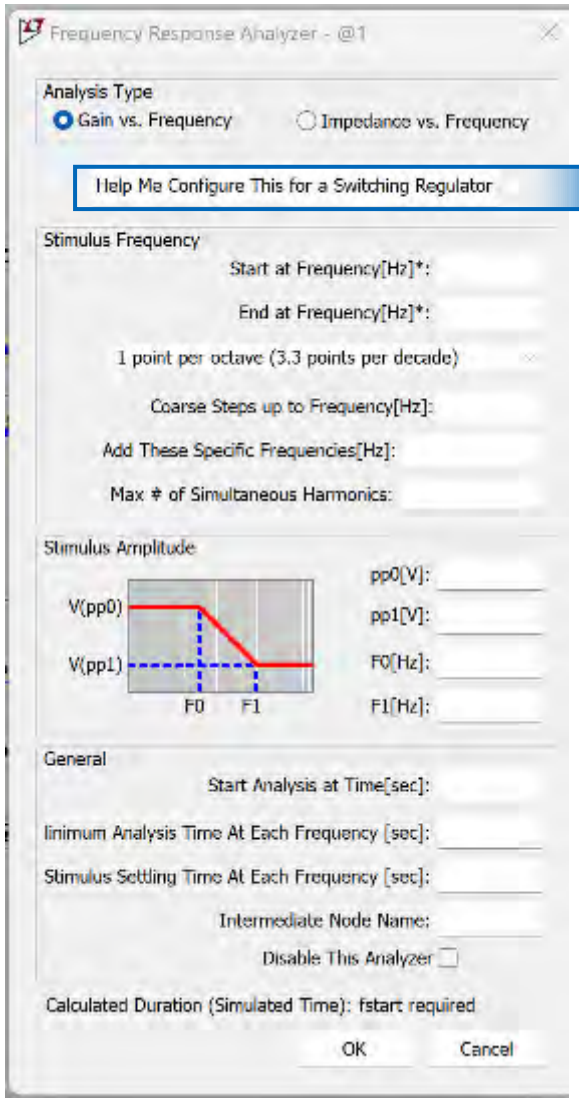


Completed in 25 seconds

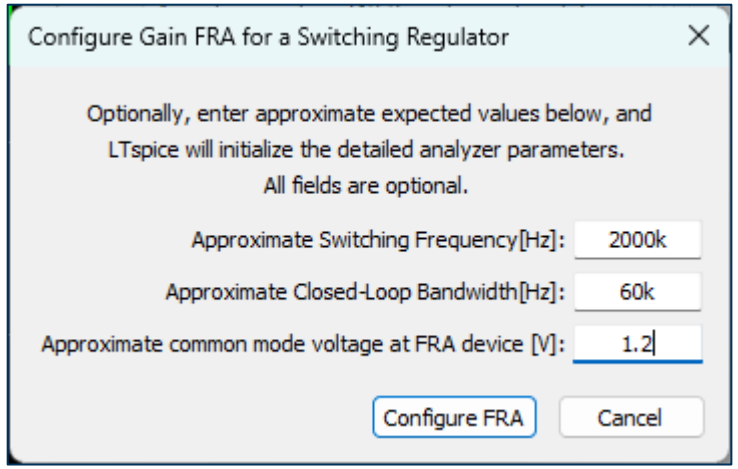


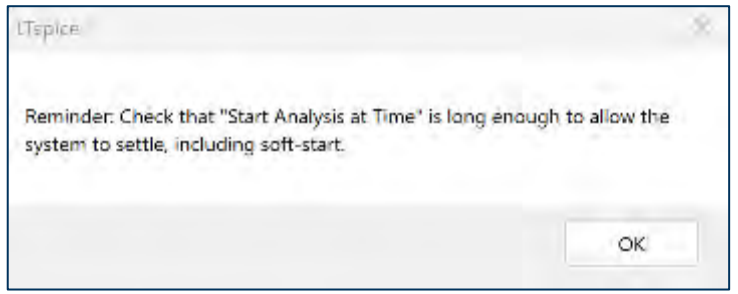
Config Wizard

FRA Device Settings: Configuration Wizard



The configuration wizard is used to quickly setup a FRA simulation

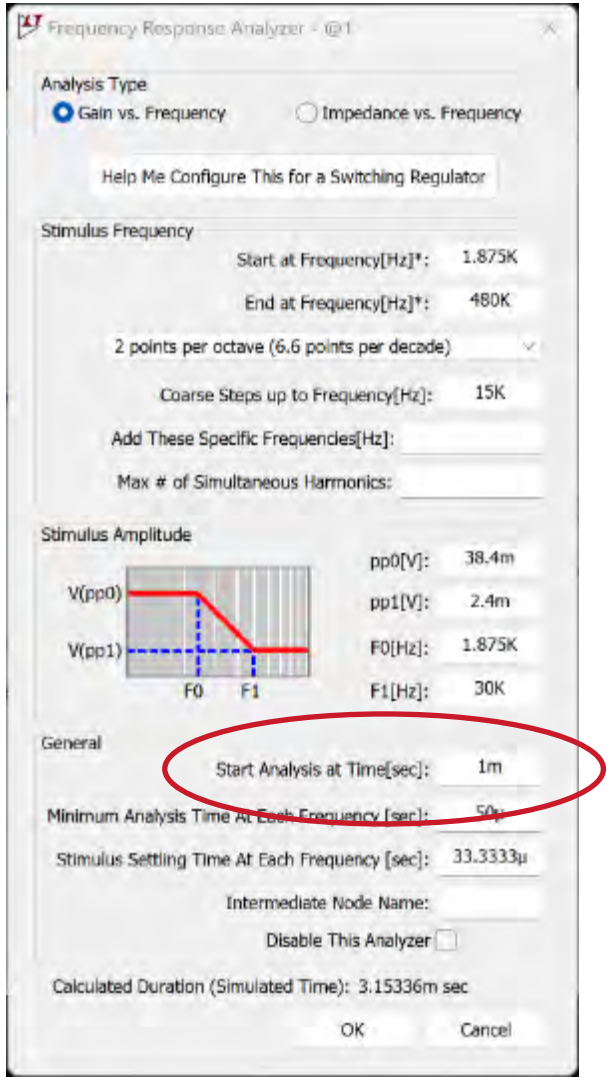


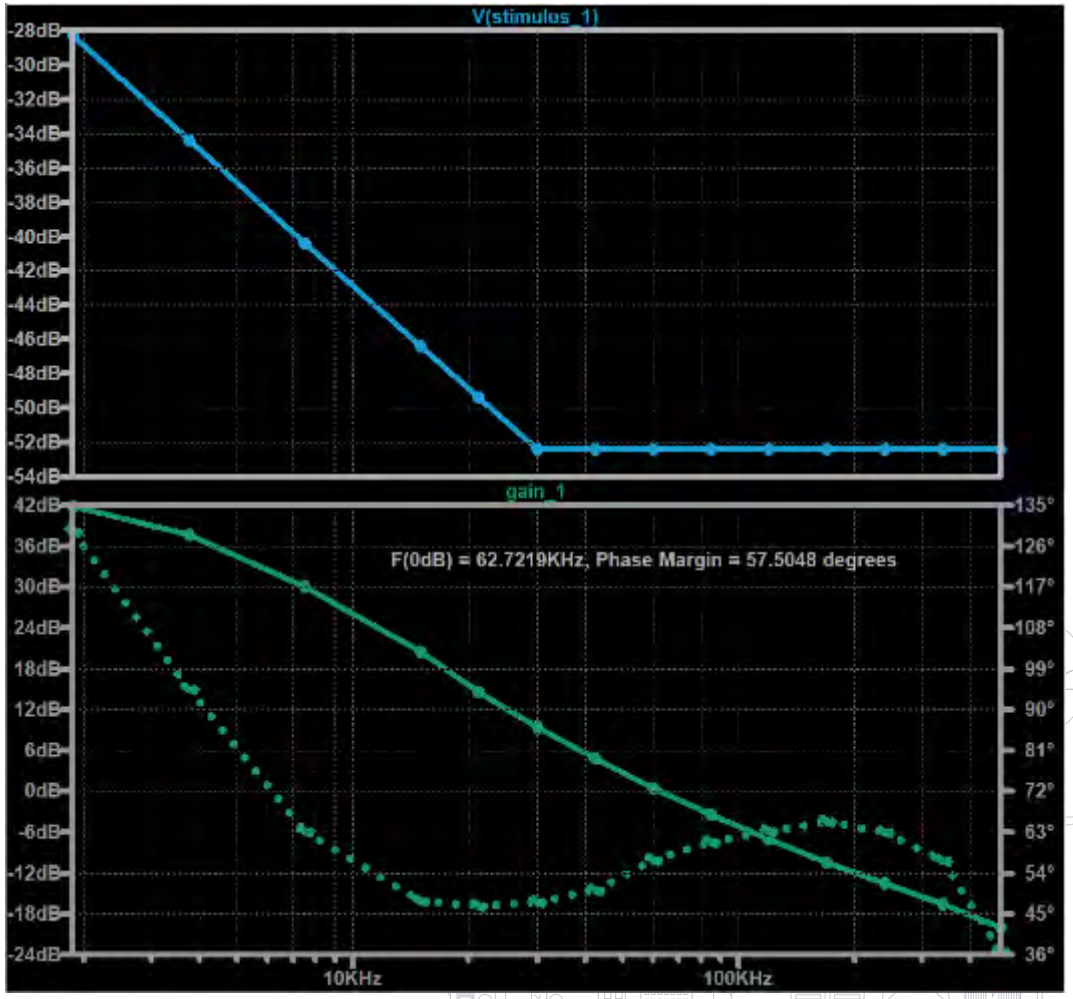


Only three things to enter

- Switching frequency
- Approximate Bandwidth
