In [ ]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from UliEngineering.EngineerIO import format_value
from si_prefix import si_format
import plecs_helper as helper
%matplotlib
%matplotlib inline

# Imports and setup
from pint import UnitRegistry
from scipy.signal import find_peaks
from scipy.optimize import fsolve

# pandas display using scientific notation
# pd.set_option('display.float_format', lambda x: f'{x:.3e}')

# use pint
units = UnitRegistry()
units.default_format = "~P.2f"
```

```
Using matplotlib backend: TkAgg
```

Lab 7

Ian Eykamp

For the pre-lab, please see the separate notebook entitled `Lab7_calculations.ipynb` for snubber and clamp values and explanations.

Oscilloscope Data

In [ ]:

```python
# 10 & 11: base values, HF & LF, Vdmax = 74.4V, Iin = 1.38A & 1.38A (?), Vout = 9.8
# 12 & 13: HF resistor (10 ohm), HF & LF, Iin = 1.39A & 1.39A,Vdmax = 67.0V, Vout =
# 14 & 15: LF resistor (100 ohm), LF & HF, Iin = 1.33A & 1.38A, Vdmax = 72.1, Vout
# Vin = 18V

# 49: failed, 1.39A, 9.93V Vd = 54.4
# 50: for funsies, 6.5V, 0.62A
# 51: FINAL, 9.93Vout, 1.39A Iin,  Vd = 54V

(df_baseline_hf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile10
(df_baseline_lf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile11
(df_hf_snub_hf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile12.
(df_hf_snub_lf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile13.
(df_lf_snub_hf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile15.
(df_lf_snub_lf, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile14.
(df_clamp, tspan, tstep) = helper.read_rigol_csv("oscilloscope_data/NewFile51.csv",
```

```python
# Combine all variables into one for convenience
# df = df_4V_zoom1.set_index("t").join([df_4V.set_index("t"), df_7V.set_index("t"),
df_hf = df_baseline_hf.set_index("t").join([df_hf_snub_hf.set_index("t"), df_lf_snu
df_lf = df_baseline_lf.set_index("t").join([df_hf_snub_lf.set_index("t"), df_lf_snu
# df_clamp
```

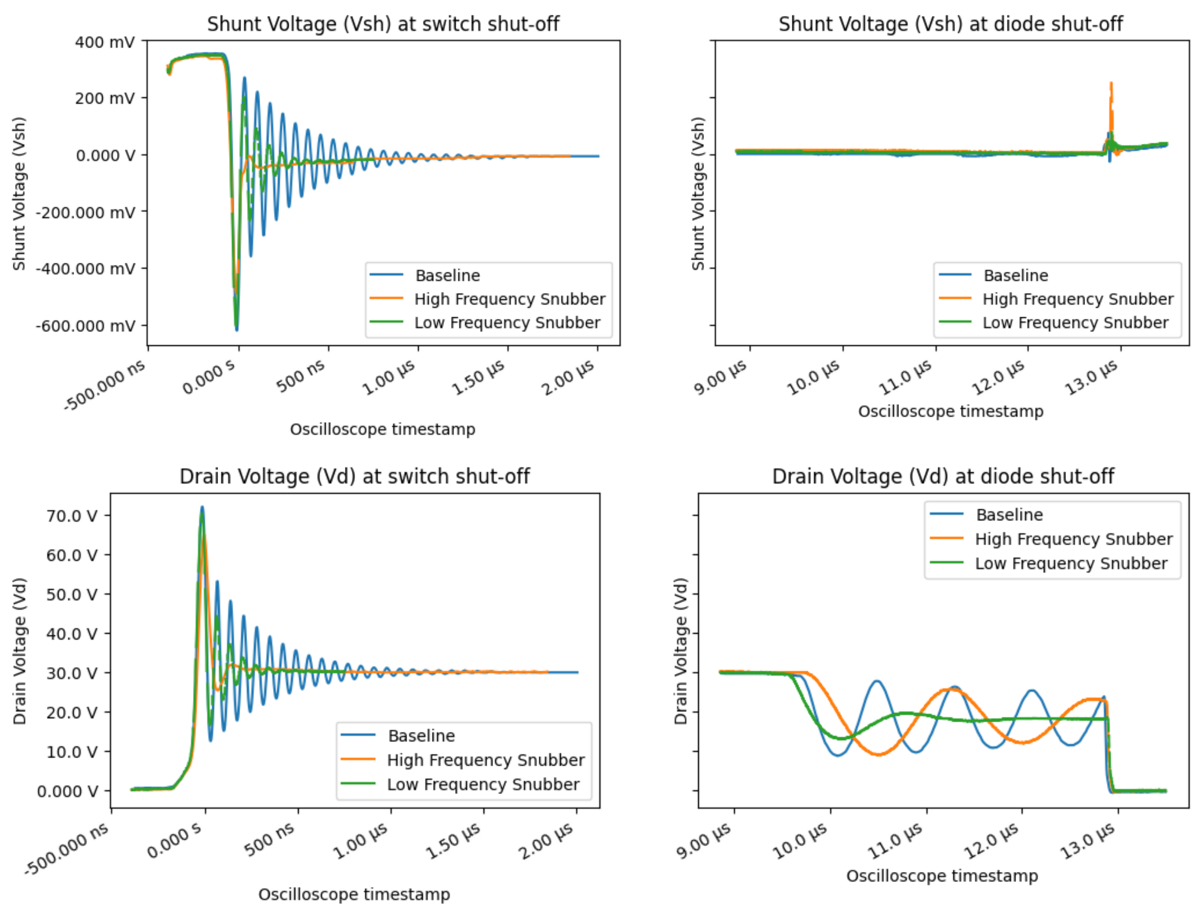
In [ ]:

