Lab 6

Lab Link: https://pages.hmc.edu/mspencer/e157/fa24/labs/06.pdf

GitHub with files and scripts: https://github.com/kavidey/e157/tree/main/lab_06

Theory Questions

1. Do some math to analyze the spectrum analyzer in our lab (an N9320b)
   a. What is the maximum input power of the spectrum analyzer? How many 30dBm amplifiers can you put in series with a -30dBm function generator before you are violating the maximum input power?
      i. The maximum average continuous power is +37 dBm and the maximum peak pulse power is + 50 dBm. For this problem, we care about average continuous power.

$$-30 \text{ dBm } + n30 \text{ dB} \leq 37 \text{ dBm} \implies n = 2$$

      ii. Therefore we can put two 30 dBm amplifiers in series with a -30 dBm signal.
   b. What is the system temperature of our spectrum analyzer and how does it relate to DANL (displayed average noise level).
      i. The relationship between DANL and system temperature can be found as follows:

$$P \text{ dBm} = 10\log(kTB \cdot 1000)$$
$$1000 \cdot kTB = \exp_{10}\left(\frac{P \text{ dBm}}{10}\right)$$
$$T = \frac{\exp_{10}\left(\frac{P \text{ dBm}}{10}\right)}{kB \cdot 1000}$$

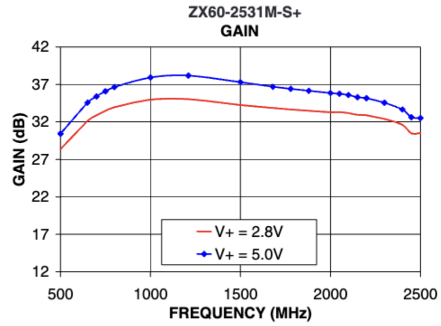
      1. From the datasheet, the DANL is

Displayed average noise level (DANL)			
Input terminated, 0 dB RF attenuation, RBW = 10 Hz, VBW = 1 Hz, sample detector			
Preamp off		Specification	Typical
	9 to 100 kHz		-90 dBm nominal
	100 to 500 kHz	-90 dBm – 3 x (f/100 kHz) dB	-106 dBm
	500 kHz to 1 MHz		-126 dBm
	1 to 10 MHz	-124 dBm	-130 dBm
	10 to 500 MHz	-130 dBm +3 x (f/1GHz)dB	-132 dBm
	500 MHz to 1.5 GHz		-130 dBm
	1.5 to 2.5 GHz		-128 dBm
	2.5 to 3 GHz		-125 dBm
Preamp on	100 to 500 kHz	-108 dBm – 3 × (f/100 kHz) dB	-124 dBm
	500 kHz to 1 MHz		-145 dBm
	1 to 10 MHz	-142 dBm	-149 dBm
	10 to 500 MHz	-148 dBm + 3 x (f/1 GHz) dB	-150 dBm
	500 MHz to 1.5 GHz		-148 dBm
	1.5 to 2.5 GHz		-146 dBm
	2.5 to 3 GHz		-141 dBm

DANL for spectrum analyzer from page 4 of the datasheet

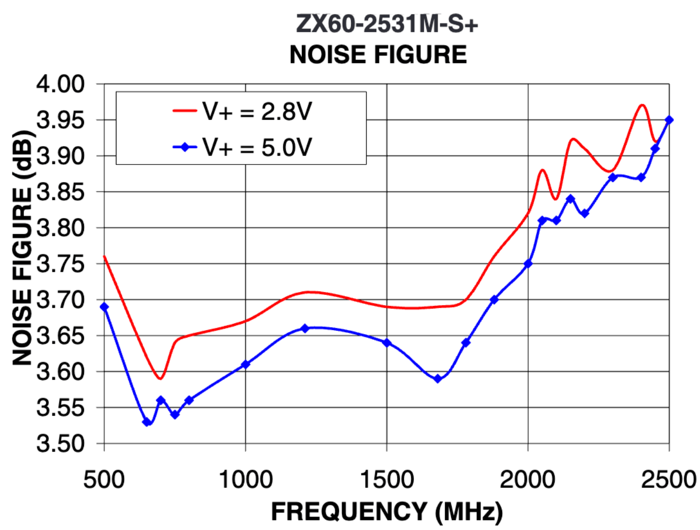
At the range we will be operating at with the preamp off, the noise temperature is -110 dB. This results in a system temperature is 724e6 kelvin.

2. Use values on the datasheet to calculate the gain, noise figure, output power at P-1dB, and OIP3 of the ZX60-2531M-S+ amplifier at 600 MHz powered with a +5V supply.
   a. Gain:



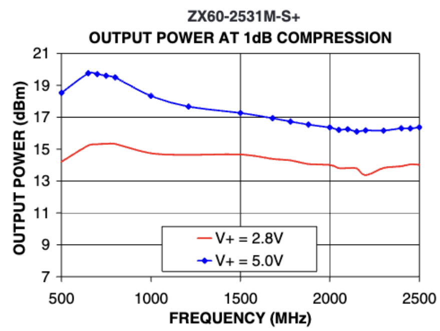
      1. At 600 MHz and 5V the gain is around 32 dB

   b. Noise Figure



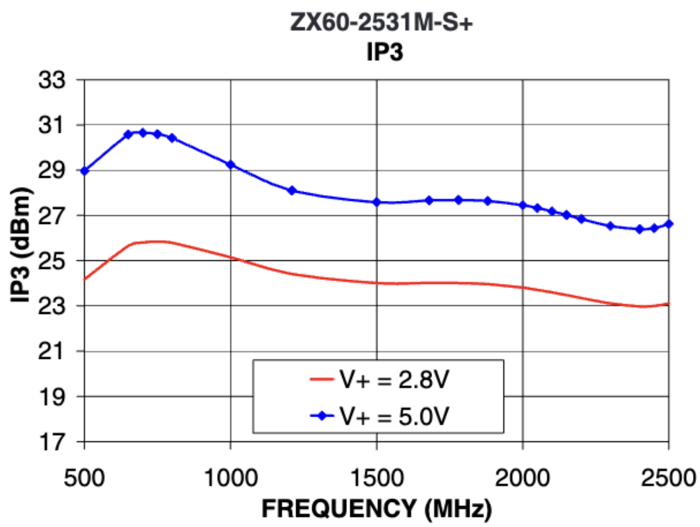
      1. At 600 MHz and 5V the noise factor is 3.6 dB

   c. Output power at P-1dB



1. At 600 MHz and 5V the output power at P-1dB is around 19 dBm

d. OIP3



1. In general, datasheets can list IP3 as IIP3, OIP3, or the 2D point. In this datasheet, we know that the output power at P-1dB is 19 dBm. This means that the input P-1dB power is 19 dBm - 32 dB + 1 = -12 dBm. Then we can calculate IIP3 = -2.4 dB as IIP3 = P-1dB + 9.6 dB.
2. This is around 30 dB off of the value in the IP3 plot, implying that this is actually a plot of OIP3. Adding 32 dB to IIP3 or looking at the plot both give an OIP3 value of 29.6 dBm.