- Output voltage

FRA Device Settings: Configuration Wizard Results





ANALOG DEVICES | ADI Power

Power Supply Design using LTpowerCAD

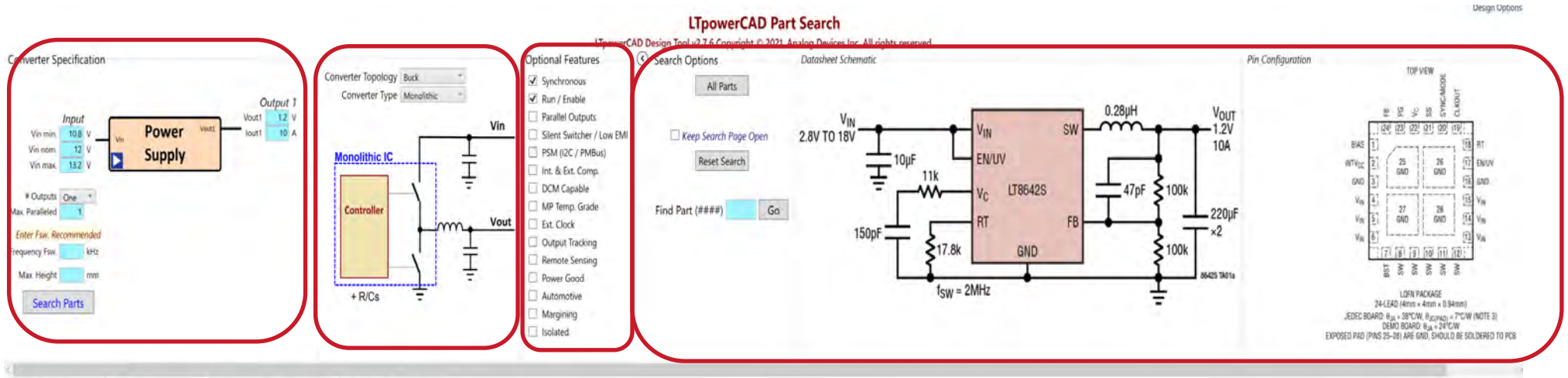
analog.com

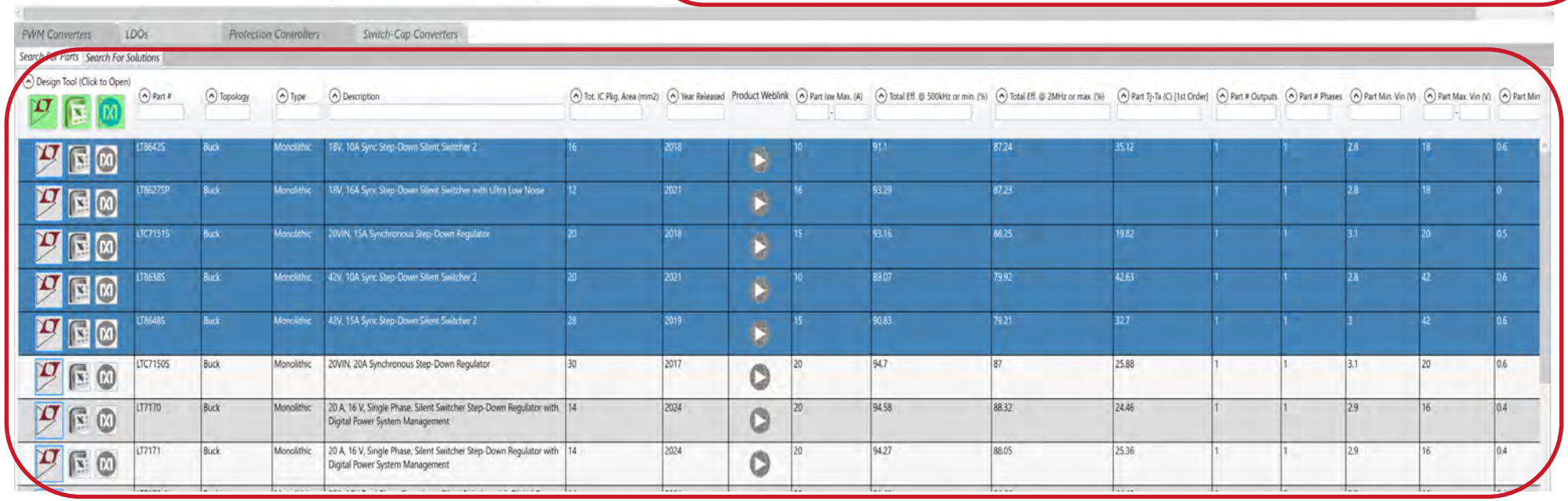
Homepage

Product Selection Page

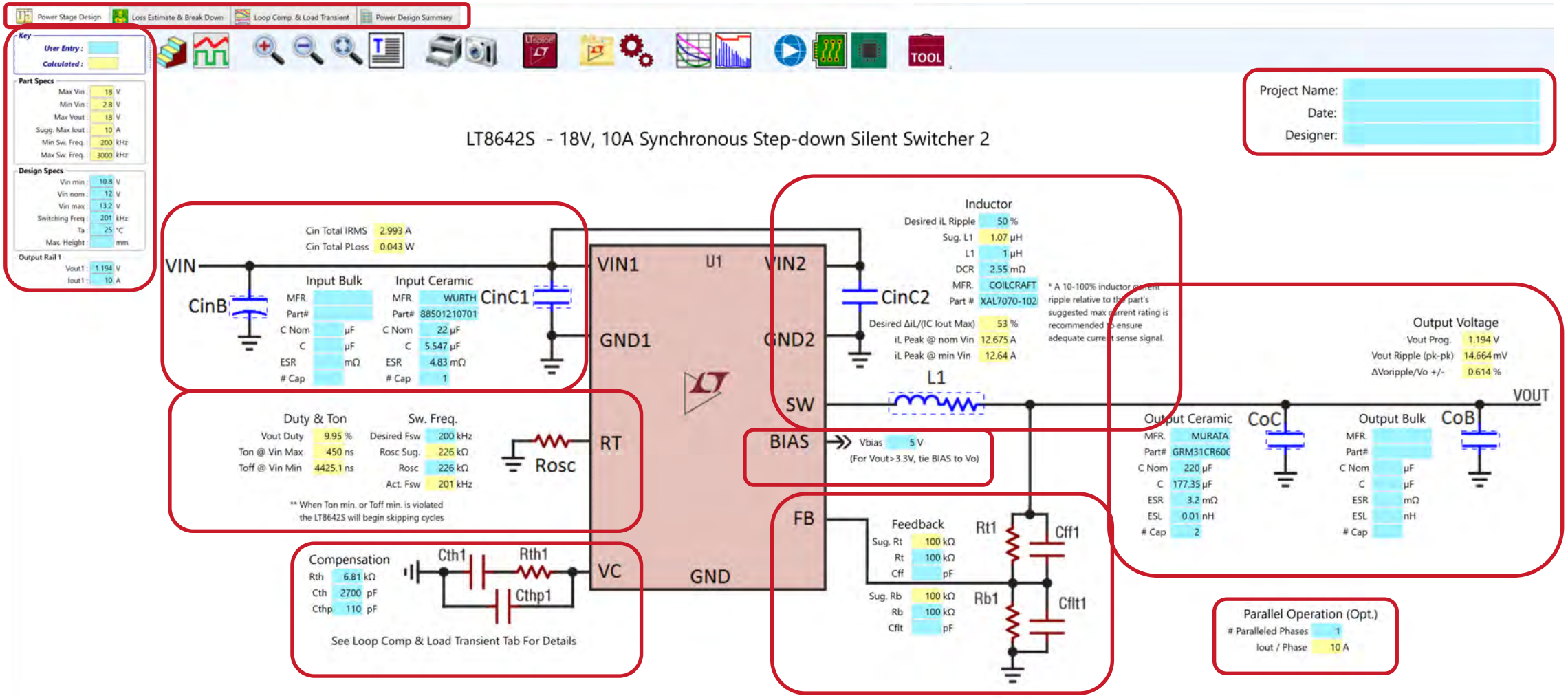
Monolithic: Top and Bottom MOSFET are integrated in the part.

uModule: Top and Bottom MOSFET, inductor, and CIN/COUT are integrated.