```python
# Vshunt
df_hf_zoom = df_hf
# df_hf_zoom = df_hf[(df_hf["t"] > 0e-6) & (df_hf["t"] < 2e-6)]
df_lf_zoom = df_lf[(df_lf["t"] > 8e-6) & (df_lf["t"] < 13.5e-6)]
# df_lf_zoom = df_lf[(df_lf["t"] > 12.75e-6) & (df_lf["t"] < 13e-6)]

print("Switching Losses")
Vsh_baseline_lf_switch_area = np.trapz(df_lf_zoom[np.logical_not(np.isnan(df_lf_zoo
print("Baseline", (Vsh_baseline_lf_switch_area * units.volt ** 2 * units.second / (
Vsh_hf_snub_lf_switch_area = np.trapz(df_lf_zoom[np.logical_not(np.isnan(df_lf_zoom
print("High Frequency Snubber", (Vsh_hf_snub_lf_switch_area * units.volt ** 2 * uni
Vsh_lf_snub_lf_switch_area = np.trapz(df_lf_zoom[np.logical_not(np.isnan(df_lf_zoom
print("Low Frequency Snubber", (Vsh_lf_snub_lf_switch_area * units.volt ** 2 * unit

print("High-Frequency Ringing")
Vsh_baseline_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoo
print("Baseline", (Vsh_baseline_hf_switch_area * units.volt ** 2 * units.second / (
Vsh_hf_snub_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoom
print("High Frequency Snubber", (Vsh_hf_snub_hf_switch_area * units.volt ** 2 * uni
Vsh_lf_snub_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoom
print("Low Frequency Snubber", (Vsh_lf_snub_hf_switch_area * units.volt ** 2 * unit

print("High-Frequency Ringing AC Resistance")
Vsh_baseline_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoo
print("Baseline", (Vsh_baseline_hf_switch_area * units.volt ** 2 * units.second / (
Vsh_hf_snub_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoom
print("High Frequency Snubber", (Vsh_hf_snub_hf_switch_area * units.volt ** 2 * uni
Vsh_lf_snub_hf_switch_area = np.trapz(df_hf_zoom[np.logical_not(np.isnan(df_hf_zoom
print("Low Frequency Snubber", (Vsh_lf_snub_hf_switch_area * units.volt ** 2 * unit

fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, sharex = False, sharey = True,
fig.autofmt_xdate()
helper.axes_labels("Oscilloscope timestamp", "s", "Shunt Voltage (Vsh)", "V", title
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vsh_baseline_hf"], label = "Baseline")
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vsh_hf_snub_hf"], label = "High Frequency Snu
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vsh_lf_snub_hf"], label = "Low Frequency Snub
ax1.legend(loc = "lower right")

helper.axes_labels("Oscilloscope timestamp", "s", "Shunt Voltage (Vsh)", "V", title
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vsh_baseline_lf"], label = "Baseline")
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vsh_hf_snub_lf"], label = "High Frequency Snu
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vsh_lf_snub_lf"], label = "Low Frequency Snub
ax2.legend(loc = "lower right")