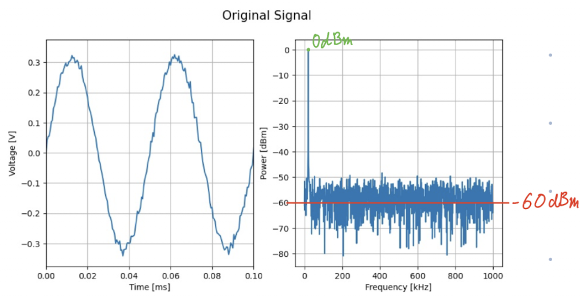
3. What is the noise temperature of a 30dB attenuator?
   a. From theory we know that the noise temperature of a lossy passive as follows:

$$T_p = \left(\frac{1}{L} - 1\right) T$$
$$T_p = (1000 - 1) \cdot 300\ K = 299700\ K$$

Optional Extra Credit:

We are going to do some joint computation / hand analysis to practice understanding noise density and noise power. The steps below instruct you to conduct various operations in Matlab (or Python). For each operation, you need to compare your computed results to “what you expect”, which you should be able to determine using hand calculations. Include hand calculations and an FFT annotated with the important extracted values (usually signal and noise power levels) for each step. Instead of FFT coefficients, represent the power in each FFT bin (which may require you to find the magnitude of the bins to start), and do so in units of dBm. Also include your code.

1. Create 20.0 cycles of a 20kHz sine wave representing a 0dBm sinusoid driven into a 50 Ohm load. This signal is intended to represent a continuous analog signal, though numerical software will always represent it as a vector of samples, so use a high sample rate: fs=2MHz. Add white noise to the sine wave with σ_v = 10mV.



Noise Floor

$\sigma_v = 10mV = 0.01V$

FT Freq. Resolution: $\frac{f_s}{N_{sample}} = \frac{2MHz}{2000} = 1kHz$

Noise Power Density: $\overline{\sigma_p}^2 = \frac{\sigma_v^2}{4R} = \boxed{-63\,dBm}$

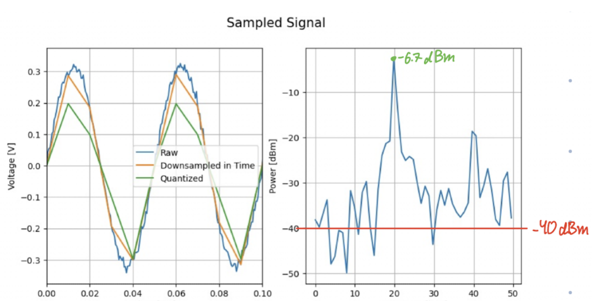
50Ω

Signal Power Level

Expected: $1mW = \boxed{0dBm}$

   1. Where is the noise floor? (Note that the simulated value is easy enough to pluck off the graph, but finding the analytical value requires you to observe that 10mV is your total noise standard deviation, so you need to convert that back into a noise power in each bin. Here the observed noise in each bin represents some density captured across a bandwidth of the bin width.)
      a. expected: -63 dBm
      b. actual: -60 dBm
   2. Where is the signal power level?
      a. expected: 0 dBm
      b. actual: 0 dBm

2. Write a function that decimates and quantizes your signal like an 8-bit ADC sampling at 100kHz. Assuming the ADC full scale is equal to five times your wave amplitude.



Noise Floor

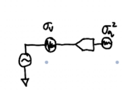
$\sigma_v = 10mV = 0.01V$

FT Freq. Resolution: 1kHz

$\sigma_q^2 = \frac{LSB^2}{12} = 0.00833\ V^2$

$\sigma_{v,tot} = \sigma_v^2 + \sigma_q^2 = 0.00843\ V^2$

Noise Power Density: $\overline{\sigma_p}^2 = \frac{\sigma_{v,tot}^2}{4R} = \boxed{-43.75\,dBm}$



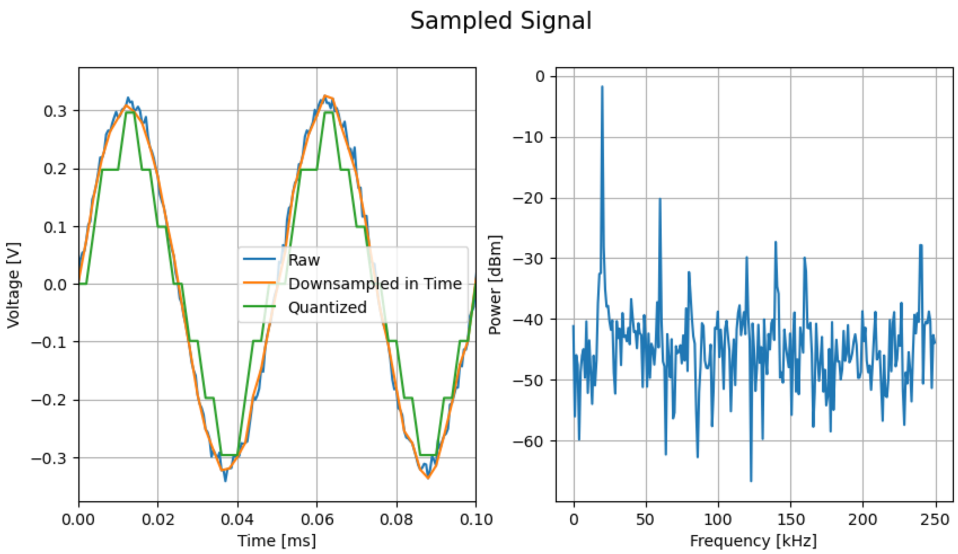
Signal Power Level

Expected: $\boxed{0\,dBm}$

   1. Is the noise floor where you expect? (Again, finding the simulated noise floor is easy by inspection, but we have to go back from total noise to a noise per bin, but we include quantization noise in our total noise this time. Note that quantization noise is white when viewed in the frequency domain, which is a byproduct of it being uncorrelated from one sample to the next.)