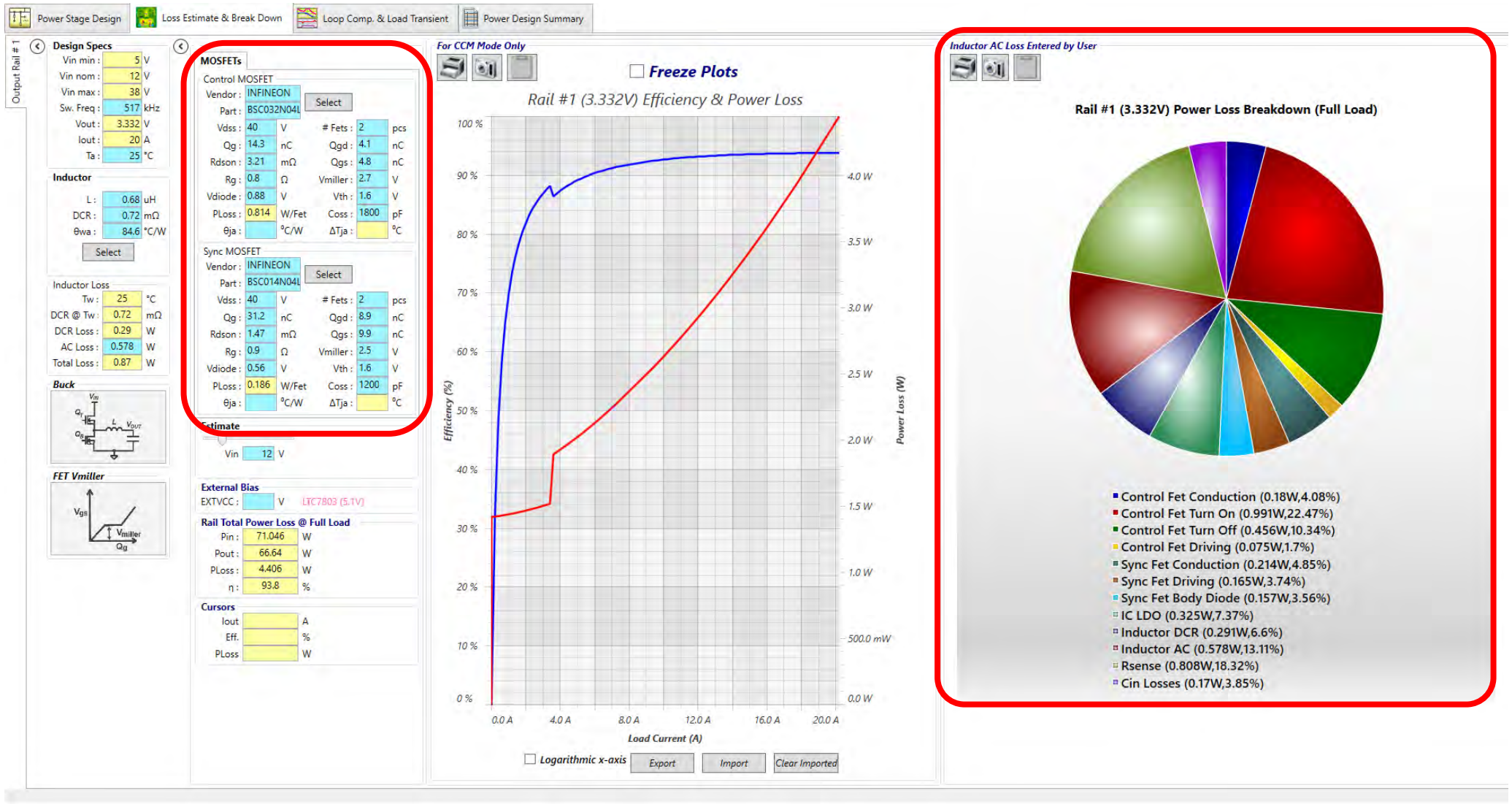




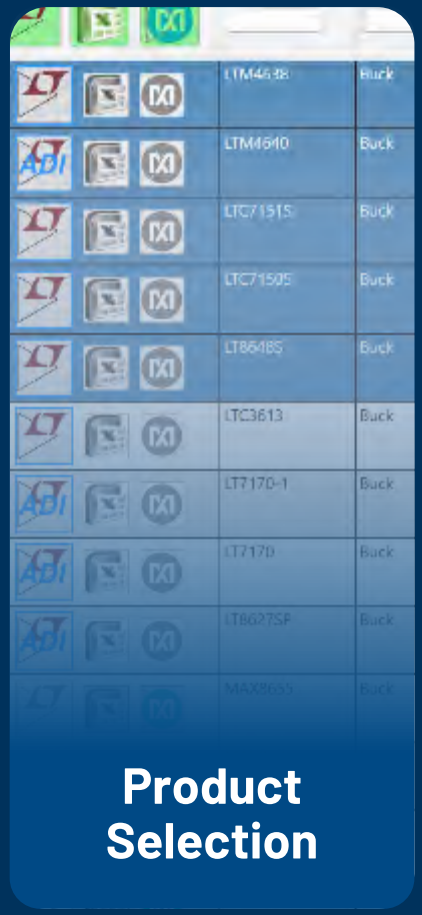
LTpowerCAD Power Stage Design Overview



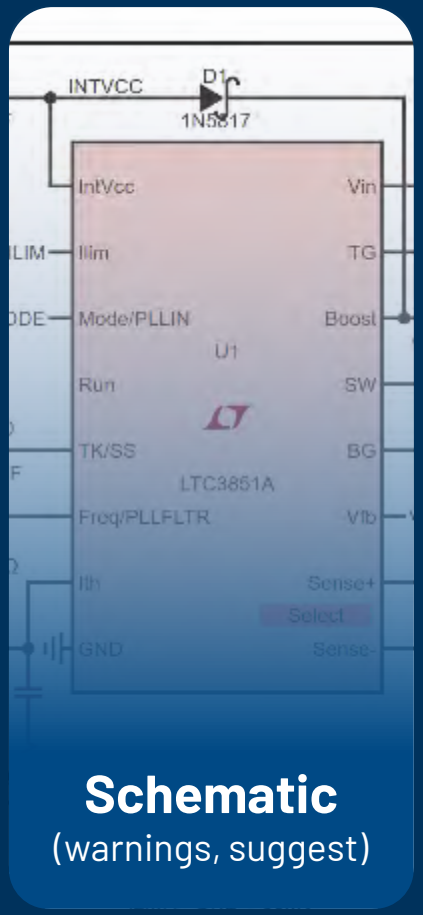
Loss Estimate & Break Down - Controller



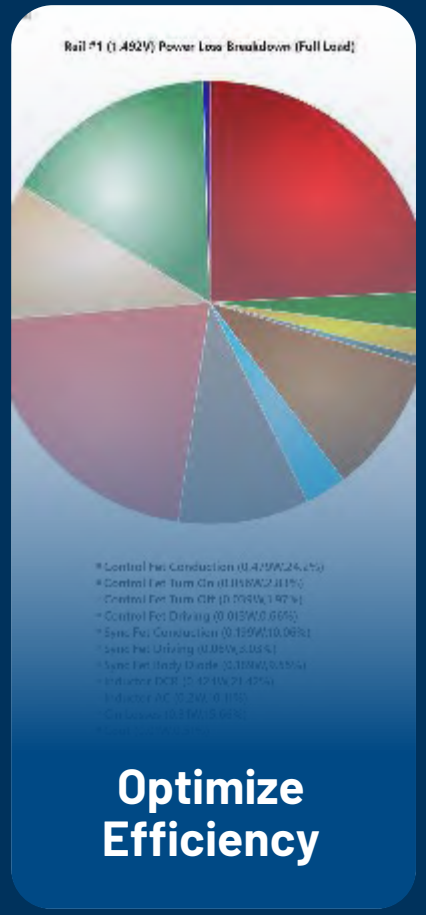
LTpowerCAD: Power Supply Design in 5 Easy Steps



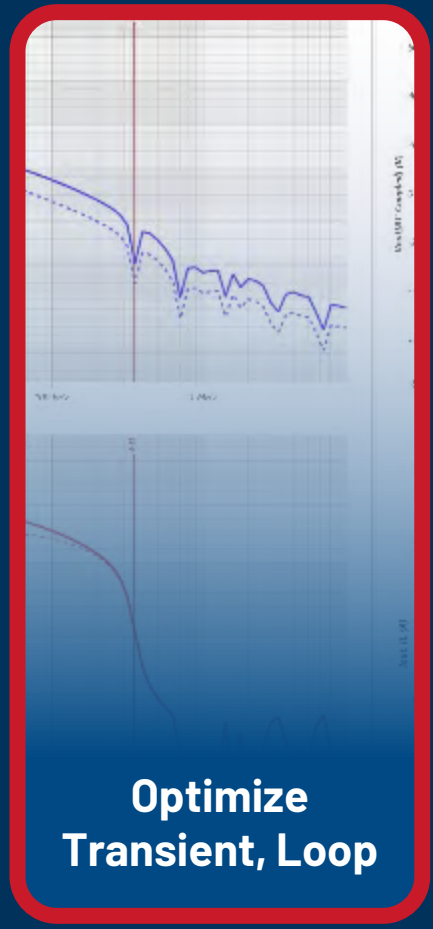
Product Selection



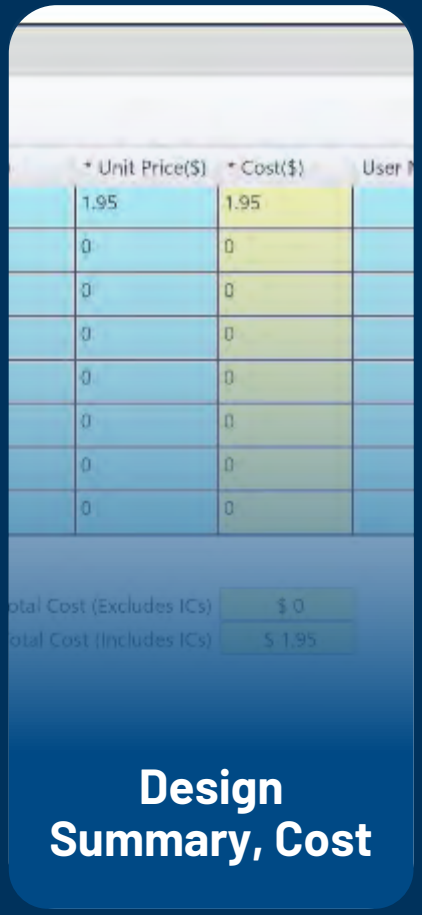
Schematic
(warnings, suggest)



Optimize Efficiency



Optimize Transient, Loop

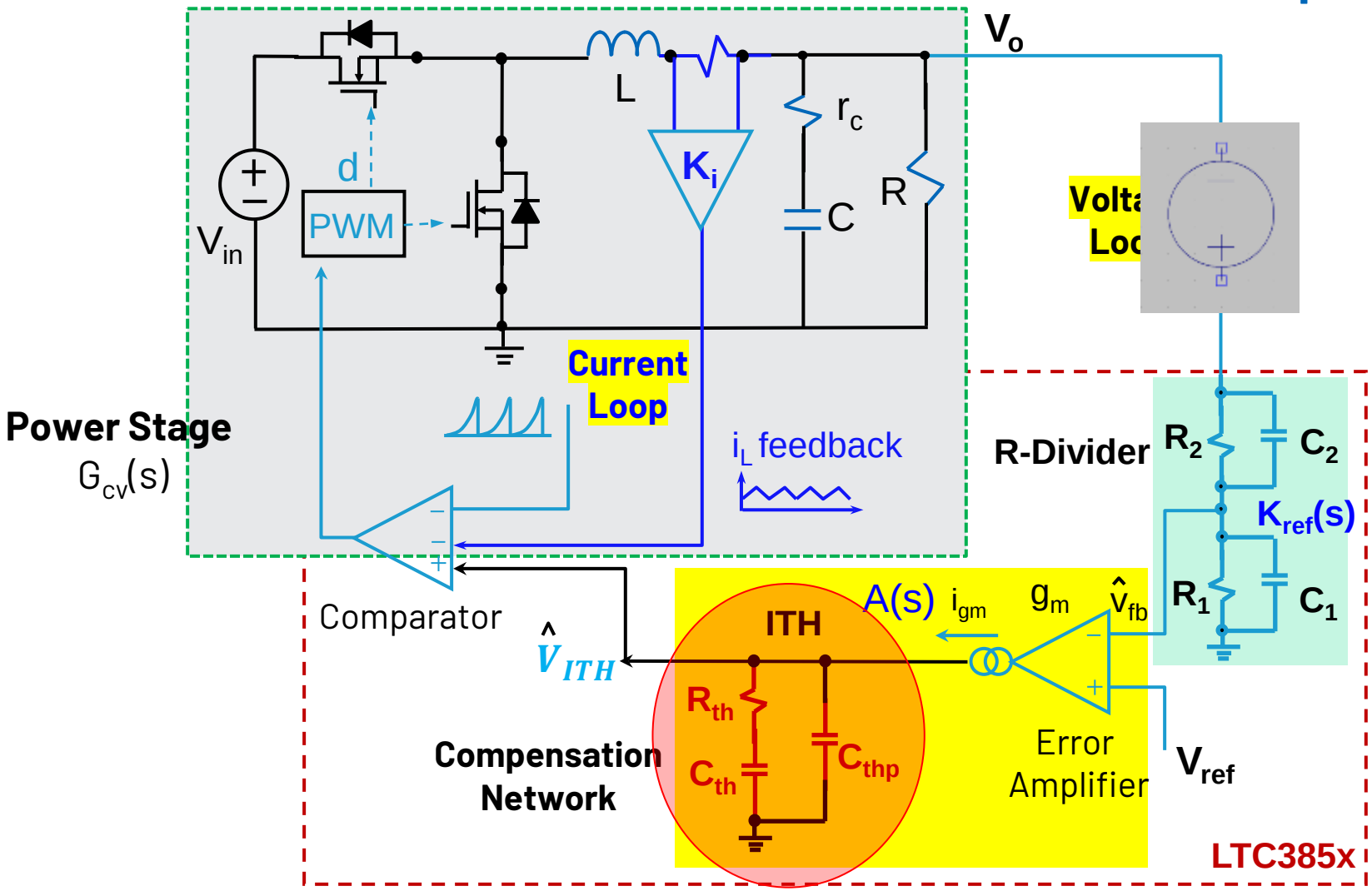


Design Summary, Cost

Free download @ www.analog.com/LTpowerCAD

LTpowerCAD Loop Comp & Load Transient Overview

Current Mode DC/DC Converter and its Loop Gain



Buck Regulator: Design of Compensation Network

- High loop DC Gain (by default).