# Vd
df_hf_zoom = df_hf
df_lf_zoom = df_lf[(df_lf["t"] > 8e-6) & (df_lf["t"] < 13.5e-6)]

fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, sharex = False, sharey = True,
fig.autofmt_xdate()
```

```
helper.axes_labels("Oscilloscope timestamp", "s", "Drain Voltage (Vd)", "V", title
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vd_baseline_hf"], label = "Baseline")
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vd_hf_snub_hf"], label = "High Frequency Snub
ax1.plot(df_hf_zoom["t"], df_hf_zoom["Vd_lf_snub_hf"], label = "Low Frequency Snubb
ax1.legend(loc = "lower right")

helper.axes_labels("Oscilloscope timestamp", "s", "Drain Voltage (Vd)", "V", title
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vd_baseline_lf"], label = "Baseline")
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vd_hf_snub_lf"], label = "High Frequency Snub
ax2.plot(df_lf_zoom["t"], df_lf_zoom["Vd_lf_snub_lf"], label = "Low Frequency Snubb
ax2.legend(loc = "upper right")
```

```
Switching Losses
Baseline 375.78 µW
High Frequency Snubber 1.85 mW
Low Frequency Snubber 713.97 µW
High-Frequency Ringing
Baseline 59.59 mW
High Frequency Snubber 44.89 mW
Low Frequency Snubber 51.33 mW
High-Frequency Ringing AC Resistance
Baseline 5.57 W
High Frequency Snubber 4.19 W