a. Yes it is! the expected noise floor was -43 dBm, and the actual noise floor was around -40 dBm

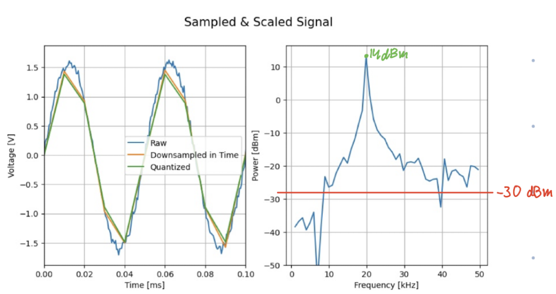
b. it is not clear on this plot, but increasing the ADC sampling frequency extends the range of frequencies that can be captured and shows the true noise floor outside of the widened peak



same plot with 500 kHz sampling frequency to show actual noise floor

2. Does sampling produce tones where you expect?
    a. the sampling produces some tones at higher levels that appear to be either weird harmonics or large noise spikes
3. Is your signal power level effected by decimation and quantization?
    a. yes it is, the signal power is -6.7 dBm

3. Finally, multiply your signal by 5 before quantizing it with the same ADC, representing passing the signal through an impedance matched amplifier with a voltage gain of five.

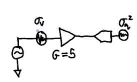


Noise Floor

$\sigma_v = 10mV = 0.01V$

FT Freq. Resolution: 1kHz

$\sigma_q^2 = \frac{LSB^2}{12} = 0.00833\ V^2$



$\sigma_{v,tot} = (5\sigma_v)^2 + \sigma_q^2 = 0.010833 V^2$

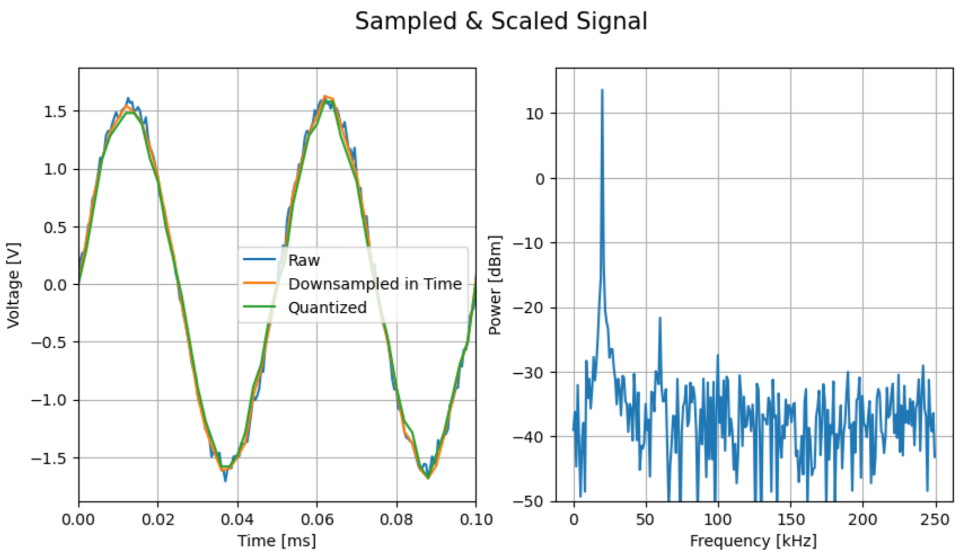
Noise Power Density: $\overline{\sigma_p^2} = \frac{\sigma_{v,tot}^2}{4R} = \boxed{-42.66 dBm}$

Signal Power Level

Expected: $1mW \rightarrow$ [G=5] $\rightarrow 1mW \cdot G^2 = 25mW = \boxed{14\ dBm}$

Voltage Gain

1. Is the noise floor where you expect?
    a. The noise floor is higher than expected at -38 dBm vs -42 dBm. As before, this is more clearly visible when sampling the same signal at 500 MHz.