Target #1) Fast Transient Response. (High BW)

→ **Loop BW = 10%~ 20% f_{SW}**

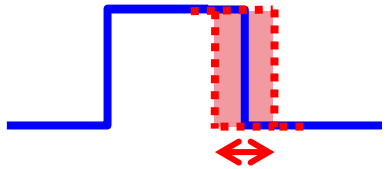
Target #2) Stability. (across operating range)

→ **PM ≥45 degree. (≥60 degree preferred.)**

→ **GM ≤ -10dB**

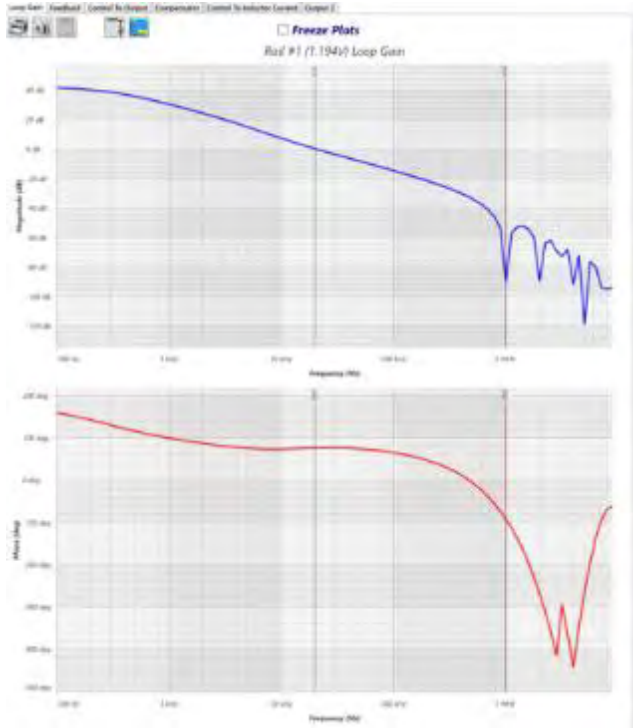
Target #3) Attenuate switching noises.

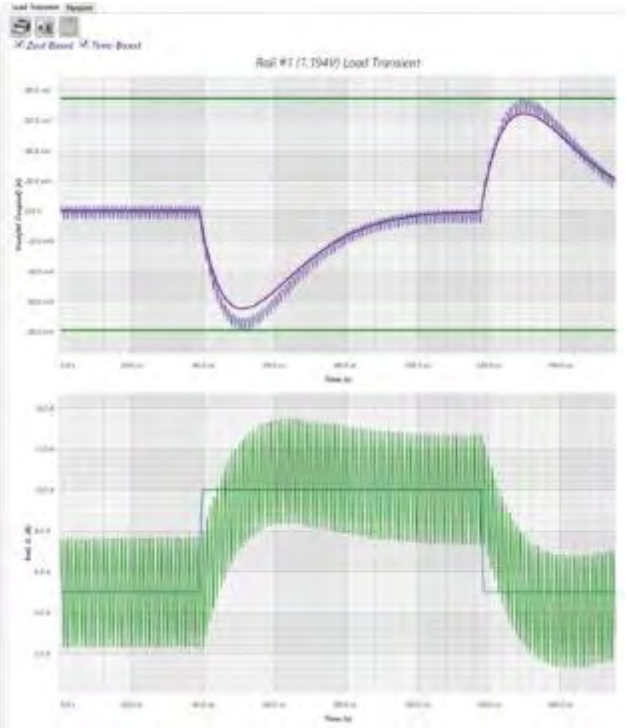
→ **Gain @ $f_{SW}/2 \leq -8dB$.**



Target #4) Meet transient Vpkpk spec.

→ **$\Delta V_{OUT} < \pm 3\%$.**



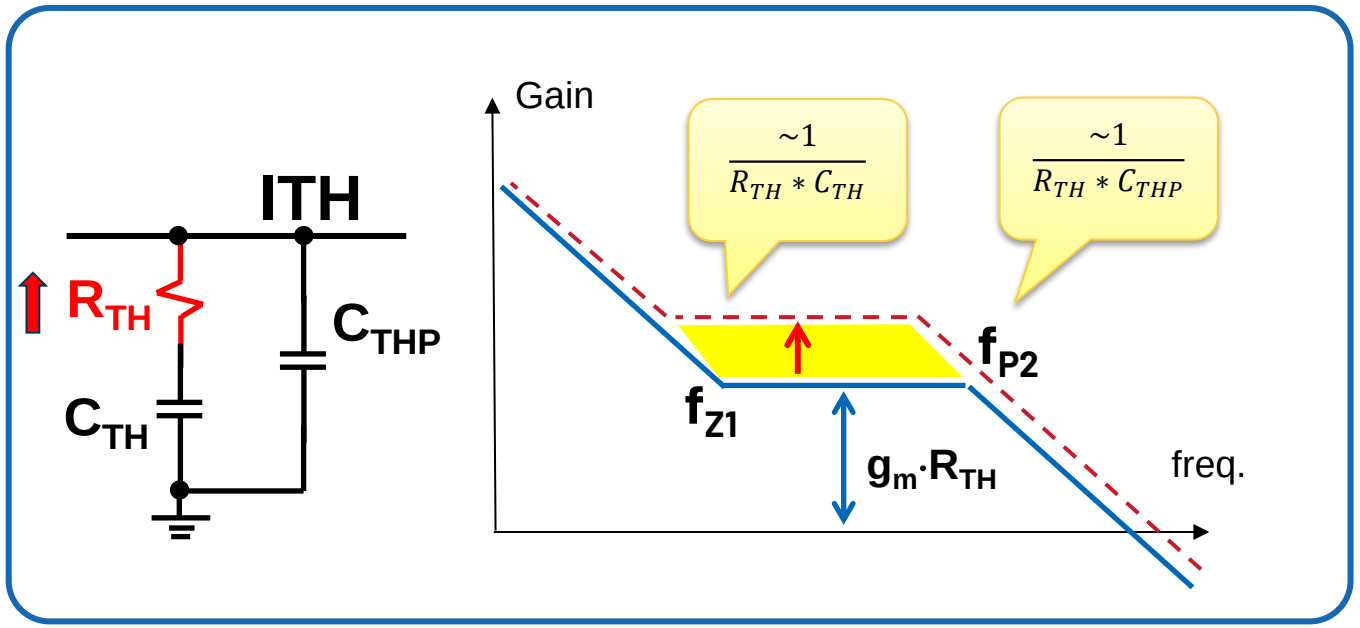


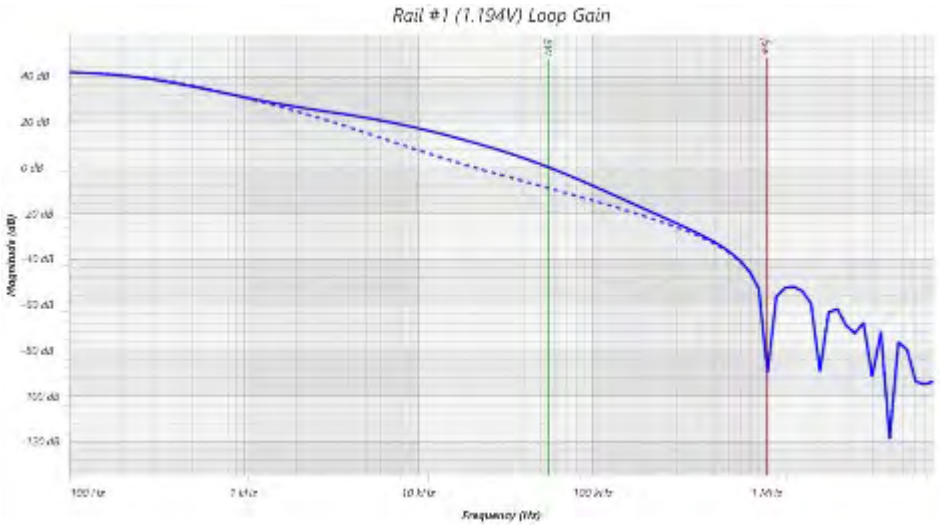
Component	Value	Units
R_{TH}	6.81	kΩ
C_{TH}	2700	pF
C_{THP}	110	pF
C_{FF}	none	pF

BW	19.95 kHz
PM	76.16°
GM	-33.44 dB
Gain at ½ Fsw	-33.44 dB
ΔV_{OUT}	+/- 6.04%

Initial values for LT8642S example
Fsw = 1MHz
Inductor = XEL4030-201MEB

Step 1: Set Loop Gain & Load Transient





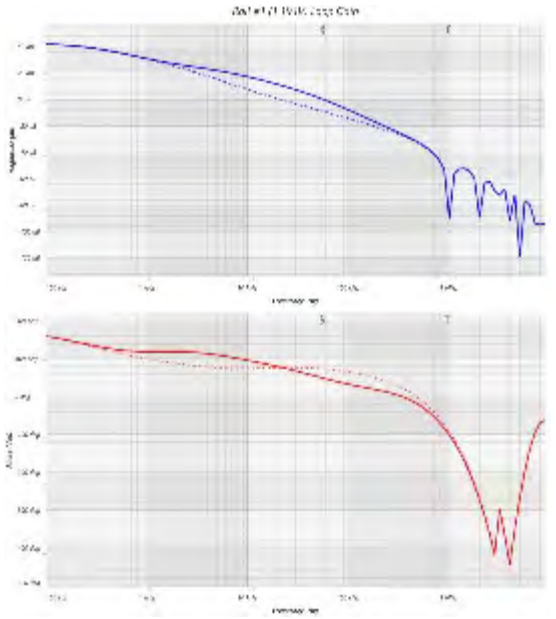
- Higher R_{TH} increases gain between Fz1 and Fp2
  - Reduces ΔV_{OUT} during load transients
  - Reduces phase margin at higher frequency

Step 1: Set Loop Gain & Load Transient

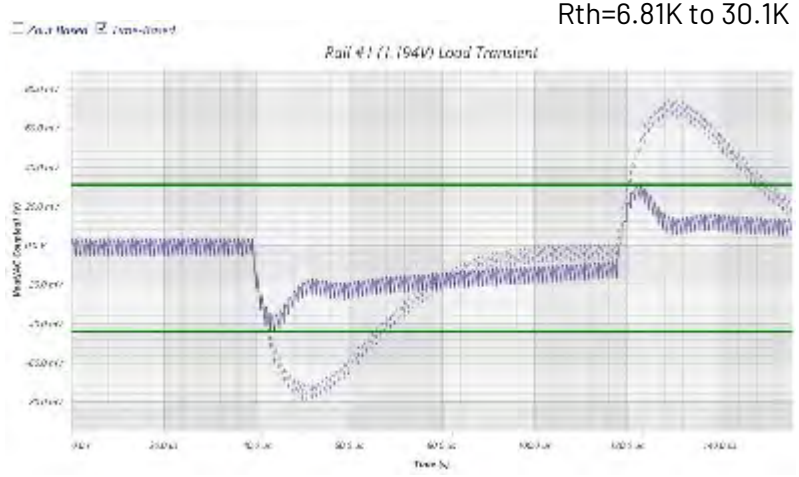
Rth = 6.81K to 30.1K

CTH = 2700pF and CTHP = 110pF

Stability Metric	R_{TH} = 6.81kΩ	R_{TH} = 10kΩ	R_{TH} = 15kΩ	R_{TH} = 20kΩ	R_{TH} = 24.9kΩ	R_{TH} = 30.1kΩ	R_{TH} = 35.7kΩ	Effects of increasing Rth
Bandwidth (kHz)	**19.95**	**28.18**	**39.81**	**50.12**	**50.12**	**56.23**	**56.23**	**Increases**
Phase Margin (Deg)	76.16	78.57	71.5	61.22	56.53	48.7	44.8	**Decreases Margin**
Gain @ fSW/2 (dB)	-33.46	-33.08	-32.88	-32.81	-32.78	-32.76	-32.74	**Decreases Margin**
Gain Margin (dB)	-33.46	-31.05	-28.91	-28.8	-26.92	-26.88	-26.86	**Decreases Margin**
ΔV_{OUT} (%)	7.01	5.86	4.88	4.46	4.28	4.18	4.14	**Improves**