Low Frequency Snubber 4.79 W
```

Out[ ]: <matplotlib.legend.Legend at 0x1749967aa30>



High-Frequency Ring (after switch shut-off)

The high-frequency snubber design works exceptionally well; the high frequency oscillation is very close to critically damped. Whereas the baseline signal oscillates for at least 1us, the damped signal is completely attenuated within a quarter of that time. The low-frequency snubber design is less effective at dampening the high-freuqency ringing, but it is nonetheless significantly better than the baseline.

Low-Frequency Ring (after diode shut-off)

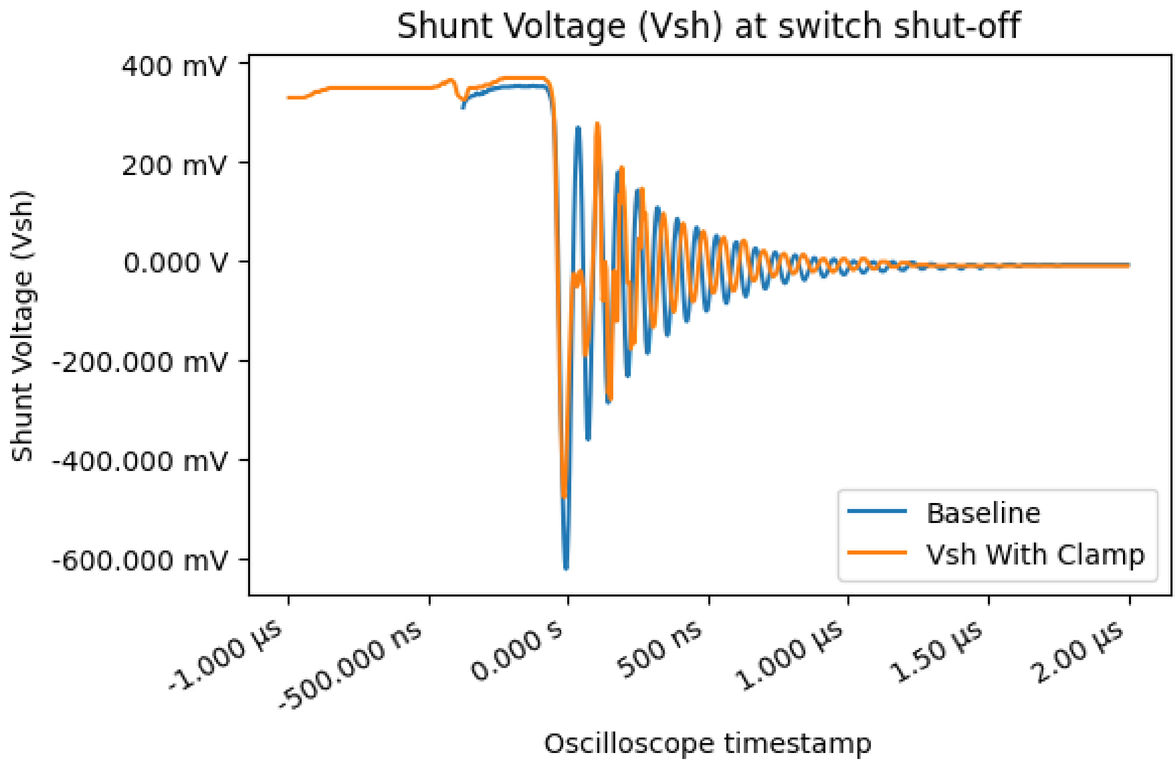
The high-frequency snubber significantly changes the period and phase shift of the low-frequency ring relative to the baseline, but it provides little dampening. The low-frequency snubber again is nearly critically damped and effectively halts the oscillation within about 2us, well before the switch turns on again.

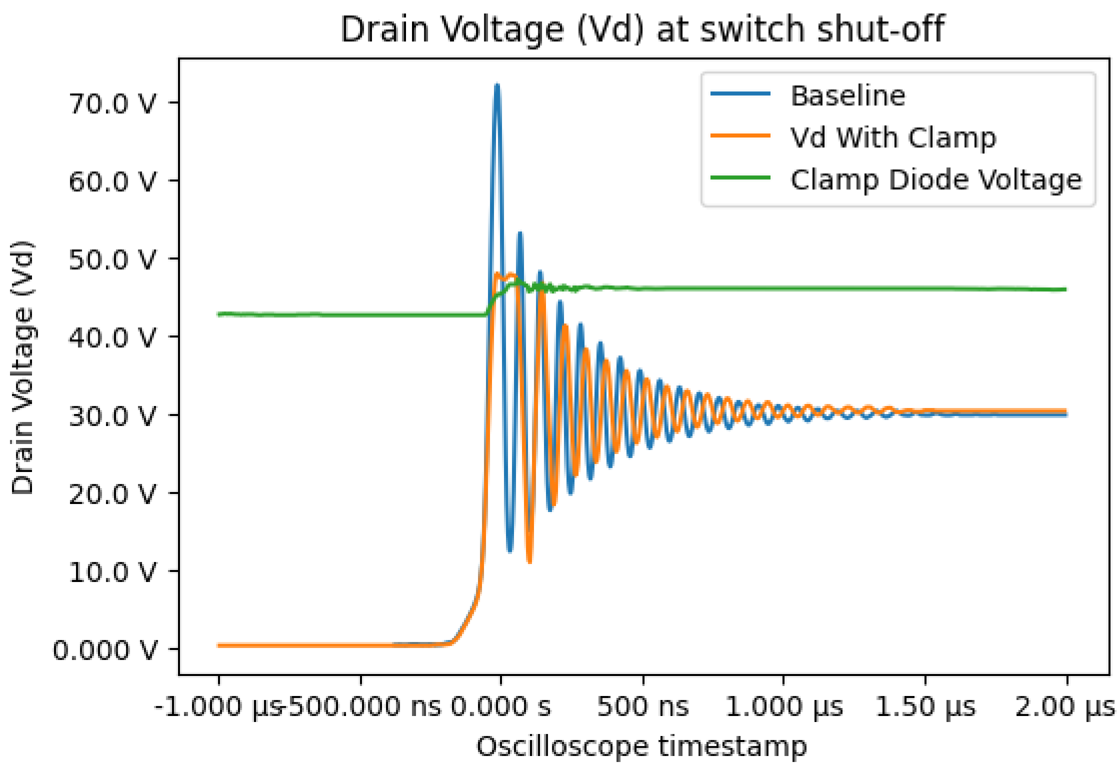
In [ ]:

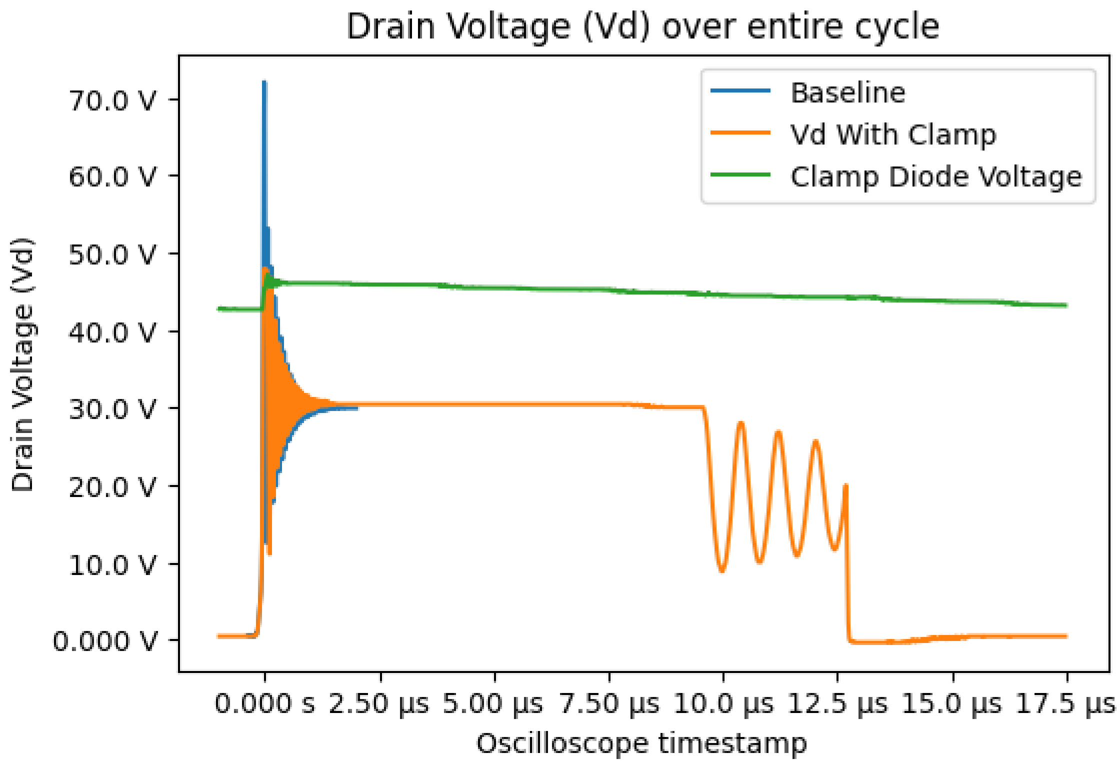
```
# Vd
bounds = (-1e-6, 2e-6)
df_clamp_zoom = df_clamp
df_baseline_zoom = df_baseline_hf
df_clamp_zoom = df_clamp[(df_clamp["t"] > bounds[0]) & (df_clamp["t"] < bounds[1])]
df_baseline_zoom = df_baseline_hf[(df_baseline_hf["t"] > bounds[0]) & (df_baseline_