2. Is the signal level where you expect?
    a. The signal power is at the expected location of 14 dBm!

```
# %%
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.default_rng(seed=42)

# %matplotlib widget
# %%
def fft(x, fs):
    N = x.shape[0]

    X = np.fft.rfft(x)

    freq = np.arange(N / 2) / (float(N) / fs)

    return X, freq

def sample_signal(x, fs_original, fs, bits, scale):
    # Quantize in time
    ratio = int(fs_original/fs)
    x = x[::ratio]

    # Quantize in voltage
    normalized = (x / scale)
    digital = (normalized * 2**bits).astype(int)

    return (digital / 2**bits) * scale
```

```
# %%
f = 20e3 # 20 kHz
fs = 2e6 # 2 MHz
cycles = 20


# https://markimicrowave.com/tools/power-to-voltage.pdf
R = 50 # ohms
Vrms = np.sqrt(1e-3 * R) # V = sqrt(PR)
Vp = Vrms*np.sqrt(2)
# %%
t = np.arange(0, cycles * 1/f, 1/fs)
t_orig = t
raw_samples = Vp*np.sin(2*np.pi*f*t)

noise = rng.normal(0, 10e-3, t.shape[0])
samples = raw_samples + noise

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].plot(t*1e3, samples)
axs[0].set_xlabel("Time [ms]")
axs[0].set_ylabel("Voltage [V]")
axs[0].grid()

X, freq = fft(samples, fs)
P = 2 * np.power(np.abs(X/samples.shape[0]),2)/R/1e-3

axs[1].plot(freq/1e3, 10*np.log10(P))
axs[1].grid()
axs[1].set_xlabel("Frequency [kHz]")
axs[1].set_ylabel("Power [dBm]")

plt.suptitle("Original Signal", fontsize=15)
plt.show()
# %%
# signal = np.sin(np.linspace(0, 2*np.pi, 1000))
signal = np.linspace(-1,1,1000)
plt.plot(signal, label="Original")
plt.plot(sample_signal(signal, 1, 1, 4, 5), label="Sampled")
plt.plot(sample_signal(signal*5, 1, 1, 4, 5), label="Sampled with Gain of 5")
plt.xlabel("Sample Number")
plt.ylabel("Voltage")
plt.grid()
plt.show()
# %%
new_fs = 100e3
t = np.arange(0, cycles * 1/f, 1/new_fs)
```

```
adc_sampled = sample_signal(samples, fs, new_fs, 4, Vp*5)

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].plot(t_orig*1e3, samples, label="Raw")
axs[0].plot(t*1e3, samples[::int(fs/new_fs)][:-1], label="Downsampled in Time")
axs[0].plot(t*1e3, adc_sampled[:-1], label="Quantized")
axs[0].set_xlabel("Time [ms]")
axs[0].set_ylabel("Voltage [V]")
axs[0].grid()
axs[0].set_xlim(0, 0.1)
axs[0].legend()

X, freq = fft(adc_sampled, new_fs)
P = 2 * np.power(np.abs(X/adc_sampled.shape[0]),2)/R/1e-3

axs[1].plot(freq/1e3, 10*np.log10(P))
axs[1].grid()
axs[1].set_xlabel("Frequency [kHz]")
axs[1].set_ylabel("Power [dBm]")

plt.suptitle("Sampled Signal", fontsize=15)
plt.show()
# %%
new_fs = 100e3
t = np.arange(0, cycles * 1/f, 1/new_fs)
adc_sampled = sample_signal(samples*5, fs, new_fs, 4, Vp*5)

fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].plot(t_orig*1e3, samples*5, label="Raw")
axs[0].plot(t*1e3, samples[::int(fs/new_fs)][:-1]*5, label="Downsampled in Time")
axs[0].plot(t*1e3, adc_sampled[:-1], label="Quantized")
axs[0].set_xlabel("Time [ms]")
axs[0].set_ylabel("Voltage [V]")
axs[0].grid()
axs[0].set_xlim(0, 0.1)
axs[0].legend()

X, freq = fft(adc_sampled, new_fs)
P = 2 * np.power(np.abs(X/adc_sampled.shape[0]),2)/R/1e-3

axs[1].plot(freq/1e3, 10*np.log10(P))
axs[1].grid()
axs[1].set_xlabel("Frequency [kHz]")
axs[1].set_ylabel("Power [dBm]")
axs[1].set_ylim(-50)

plt.suptitle("Sampled & Scaled Signal", fontsize=15)
```

```
plt.show()
# %%
```

Lab Notebook

This lab doesn't require any circuit simulations. Instead of comparing measurement, simulation and analysis for each problem, just compare measurement and analysis. We will use the Keysight N9320B Spectrum Analyzer, the Keysight E4438C Signal Generator, the HP 8665B signal generator, the ZN2PD2-63-S+ power splitter and the ZX60-2531M-S+ amplifier (possibly among others), plus assorted attenuators. Datasheets below:

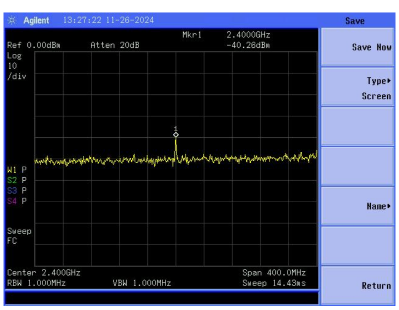
- https://www.keysight.com/us/en/assets/7018-01039/data-sheets-archived/5988-4039.pdf
- https://www.keysight.com/us/en/product/8665B/highperformance-signal-generator-6-ghz.html
- https://www.minicircuits.com/pdfs/ZN2PD2-63-S+.pdf
- https://www.minicircuits.com/pdfs/ZX60-2531M-S+.pdf

@November 25, 2024

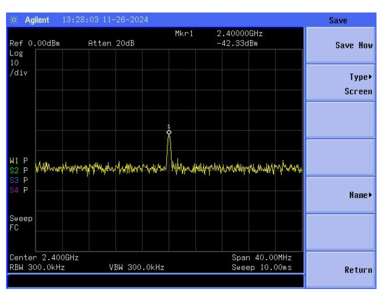
1. Noise performance of the spectrum analyzer

> Attach the output of the vector signal generator to the input of the spectrum analyzer. Set the center frequency to 2.4GHz and the span to 400MHz on the spectrum analyzer. Set the signal generator to produce a -40dBm signal. Observe the signal and make note of the resolution and video bandwidth of the spectrum analyzer. Repeat for a span of 40MHz and a span of 4MHz. allowing the resolution bandwidth to auto-adjust as the span changes. Using these measurements, determine the noise temperature of the spectrum analyzer. Assume the input signal has no noise.

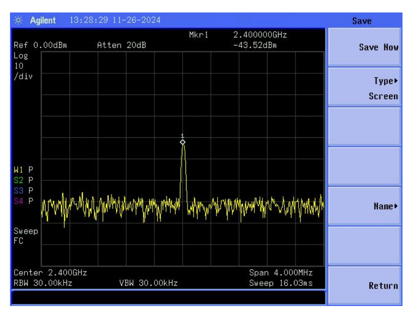
Connected the E4438C Signal Generator directly to the N9320B Spectrum Analyzer using an sma cable and tested each of the spans above. Set signal generator to generate a -40 dBm signal at 2.4 GHz.



400 MHz



40 MHz



4 MHz

Got the following data

Span	RBW	VBW	Sweep Time [ms]	Avg Noise Level [dBm]	Avg Noise Level [W]
400 MHz	1 MHz	1 MHz	14.43	-50	1.00E-07

40 MHz	300 kHz	300 kHz	10	-60	1.00E-08
4 MHz	30 kHz	30 kHz	26.03	-70	1.00E-09

The noise temperature of a device is related to the noise power and bandwidth with the following equation (k is the Boltzmann constant):

$$T_n = \frac{P_n}{k\Delta f}$$

Calculating T_n for each measurement we get T_n = 7.25e9 K for 400 MHz, 2.42e9 K for 40 MHz, and 2.42e9 K for 4 MHz. All of these values are within a few orders of magnitude of each other, and of the expected DANL which can change by several orders of magnitude depending on the input frequency and other settings.

Averaging the temperatures together, we get 4.03e9 kelvin. This is within margin of error of the expected noise temperature of 724e6 (with a DANL of -110 dBm). Because the temperatures are linear and the DANL is log, the reasonable margin of error for noise temperature is much larger.