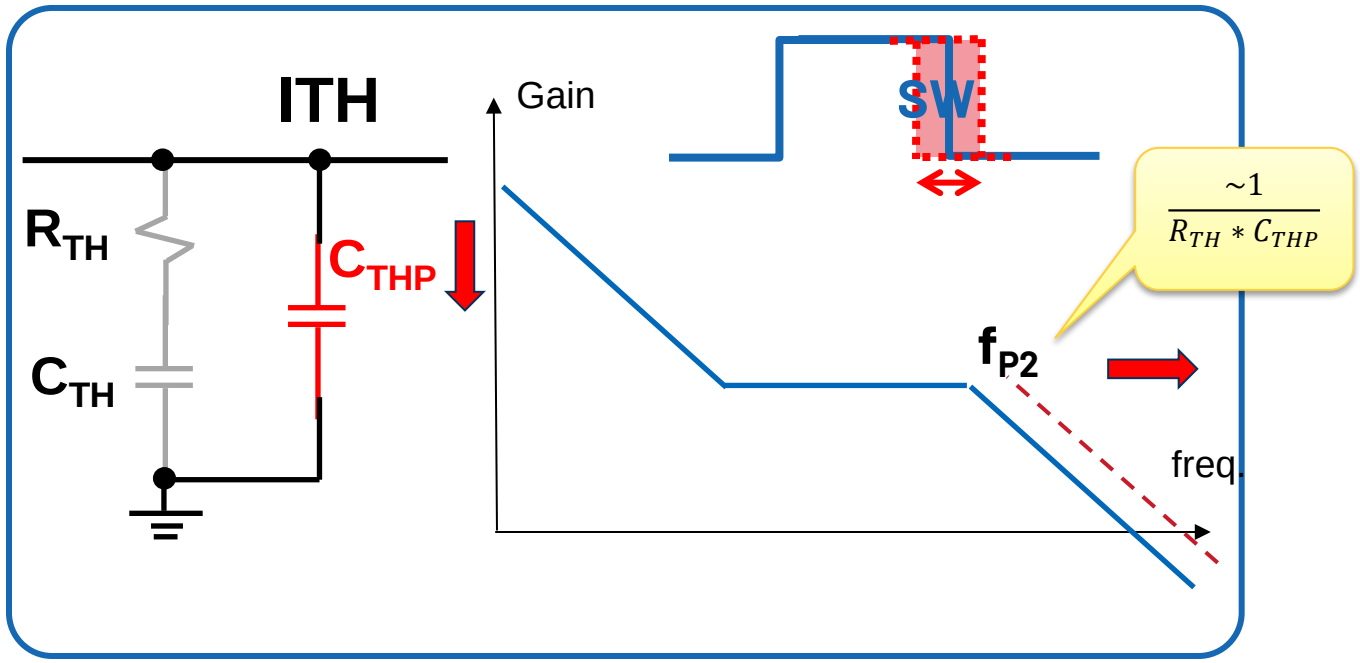


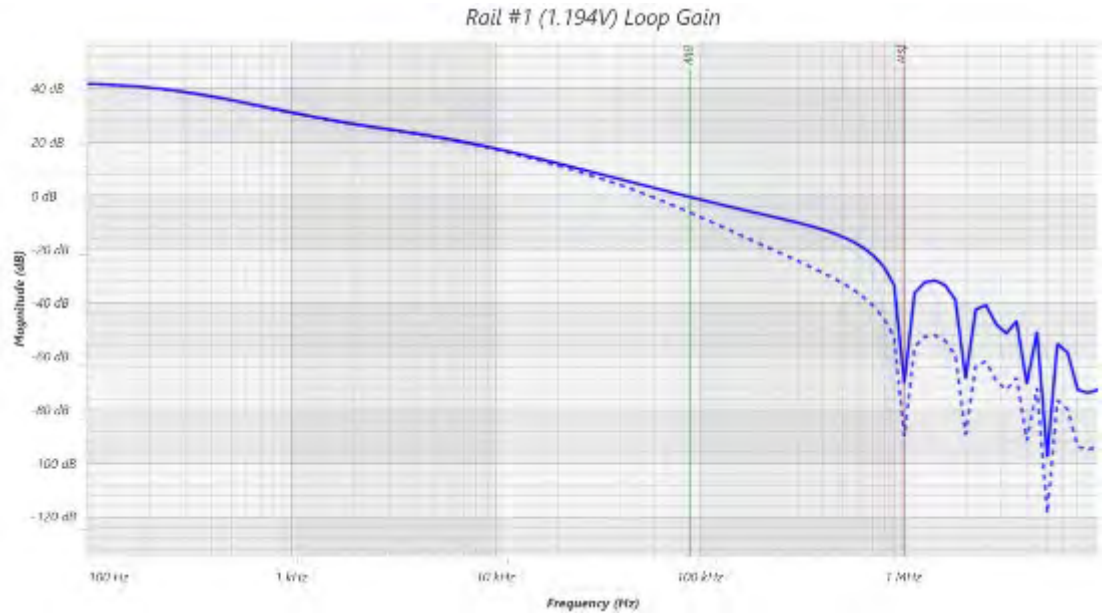
Rth=6.81K to 30.1K



- ❑ Loop BW = 10%~ 20% f_{SW}
- ❑ ΔV_{OUT} < ±3%
- ❑ PM > 60°
- ✓ **GM < -10dB**
- ✓ **Gain @ f_{SW}/2 < -8dB**

Step 2: HF Gain Attenuation (C_{THP})





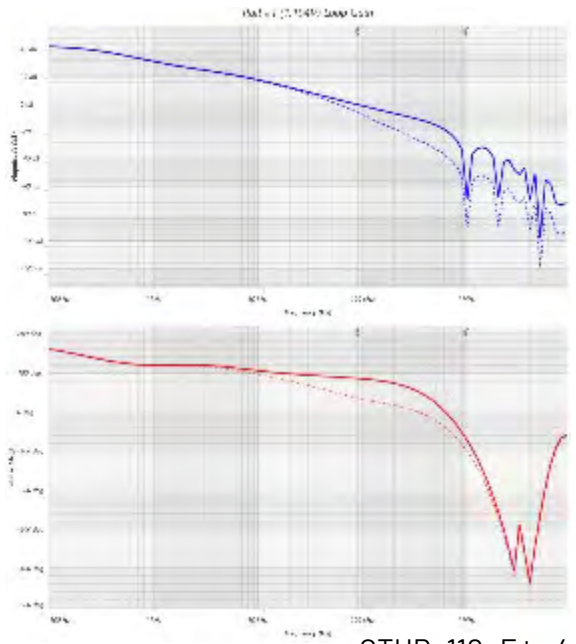
- Higher C_{THP} reduces HF gain, improve HF noise immunity
- Lower C_{THP} may improve transient ΔV_{OUT}
  - f_{P2} moves to a higher frequency and will affect PM and GM

Step 2: HF Gain Attenuation (C_{THP}) cont.

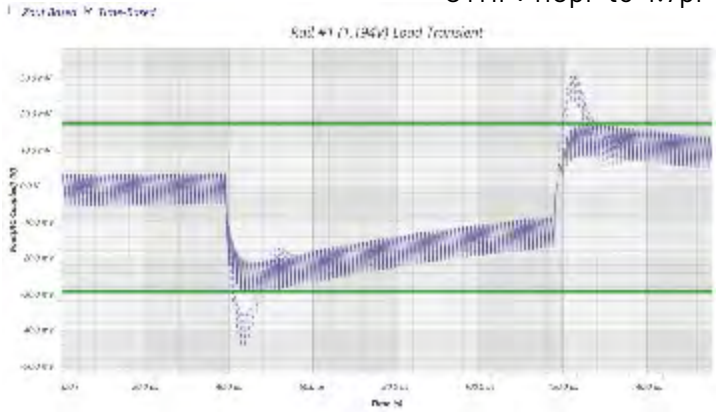
CTHP: 110pF to 4.7pF

RTH = 30.1K and CTH = 2700pF

Stability Metric	CTHP = 110pF	CTHP = 100pF	CTHP = 47pF	CTHP = 22pF	CTHP = 10pF	CTHP = 4.7pF	Effects decreasing CTHP
Bandwidth (kHz)	56.3	56.23	70.79	79.43	79.43	89.13	**Increases**
Phase Margin (Deg)	48.7	51.22	63.3	74.64	82.87	85.69	**Increases Margin at higher bandwidth**
Gain @ fSW/2 (dB)	-32.76	-31.98	-26.07	-20.99	-17.3	-15.42	**Decreases Margin**
Gain Margin (dB)	-26.88	-26.11	-24.01	-20.99	-19.31	-19.48	**Decreases Margin**
ΔV_{OUT} (%)	4.18	4.07	3.36	2.99	2.94	2.91	**Improves**