fig, (ax1) = plt.subplots(nrows = 1, ncols = 1, sharex = True, sharey = False, figs
fig.autofmt_xdate()
helper.axes_labels("Oscilloscope timestamp", "s", "Shunt Voltage (Vsh)", "V", title
ax1.plot(df_baseline_zoom["t"], df_baseline_zoom["Vsh_baseline_hf"], label = "Basel
ax1.plot(df_clamp_zoom["t"], df_clamp_zoom["Vsh"], label = "Vsh With Clamp")
ax1.legend(loc = "lower right")

fig, (ax1) = plt.subplots(nrows = 1, ncols = 1, sharex = True, sharey = False, figs
helper.axes_labels("Oscilloscope timestamp", "s", "Drain Voltage (Vd)", "V", title
ax1.plot(df_baseline_zoom["t"], df_baseline_zoom["Vd_baseline_hf"], label = "Baseli
ax1.plot(df_clamp_zoom["t"], df_clamp_zoom["Vd"], label = "Vd With Clamp")
ax1.plot(df_clamp_zoom["t"], df_clamp_zoom["Vdiode"], label = "Clamp Diode Voltage"
ax1.legend(loc = "upper right")

df_clamp_zoom = df_clamp[(df_clamp["t"] > bounds[0]) & (df_clamp["t"] < 40e-6)]