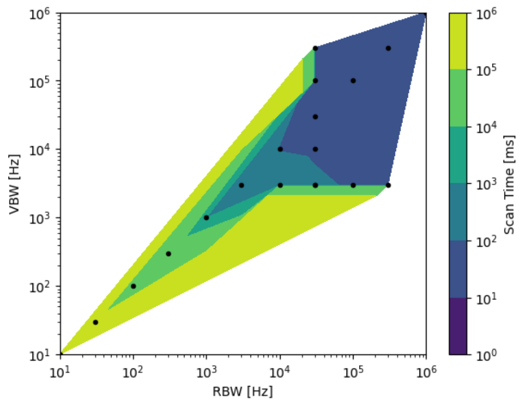
Keep the span constant and adjust the resolution and video bandwidth. Find a qualitative relationship between these bandwidths and the sweep time. Adjust the ratio of video bandwidth to resolution bandwidth and comment on the effects on the trace.

kept the same setup as before, just adjusted RBW and VBW to different values and recorded sweep time:

Span	RBW	VBW	Sweep Time [ms]
400 MHz	1 MHz	1 MHz	14.43
40 MHz	300 kHz	300 kHz	10
4 MHz	30 kHz	30 kHz	26.03
4 MHz	10 kHz	10 kHz	85.12
4 MHz	10 Hz	10 Hz	321600
4 MHz	30 Hz	30 Hz	110900
4 MHz	100 Hz	100 Hz	36130
4 MHz	300 Hz	300 Hz	14000
4 MHz	1 kHz	1 kHz	550
4 MHz	3 kHz	3 kHz	215.4
4 MHz	30 kHz	30 kHz	16.03
4 MHz	100 kHz	100 kHz	10
4 MHz	300 kHz	300 kHz	10
4 MHz	1 MHz	1 MHz	10
4 MHz	30 kHz	100 kHz	16.03
4 MHz	30 kHz	300 kHz	16.03
4 MHz	30 kHz	10 kHz	42.76
4 MHz	30 kHz	3 kHz	141.6
4 MHz	10 kHz	3 kHz	227.1

Span	RBW	VBW	Sweep Time [ms]
4 MHz	30 kHz	3 kHz	141.6
4 MHz	100 kHz	3 kHz	53.11
4 MHz	300 kHz	3 kHz	31.87

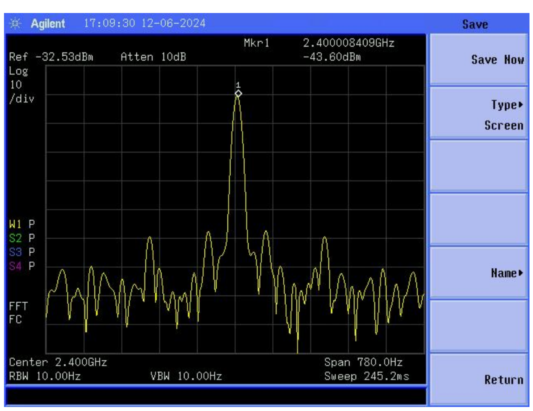
Plotting this we get the following contour plot



This combined with the data above makes it clear that the scan time increases as both RBW and VBW decrease. The more smaller chunks need to be sampled (RBW) or more aggressive the low pass filter (VBW) the longer the sampling process takes. When RBW and VBW are far apart, whichever is larger "takes over" and drives the increase in sampling time.

> Adjust the span and the resolution bandwidth such that the resolution bandwidth is 10Hz and the sweep time is reasonable. What is the measured signal power and why?

We kept the same setup as the previous two parts, and took data with span of 780 hz:



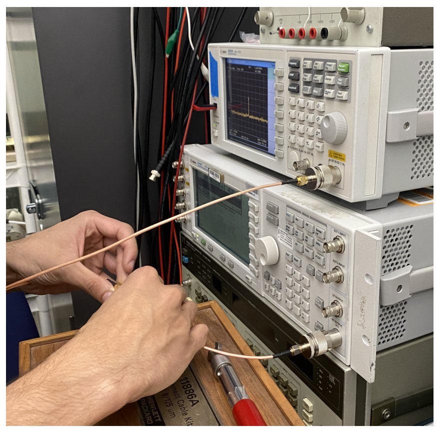
Because of the small span, we are experiencing bin splitting (the bandwidth of the signal from the signal generator is wider than the bin width, so the power of the signal is split across multiple bins and the peak height is decreased).

Previous measurements we took (such as with 400/40/4 MHz spans) had a signal power of -40-42 dBm, however here we see -43.6 dBm demonstrating bin splitting.

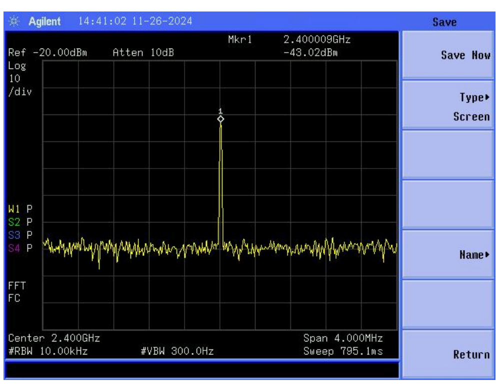
2. Noise in cascaded elements

> Set the resolution bandwidth of the spectrum analyzer to 10kHz. Add 30dB of attenuation in series between the function generator and the spectrum analyzer. Can you observe the attenuation's contribution to the noise floor? What is the noise temperature of the attenuator and how does it compare to the noise temperature of the spectrum analyzer?

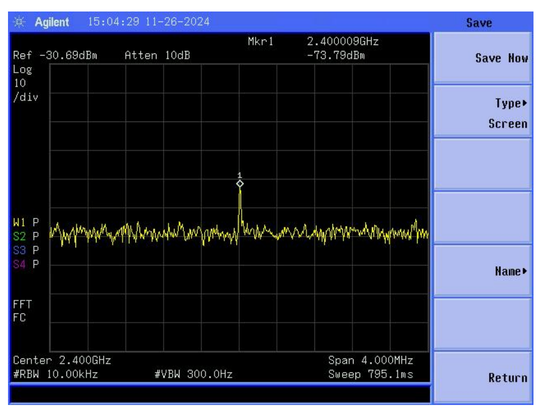
Kept same setup from part 1.



added 20 dB attenuator (MiniCircuits VAT-20+) and then 10 dB attenuator (MiniCircuits VAT-10+) between the signal generator and spectrum analyzer



No attenuation, same as part 1



20 dB + 10 dB = 30 dB Attenuation

Adding 30 dB of attenuation has no effect on the noise floor, because the noise temperature of the attenuator combination (300e3 K from theory question 3) is significantly less than the noise temperature of the spectrum analyzer (724e6 K from theory question 1).