CTHP: 110pF to 4.7pF



❑ Loop BW = 10%~ 20% f_{SW}

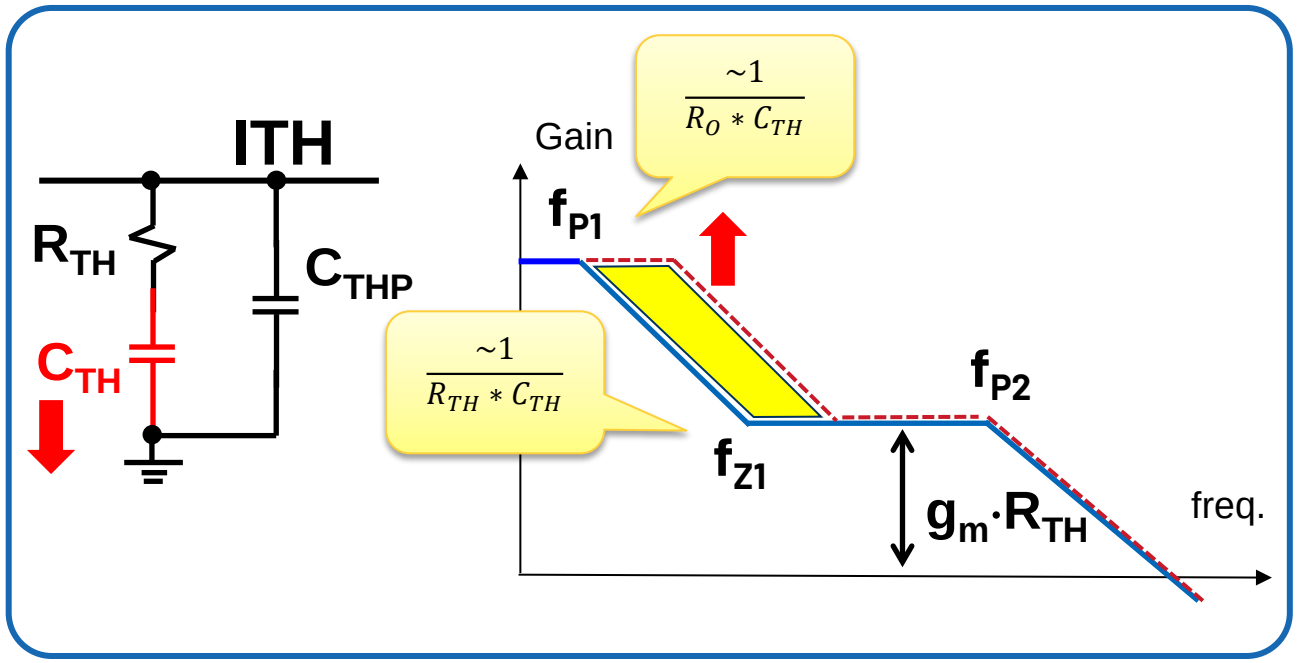
✓ **ΔV_{OUT} < ±3%**

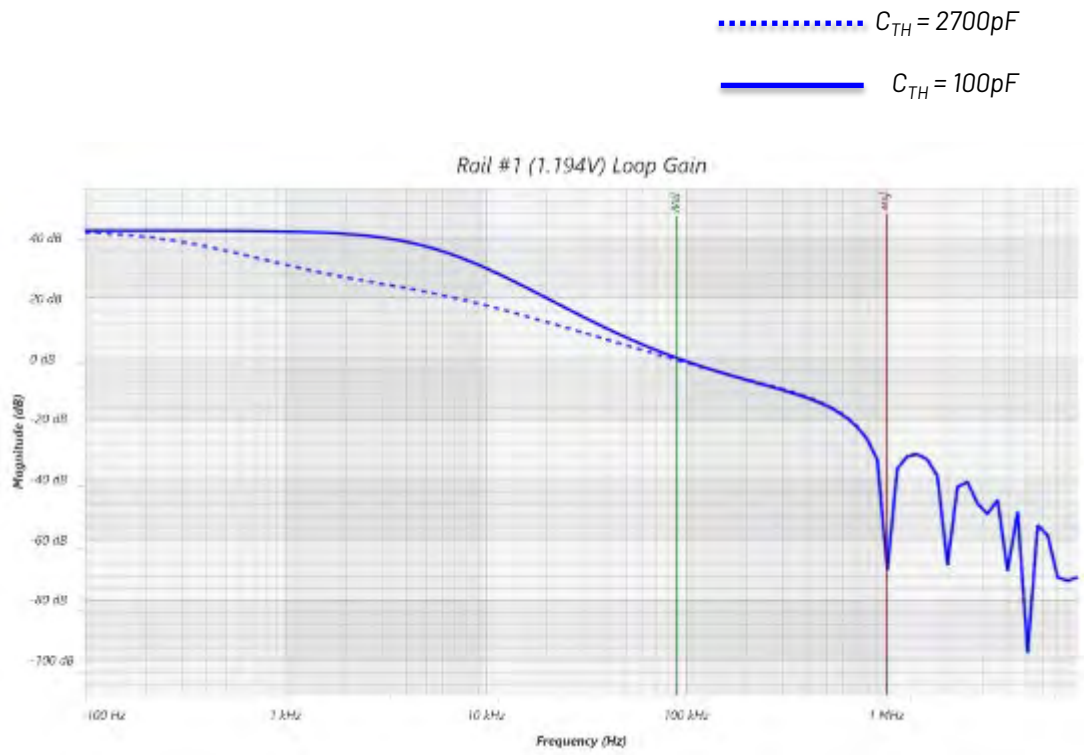
✓ **PM > 60°**

✓ **GM < -10dB**

✓ **Gain @ $f_{SW}/2$ < -8dB**

Step 3: Adjust Settling Time





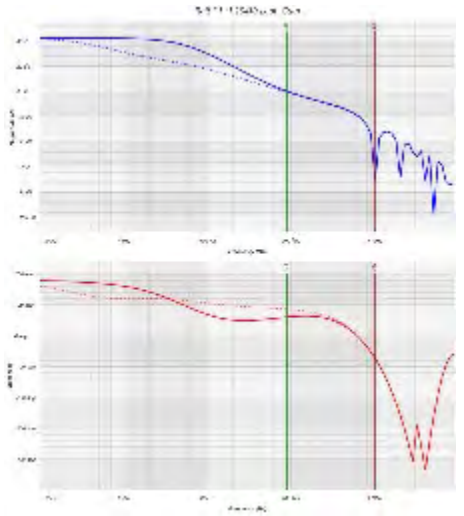
- Lower C_{TH} reduces settling time because of the High Gain in low frequency.
  - Potentially reducing Phase Margin if BW is around Fz1

Step 3: Adjust Settling Time cont.

CTH: 2700pF to 100pF

RTH = 30.1K and CTHP = 4.7pF

Stability Metric	CTH = 2700pF	CTH = 2200pF	CTH = 1500pF	CTH = 220pF	CTH = 100pF	Effects decreasing CTHP
Bandwidth (kHz)	89.13	89.13	89.13	89.13	89.13	**No Effect**
Phase Margin (Deg)	85.69	85.45	84.83	73.9	60.79	**Decreases**
Gain @ fSW/2 (dB)	-15.42	-15.42	-15.43	-15.57	-15.75	**Slight effect**
Gain Margin (dB)	-19.48	-19.48	-19.49	-19.61	-19.76	**Slight effect**
ΔV_{OUT} (%)	2.91	2.85	2.81	2.49	2.33	**Improves**



CTH: 2700pF to 100pF

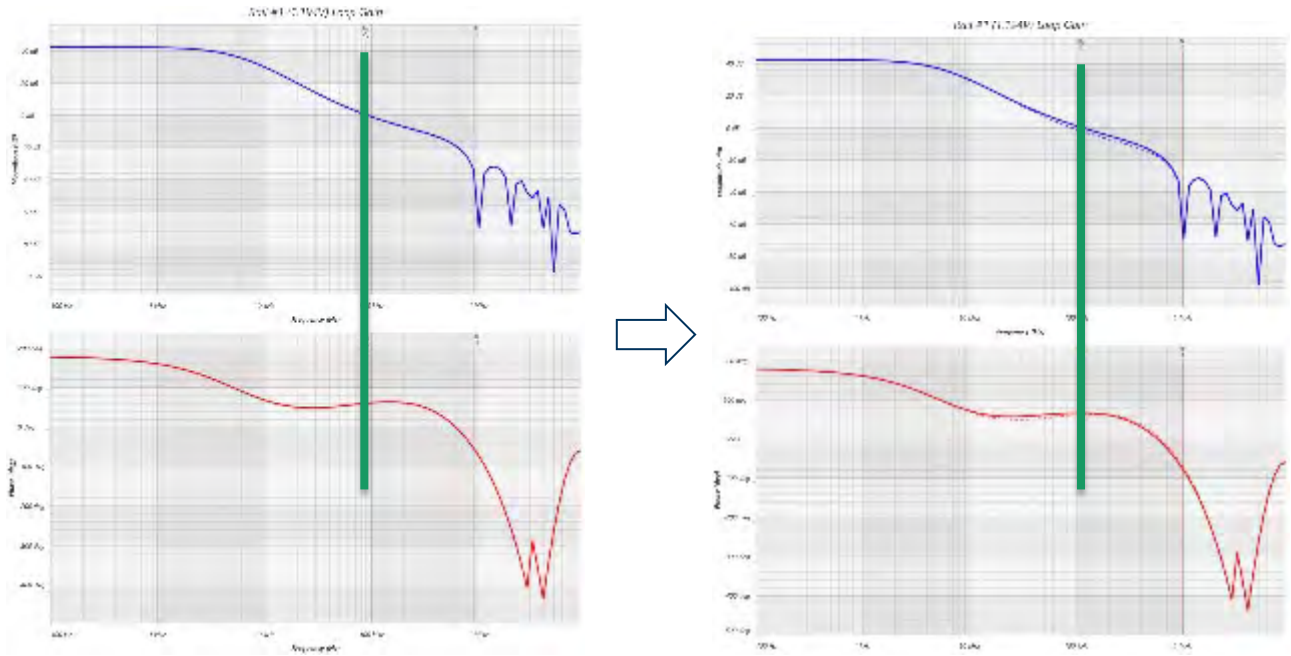
- ❑ Loop BW = 10%~ 20% f_{SW}
- ✓ **ΔV_{OUT} < ±3%**
- ✓ **PM > 60°**
- ✓ **GM < -10dB**
- ✓ **Gain @ f_{SW}/2 < -8dB**

Step 4 Optimize the best you can

Repeat Step 1 – 3 in any order depending on what is needed.

- Need more phase in Low frequency? Increase CTH
- Need more phase in High Frequency? Decrease CTHP
- Need more Gain Margin? Increase CTHP or Decrease RTH
- Need more BW? Increase RTH or decrease CTHP
- **It is an iterative process**
- **If you can't get there you may need more cap**

*Bonus: Aim for the top of the "phase peak" in phase plot for more robust design.
Increase RTH from 30.1K to 45.3K