fig, (ax1) = plt.subplots(nrows = 1, ncols = 1, sharex = True, sharey = False, figs
helper.axes_labels("Oscilloscope timestamp", "s", "Drain Voltage (Vd)", "V", title
ax1.plot(df_baseline_zoom["t"], df_baseline_zoom["Vd_baseline_hf"], label = "Baseli
ax1.plot(df_clamp_zoom["t"], df_clamp_zoom["Vd"], label = "Vd With Clamp")
ax1.plot(df_clamp_zoom["t"], df_clamp_zoom["Vdiode"], label = "Clamp Diode Voltage"
ax1.legend(loc = "upper right")
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x1749c688c40>
```


Shunt Voltage (Vsh) at switch shut-off
Shunt Voltage (Vsh)
400 mV
200 mV
0.000 V
-200.000 mV
-400.000 mV
-600.000 mV
-1.000 µs
-500.000 ns
0.000 s
500 ns
1.000 µs
1.50 µs
2.00 µs
Oscilloscope timestamp
Baseline
Vsh With Clamp

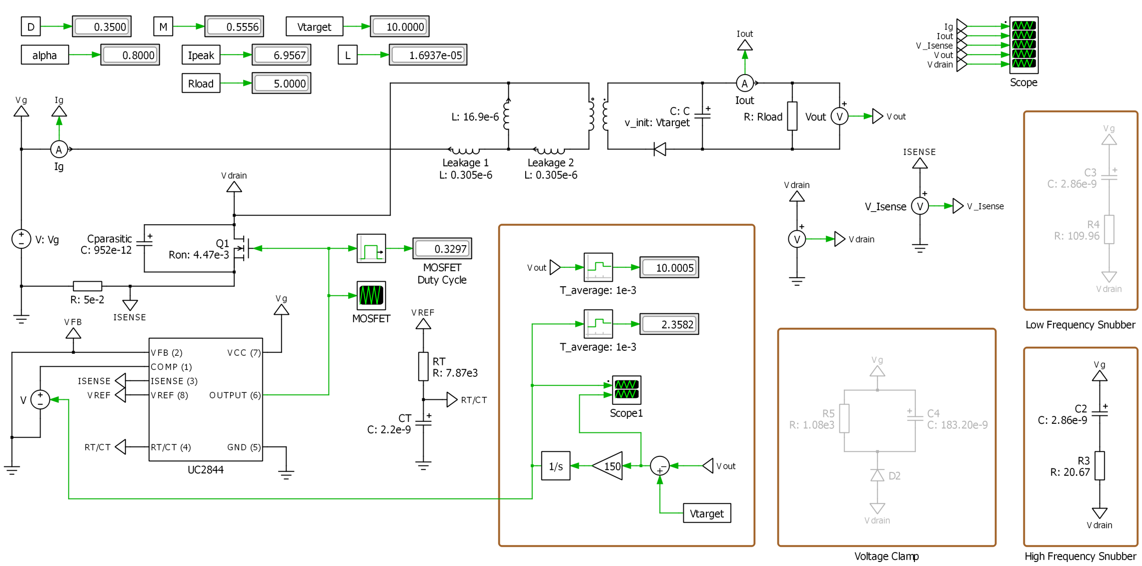

Drain Voltage (Vd) at switch shut-off