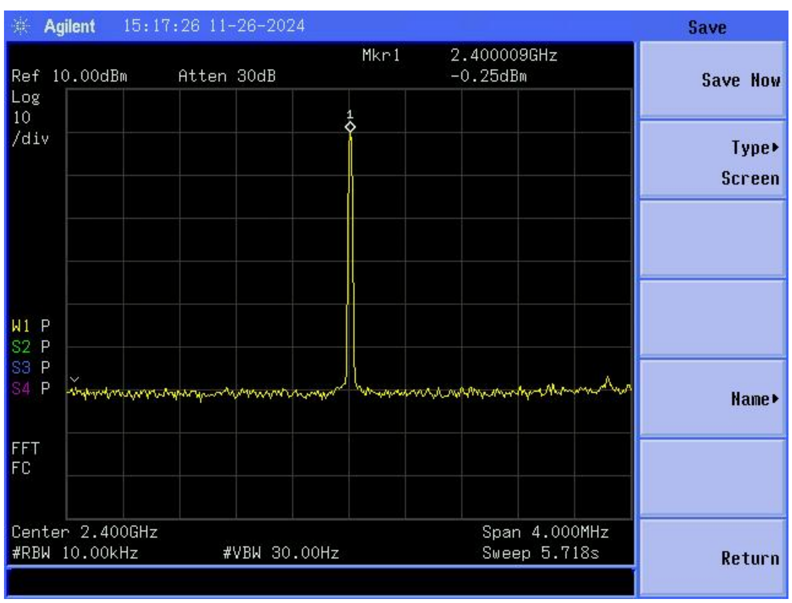
Add a 60dB of amplification after the 30 dB of attenuation. Can you observe the attenuator's contribution to the noise floor? Justify with calculations and measurements comparing your amplified resistor noise to the spectrum analyzer noise. Use interstage attenuators (of ~1dB) to keep your amplifiers stable.

modified the setup from previous part to add two 30 dB amplifiers (MiniCircuits ZX60-2531MA+) after the 30 dB attenuation. We also added a 1 dB attenuator (MiniCircuits VAT-1+) between the two amplifiers to help with stability.

We kept track of the noise at each stage using a spreadsheet available at link and copied below:

Stage	Description	Signal Power [dBm]	Stage Noise Contribution [K]	Total Noise Temperature [K]	Gain	Insertion Loss / Noise Figure [dB]
1	Input	-40	0	0.00E+00	0	
2	30 dB Attenuator	-70	299700	3.00E+05	0.001	
3	30 dB Amplifier	-33.09	431.3432455	1.47E+09	4909.1	3.87
4	1 dB Attenuator	-34.09	77.67762354	1.17E+09	0.7943	
5	30 dB Amplifier	2.82	431.3432455	5.75E+12	4909.1	3.87

This predicts a noise temperature of 1.4e10 K, which results in a noise power of -61 dBm.



The noise floor in this spectrum is around -60 dBm, close to what was predicted by the spreadsheet. Comparing the noise values we get 5.75e12 K for the amplified resistor noise to 724e6 K for the spectrum analyzer noise. The amplified resistor large is several orders of magnitude larger than the spectrum analyzer noise, and thus changes the noise floor from -90 dBm to -60 dBm.

> Replace the attenuator between your first two amplifiers with a band pass filter and discuss the measured spectrum.

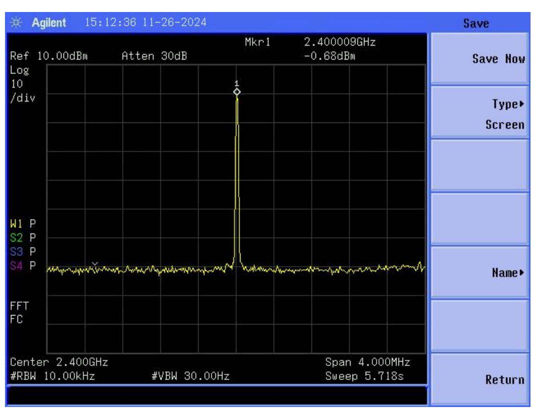
Kept the same setup as previous part.

Replaced the 1 dB attenuator with a MiniCircuits VBFZ-2340-S+ bandpass filter (passband 2020 MHz - 2660 MHz, loss < 2.2dB). At 2.4 GHz the filter has an insertion loss of around 1.6 dB.

Because the RBW filter inside of the spectrum analyzer is narrower than this filter, we don't expect to see a significant change in the measured spectrum. The change in attenuation from 1 dB to 1.6 dB is expected to have a bigger effect than the existence of a bandpass filter. Updated analysis is below:

Stage	Description	Signal Power [dBm]	Stage Noise Contribution [K]	Total Noise Temperature [K]	Gain	Insertion Loss / Noise Figure [dB]
1	Input	-40	0	0.00E+00	0	
2	30 dB Attenuator	-70	299700	3.00E+05	0.001	
3	30 dB Amplifier	-33.09	431.3432455	1.47E+09	4909.1	3.87
4	Bandpass Filter	-34.69	133.6319312	1.02E+09	0.6918	
5	30 dB Amplifier	2.22	431.3432455	5.00E+12	4909.1	3.87

Again converting the noise temperature of 5.00e12 K to dBm, we get -61.6 dBm, which is slightly lower than with the 1 dB attenuator.



As expected the spectrum does not change significantly. With the 1 dB attenuator, the noise floor was around -60 dBm. This looks about the same, maybe a bit lower. The predicted difference is only around -61 vs -61.6 →

difference 0.6 dBm which would be hard to see at this scale.

Again comparing the noise temperatures, we get 5.00e12 K for the amplified resistor noise to 724e6 K for the spectrum analyzer noise. As with the previous part, the amplified resistor noise is significantly larger than spec an noise and raises the noise floor.