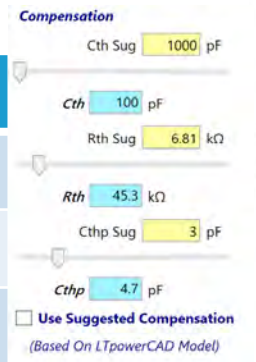


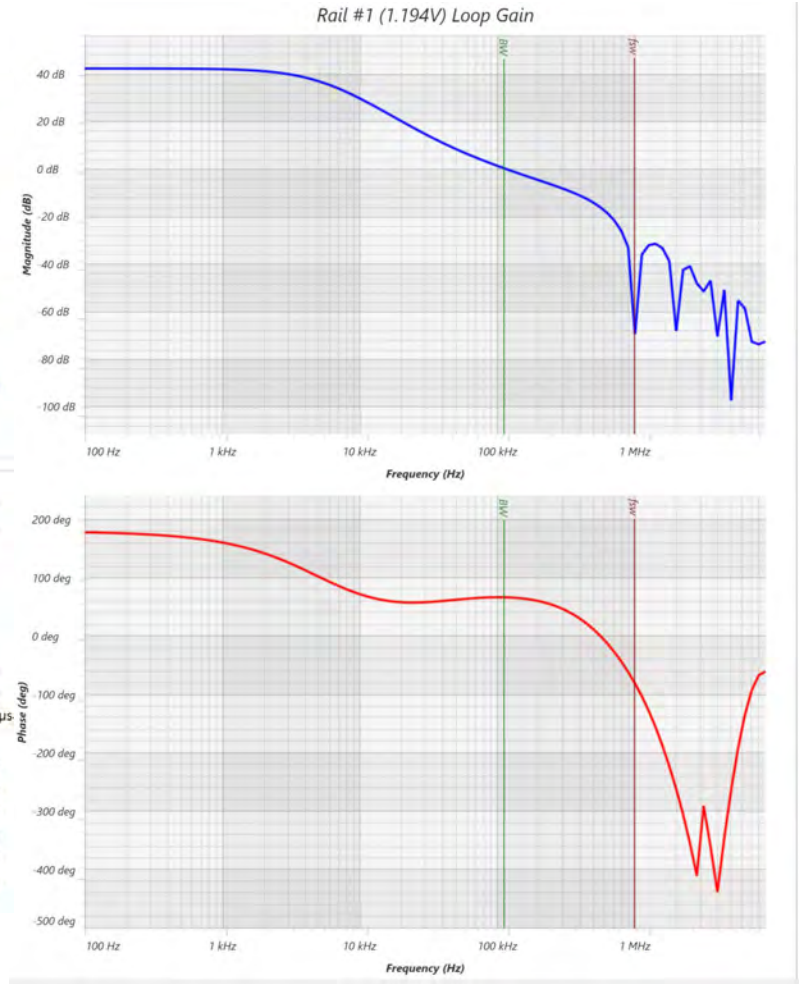
Comp	Value
CTH	100pF
RTH	30.1K
CTHP	4.7pF

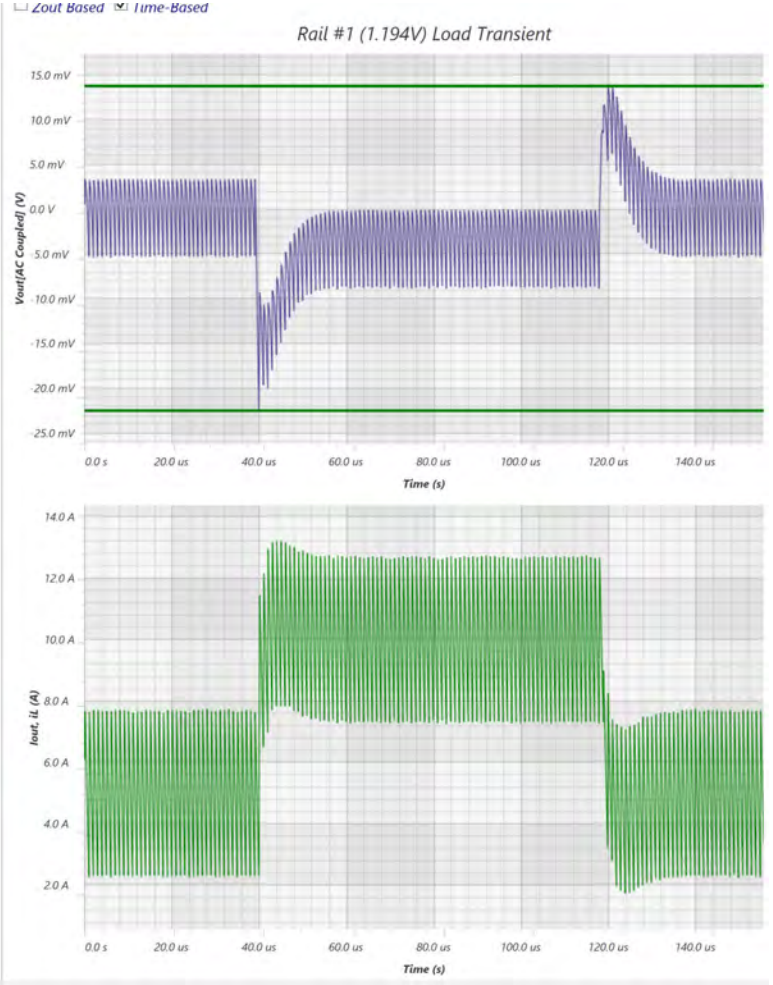
BW	89.13 kHz
PM	60.79°
GM	-15.75 dB
Gain at ½ Fsw	-19.76 dB
ΔV_{OUT}	+/- 2.91%