Drain Voltage (Vd)
70.0 V
60.0 V
50.0 V
40.0 V
30.0 V
20.0 V
10.0 V
0.000 V
-1.000 µs
-500.000 ns
0.000 s
500 ns
1.000 µs
1.50 µs
2.00 µs
Oscilloscope timestamp
Baseline
Vd With Clamp
Clamp Diode Voltage



The clamp voltage behaves exactly as expected. It cuts off the first one or two peaks of the drain voltage ringing when it exceeds 45V. Once the amplitude stays consistently below the clamp threshold, the ringing continues as normal. Note that this data was taken with the clamp in place but with the snubbers removed.

PLECS Simulations

The snubbers and clamp are implemented on the lower right hand side of the schematic. They are made to be easily commented out in various combinations.



In [ ]:

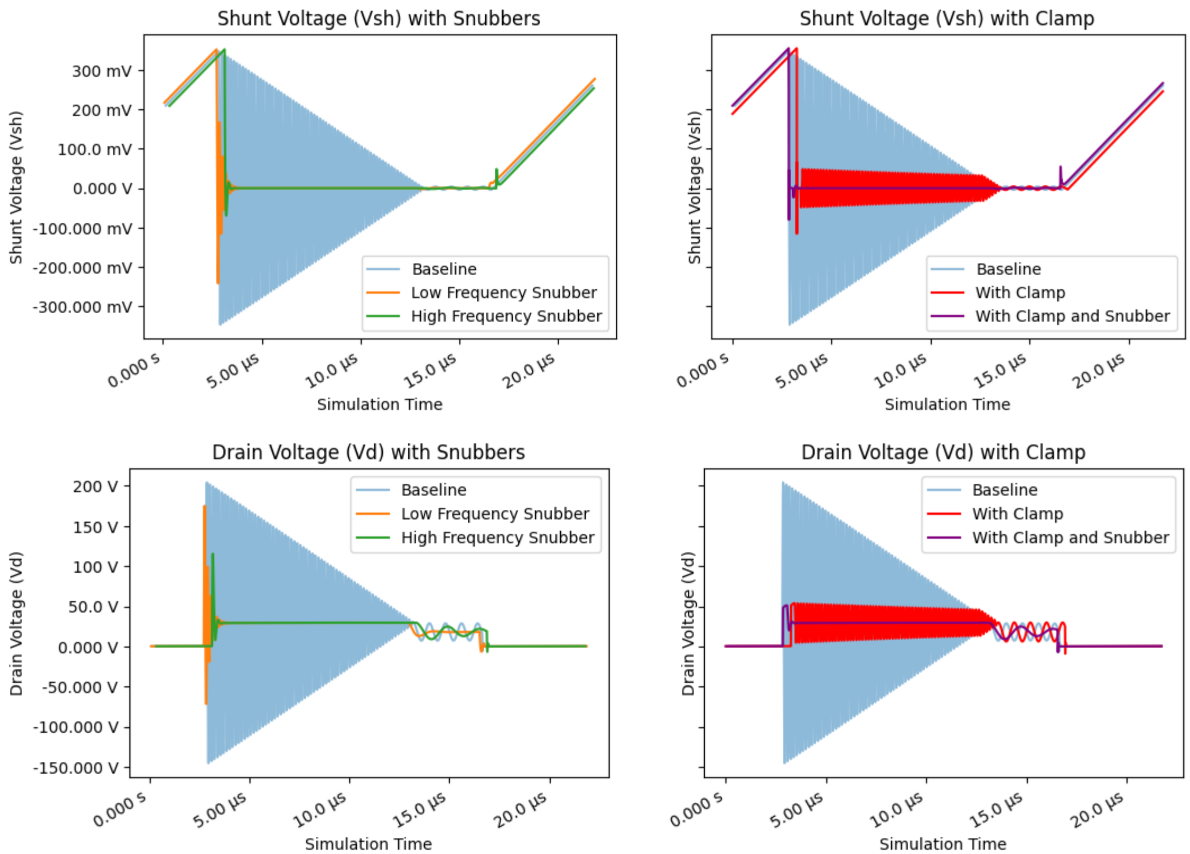
```
bounds = (8.028e-3, 8.05e-3)