3. Linearity of Amplifiers

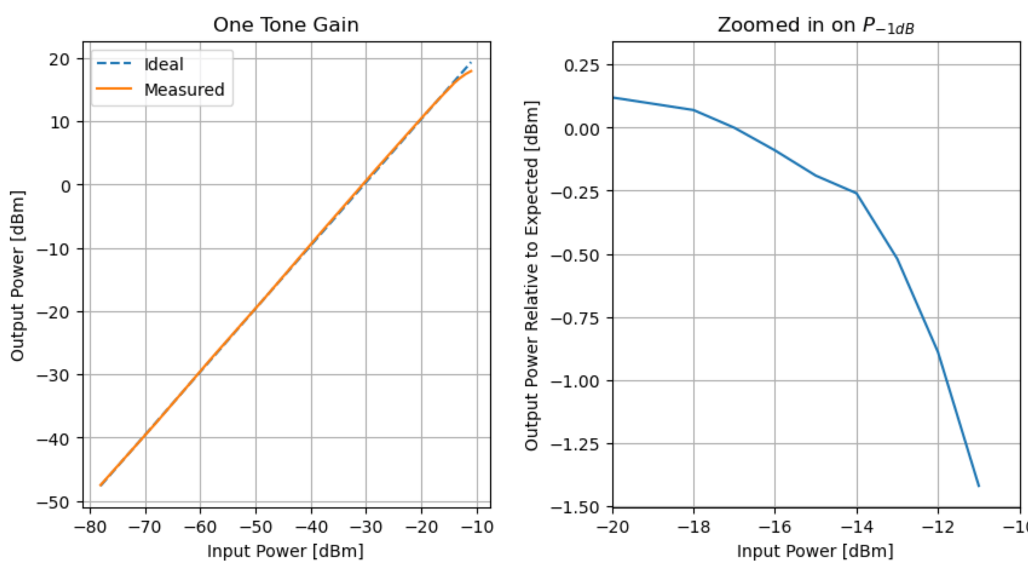
> Perform one tone nonlinearity tests on the ZX60-2531M-S+ amplifier at a frequency of 600MHz and use these tests to determine the amplifier's P-1dB. Compare your measured value to the datasheet.

Electrical Setup:

Agilent E4438C → ZX60-2531M-S+ Amplifier → 10-30 dB Attenuation (depending on output power) → Agilent N9320B Spectrum Analyzer

While not necessarily strictly necessary, the additional attenuation helps limit the amount of power going into the spectrum analyzer and protect it, especially at higher power levels. It is compensated for in all data provided in the report, and the excel file contains notes on which attenuators were used where.

Measuring the output power at several gains, we can see the gain flatten out at around -12 dBm.



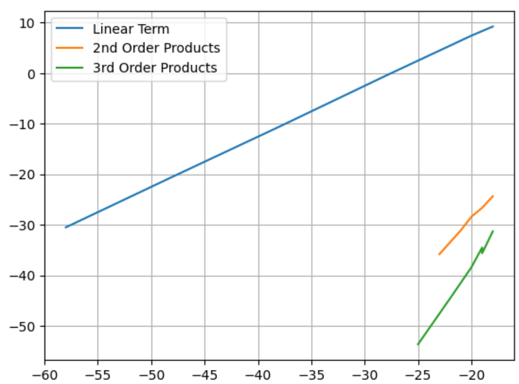
Specifically P-1dB is at -12.3 dBm, which results in an output power of 18.08 dBm. This is close to the datasheet's expected value at 600 MHz, which is 19 dBm.

> Power splitters can be used backwards as power combiners. Use a power splitter to perform two tone non-linearity tests on ZX60-2531M-S+ amplifier using two tones of P-1dB - 10dB at 600MHz and 601MHz. Vary the input power and plot the power of the intermodulation tones. Use this plot to calculate the IIP2 and IIP3. Compare these values to the datasheet and determine if IIP3 has the expected relationship to P-1dB.

kept the same electrical setup as before: signal generator → amplifier → 30 dB attenuation → spectrum analyzer

set signal generator to multi tone mode with one tone at 600 MHz and one at 601 MHz, with equal dB to create tones at 600, 601, 1200, 1201, 1202, 1800, 1801, 1802, and 1803 MHz.

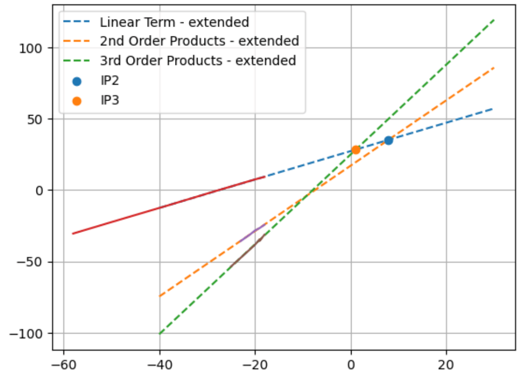
It is not possible to actually measure IIP2 or IIP3 because the amplifier cannot handle that much power, but we can estimate it by collecting data at lower powers and running a linear regression.



Running a regression we get the following values. Note that the expected slopes are 1, 2, and 3 respectively. The overestimate of the slopes is likely due to the significant amount of noise, making it hard to measure OIP2 and OIP3.

```
+-----------+---------+-------------+
| Term      |   Slope |   Intercept |
+===========+=========+=============+
| Linear    |    1    |       27.24 |
+-----------+---------+-------------+
| 2nd Order |    2.29 |       17.01 |
+-----------+---------+-------------+
| 3rd Order |    3.15 |       24.85 |
+-----------+---------+-------------+
```

Calculating the intercept between the linear, 2nd, and 3rd order terms, we find:



IIP2 = 7.90 dBm, OIP2 = 35.10 dBm

IIP3 = 1.11 dBm, OIP3 = 28.35 dBm

The expected OIP3 from the datasheet is 29.6 dBm, matching reasonably closely with the expected value. From theory, IIP3 = P-1dB + 9.6 dB. We calculated P-1dB = -12 dBm, resulting in an expected IIP3 of -2.4 dBm. This matches the expected IIP3 from the datasheet reasonably closely.

IIP2 and OIP2 are not listed on the datasheet, so we can't compare against them. However one sanity check is that the IM2 slope is close to 2, and IIP2 is greater than OIP2 as expected.

Discrepancies in these predicted IM3 values are likely due to inaccuracies in measuring the slope which is an overestimate at 3.15 instead of 3 due to the difficulty taking measurements when peaks are close to the noise floor.

Reduce the power of one tone by 3dB and record the resulting spectrum. Compare it to theory and label which tones are the result of HD2, HD3, IM2 and IM3.