Final Step: Confirm Design Targets

Comp	Value
CTH	100pF
RTH	45.3K
CTHP	4.7pF







112KHz
✓ **Loop BW <= 10%~ 20% f_{SW}**

2.23%
✓ **ΔV_{OUT} < ±3%**

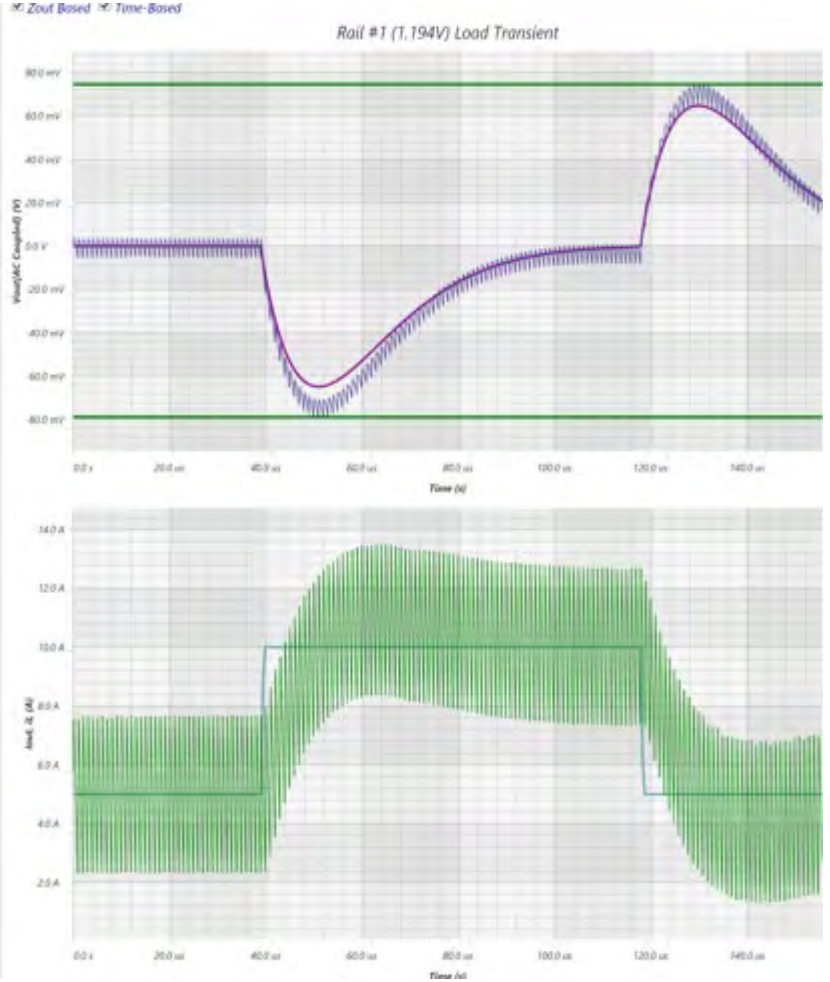
66.6 °
✓ **PM > 60°**

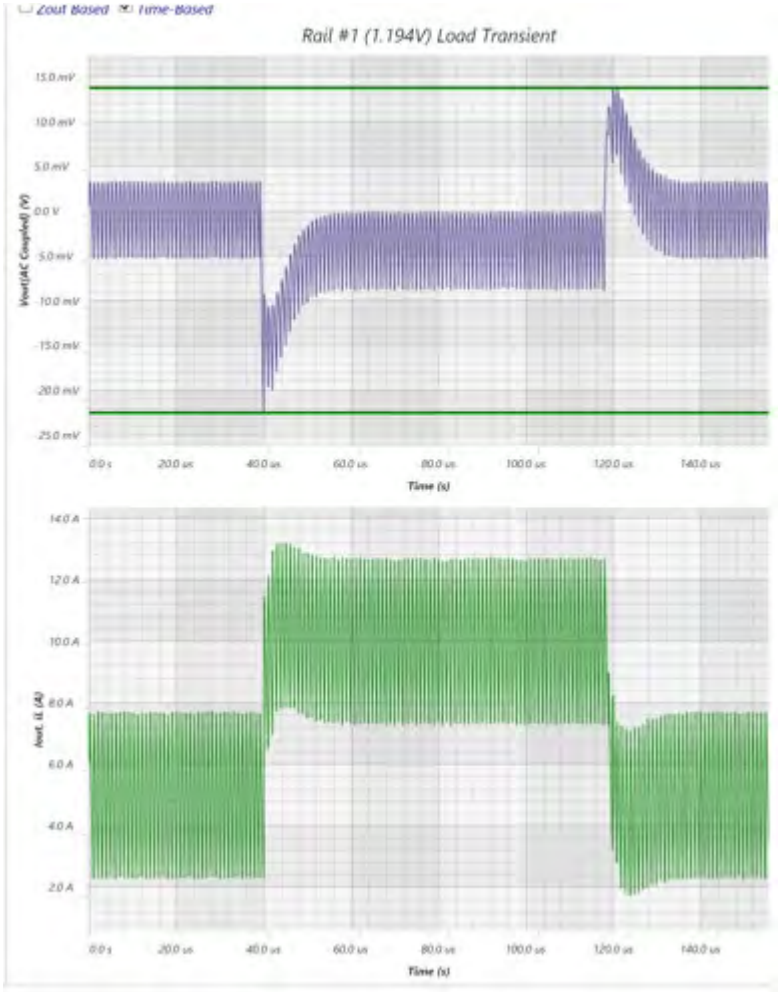
-16.05dB
✓ **GM < -10dB**

-14.1dB
✓ **Gain @ f_{SW}/2 < -8dB**

LTpowerCAD File:ADIPowerSeminar_LT8642S_RL1221_Feb2025

Final Step: Confirm Design Targets - Before/After



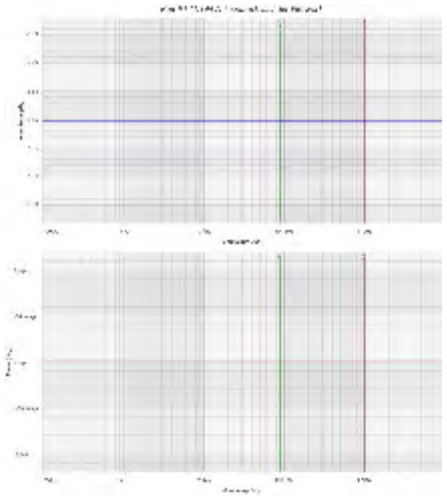


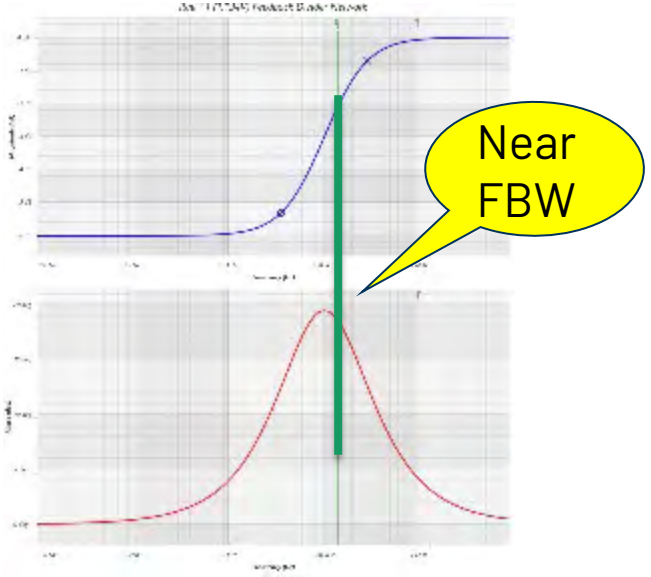
Design Targets

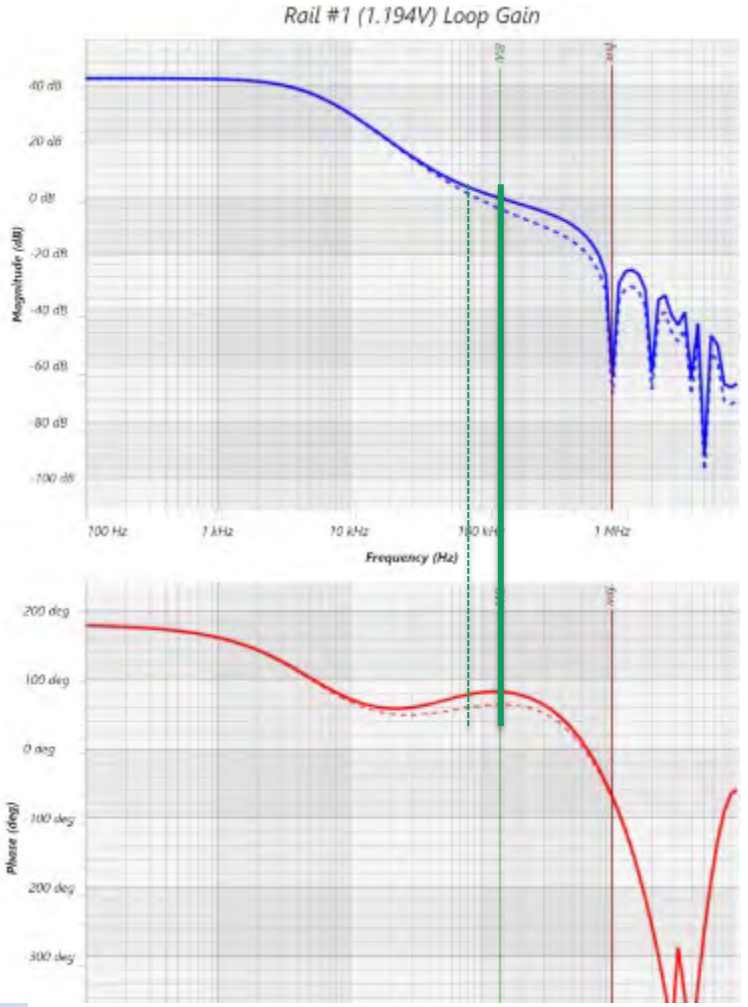
- **Loop BW = 10%~ 20% f_{SW}**
- **PM ≥45 degree. (≥60 degree preferred.)**
- **GM ≤ -10dB**
- **Gain @ $f_{SW}/2$ ≤ -8dB**
- **ΔV_{OUT} < ±3%**

	Before	After
BW	19.95 kHz	112 kHz
PM	76.16°	66.60°
GM	-33.44 dB	-16.05 dB
Gain at ½ Fsw	-33.44 dB	-14.10 dB
ΔV_{OUT}	+/- 6.04%	+/- 2.23%

Feedback Loop





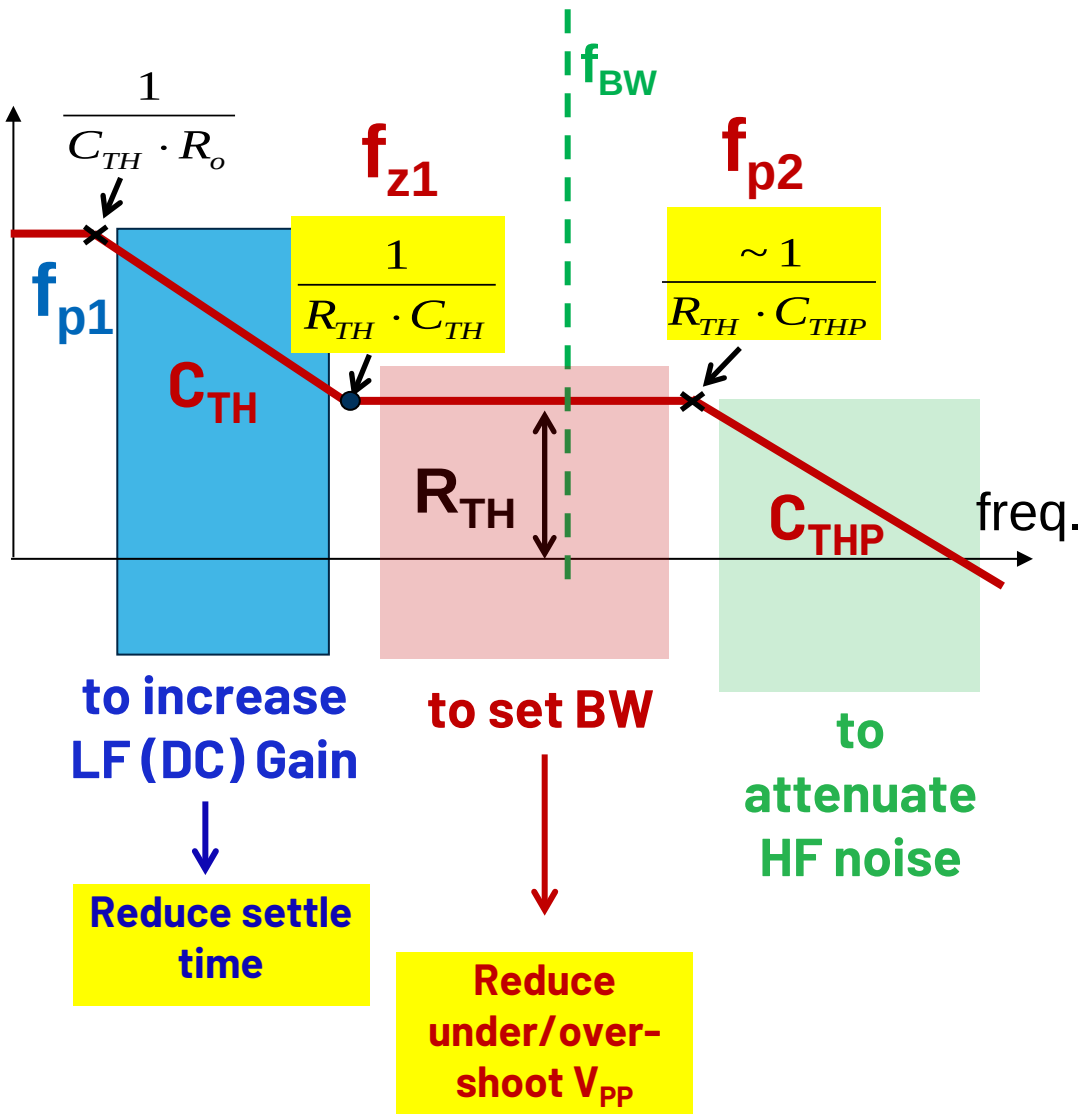


Stability Metric	CFF = OPEN	CFF = 22pF	Effects decreasing CTHP
Bandwidth (kHz)	89.13	141.25	**Increases**
Phase Margin (Deg)	60.79	82.4	**Increases Margin**
Gain @ fSW/2 (dB)	-15.75	-9.98	**Decreases Margin**
Gain Margin (dB)	-19.76	-16.8	**Decreases Margin**
ΔV_{OUT} (%)	2.33	2.23	**Improves**

✓ **Loop BW <= 10%~ 20% f_{SW}** ✓ **ΔV_{OUT} < ±3%** ✓ **PM > 60°** ✓ **GM < -10dB** ✓ **Gain @ $f_{SW}/2$ < -8dB**

Summary of Compensation Design

1. Set Bandwidth and Transient Response
    1. Adjust RTH (higher RTH increases BW, decreases PM)
    2. Check for transient response target
2. Check for HF Attenuation
    a) Adjust CTHP (higher CTHP improves noise immunity at the expense of BW)
3. Set Settling Time
    1. Adjust CTH (lower CTH reduces settling time at the expense of PM)
4. Optional: Feedforward Capacitor
    1. Check if phase boost is needed by looking at the PM and transient response
5. Close the loop
    1. Go back to Step 1 and re-check BW and Transient Response



Application Note 149 is an excellent discussion of loop compensation theory

ANALOG DEVICES

LINEAR TECHNOLOGY

Application Note 149

January 2015

Modeling and Loop Compensation Design of Switching Mode Power Supplies

Henry J. Zhang

INTRODUCTION

Today's electronic systems are becoming more and more complex, with an increasing number of power rails and supplies. To achieve optimum power solution density, reliability and cost, often system designers need to design their own power solutions, instead of just using commercial power supply bricks. Designing and optimizing high performance switching mode power supplies is becoming a more frequent and challenging task.

Power supply loop compensation design is usually viewed as a difficult task, especially for inexperienced supply designers. Practical compensation design typically involves numerous iterations on the value adjustment of the compensation components. This is not only time consuming, but is also inaccurate in a complicated system whose supply bandwidth and stability margin can be affected by several factors. This application note explains the basic concepts and methods of small signal modeling of switching mode power supplies and their loop compensation design. The buck step-down converter is used as the typical example, but the concepts can be applied to other topologies. A user-friendly LTpowerCAD™ design tool is also introduced to ease the design and optimization.

IDENTIFYING THE PROBLEM

A well-designed switching mode power supply (SMPS) must be quiet, both electrically and acoustically. An under-compensated system may result in unstable operations. Typical symptoms of an unstable power supply include: audible noise from the magnetic components or ceramic capacitors, jittering in the switching waveforms, oscillation of output voltage, overheating of power FETs and so on.

However, there are many reasons that can cause undesirable oscillation other than loop stability. Unfortunately, they all look the same on the oscilloscope to the inexperienced supply designer. Even for experienced engineers, sometimes identifying the reason that causes the instability can be difficult. Figure 1 shows typical output and switching node waveforms of an unstable buck supply. Adjusting the loop compensation may or may not fix the unstable supply because sometimes the oscillation is caused by other factors such as PCB noise. If you do not have a list of possibilities in your mind, uncovering the underlying cause of noisy operation can be very time-consuming and frustrating.

Thank you!