df_no_snubber = pd.read_csv("plecs_data_snubber_2/no_snubber.csv")
df_no_snubber.rename(mapper = helper.strip_labels, axis = "columns", inplace = True
df_no_snubber = df_no_snubber.loc[(df_no_snubber["t"] > bounds[0]) & (df_no_snubber
df_no_snubber["t"] = df_no_snubber["t"] - bounds[0]

df_hf_snubber = pd.read_csv("plecs_data_snubber_2/hf_snubber.csv")
df_hf_snubber.rename(mapper = helper.strip_labels, axis = "columns", inplace = True
df_hf_snubber = df_hf_snubber.loc[(df_hf_snubber["t"] > bounds[0]) & (df_hf_snubber
df_hf_snubber["t"] = df_hf_snubber["t"] - bounds[0]

df_lf_snubber = pd.read_csv("plecs_data_snubber_2/lf_snubber.csv")
df_lf_snubber.rename(mapper = helper.strip_labels, axis = "columns", inplace = True
df_lf_snubber = df_lf_snubber.loc[(df_lf_snubber["t"] > bounds[0]) & (df_lf_snubber
df_lf_snubber["t"] = df_lf_snubber["t"] - bounds[0]

df_with_clamp = pd.read_csv("plecs_data_snubber_2/with_clamp.csv")
df_with_clamp.rename(mapper = helper.strip_labels, axis = "columns", inplace = True
df_with_clamp = df_with_clamp.loc[(df_with_clamp["t"] > bounds[0]) & (df_with_clamp
df_with_clamp["t"] = df_with_clamp["t"] - bounds[0]

df_hf_snubber_and_clamp = pd.read_csv("plecs_data_snubber_2/hf_snubber_and_clamp.cs
df_hf_snubber_and_clamp.rename(mapper = helper.strip_labels, axis = "columns", inpl
df_hf_snubber_and_clamp = df_hf_snubber_and_clamp.loc[(df_hf_snubber_and_clamp["t"]
df_hf_snubber_and_clamp["t"] = df_hf_snubber_and_clamp["t"] - bounds[0]

# print(df_no_snubber.head(1))
# print(df_with_clamp.head(1))
```

In [ ]:

```
baseline_transparency = 0.5

# Vshunt
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, sharex = False, sharey = True,
fig.autofmt_xdate()
helper.axes_labels("Simulation Time", "s", "Shunt Voltage (Vsh)", "V", title = "Shu
ax1.plot(df_no_snubber["t"], df_no_snubber["V_Isense"], alpha = baseline_transparen
ax1.plot(df_lf_snubber["t"], df_lf_snubber["V_Isense"], label = "Low Frequency Snub
ax1.plot(df_hf_snubber["t"], df_hf_snubber["V_Isense"], label = "High Frequency Snu
ax1.legend(loc = "lower right")

helper.axes_labels("Simulation Time", "s", "Shunt Voltage (Vsh)", "V", title = "Shu
ax2.plot(df_no_snubber["t"], df_no_snubber["V_Isense"], color = "#1f77b4", alpha =
ax2.plot(df_with_clamp["t"], df_with_clamp["V_Isense"], color = "red", label = "Wit
ax2.plot(df_hf_snubber_and_clamp["t"], df_hf_snubber_and_clamp["V_Isense"], color =
ax2.legend(loc = "lower right")

# Vd
fig, (ax1, ax2) = plt.subplots(nrows = 1, ncols = 2, sharex = False, sharey = True,
fig.autofmt_xdate()
helper.axes_labels("Simulation Time", "s", "Drain Voltage (Vd)", "V", title = "Drai
ax1.plot(df_no_snubber["t"], df_no_snubber["Vdrain"], alpha = baseline_transparency
ax1.plot(df_lf_snubber["t"], df_lf_snubber["Vdrain"], label = "Low Frequency Snubbe
ax1.plot(df_hf_snubber["t"], df_hf_snubber["Vdrain"], label = "High Frequency Snubb
ax1.legend(loc = "upper right")
```

```
helper.axes_labels("Simulation Time", "s", "Drain Voltage (Vd)", "V", title = "Drai
ax2.plot(df_no_snubber["t"], df_no_snubber["Vdrain"], color = "#1f77b4", alpha = ba
ax2.plot(df_with_clamp["t"], df_with_clamp["Vdrain"], color = "red", label = "With
ax2.plot(df_hf_snubber_and_clamp["t"], df_hf_snubber_and_clamp["Vdrain"], color = "
ax2.legend(loc = "upper right")
```

Out[ ]: <matplotlib.legend.Legend at 0x1749bd41520>



The baseline simulation includes an LCR circuit consisting of the leakage inductance, parasitic capacitance, and shunt resistance in series. The simulated baseline ringing lasts much longer than the measured baseline ringing because the physical circuit has frequency-dependent AC resistance due to the skin effect and other factors which are not modeled. These high-frequency phenomena affect the switch shutoff ringing because it is much higher frequency (~20MHz) than the rest of the waveform.

Once the snubbers are added, the simulation results match closely with the measured waveform. In particular, the high frequency snubber yields near-critical damping on the switch shutoff ringing, and the low frequency snubber critically dampens or overdampens the diode shutoff response.

The snubber introduces an interesting spike in the shunt voltage

The clamp behaves as expected. In the absense of a snubber, it simply caps the drain voltage at around 50V when it would otherwise go higher; it does nothing to dampen the ringing. When both the clamp and the snubber are added, the switch shutoff response is nearly perfect: the drain voltage rises up to a safe maximum, stalls, then falls quickly to the desired output voltage.