Assuming an input of two tones following

$$V_{in}(t) = V_1 \cos(\omega_1 t) + V_2 \cos(\omega_2 t)$$

We get a combined output of

$$\begin{aligned}
V_{out}(t) = {} & a_1 (V_1 \cos(\omega_1 t) + V_2 \cos(\omega_2 t)) \\
& + \frac{a_2}{2} (V_1^2 (1 + \cos(2\omega_1 t)) + 2V_1 V_2 \left[\cos(\omega_1 + \omega_2)t + \cos(\omega_1 - \omega_2)t\right] + V_2^2 (1 + \cos(2\omega_2 t))) \\
& + \frac{a_3}{4} \left[V_1^3 (\cos(3\omega_1 t) + 3\cos(\omega_1 t) + V_2^3 (\cos(3\omega_2 t) + 3\cos(\omega_2 t)) \right. \\
& \quad + 3V_1^2 V_2 (2\cos\omega_2 t + \cos(2\omega_1 + \omega_2)t + \cos(2\omega_1 - \omega_2)t) \\
& \quad \left. + 3V_1 V_2^2 (2\cos\omega_1 t + \cos(2\omega_2 + \omega_1)t + \cos(2\omega_2 - \omega_1)t) \right]
\end{aligned}$$

If one tone is half of the other, than $V_1 = 2V_2$ or $V_2 = 2V_1$. For this problem we will assume that $V_2 = 2V_1$ and $\omega_2 = 601\ \text{MHz}$.

Because $V_1 \neq V_2$ the 2nd order harmonics and 3rd order harmonics and intermodulation products will not have symmetric heights.

To find the **worst case** (larger harmonic compared to smaller input) HD2, HD3, IM2, and IM3, we can do the following. Define

$$V_{in,max} = \max(V_1, V_2)$$
$$V_{in,min} = \min(V_1, V_2)$$

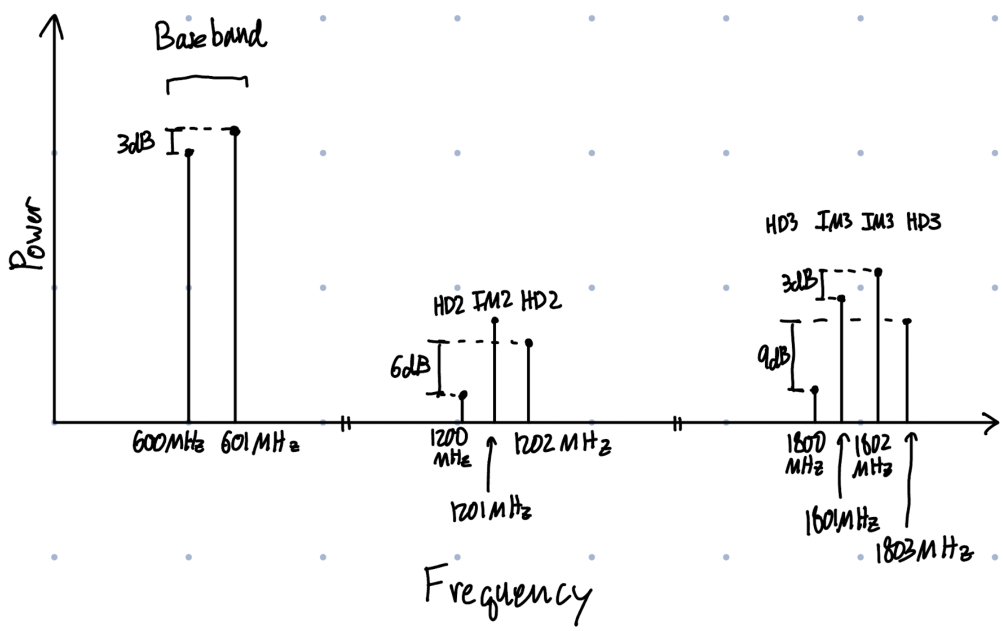
Then

$$HD2 = \frac{1}{2}\frac{a_2}{a_1}V_{in,max} \qquad HD3 = \frac{1}{2}\frac{a_3}{a_1}\frac{V_{in,max}^3}{V_{in,min}}$$

Using similar logic define

$$IM2 = \frac{a_2}{a_1}V_{in,max} \qquad IM3 = \frac{3}{4}\frac{a_3}{a_1}\frac{V_{in,max}^2 V_{in,min}}{V_{in,min}}$$

Working through this same math for all combinations of terms that create different HD2, IM2, HD3, and IM3, we get the following plot:

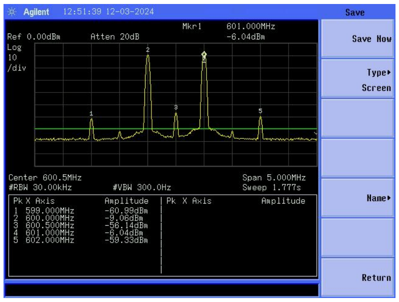


The relative power between tones is calculated assuming that $V_{in,max} = 2V_{in,min}$ and $1/2 \rightarrow -3$ dB.

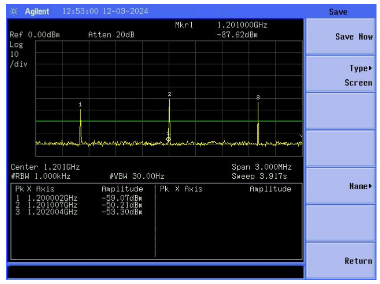
To find the relative height between HD2 and IM2 we can see that each of the HD2 harmonics is 3 dB greater than or less than a "normal" HD2 where the tones have the same power (in the "normal" case IM2 = HD2 + 6 dB). This means that IM2 should be 3 dB greater than the larger tone, and and 9 dB greater than the smaller one.

For the physical setup, kept the same as the previous part, just adjusted multi-tone mode on the signal generator to increase the power of the 601 MHz tone by 3 dB. No attenuation was used.

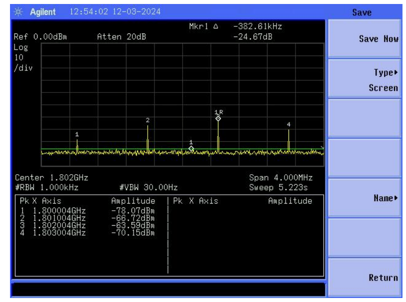
Collecting spectra around each relevant range, we get:



Baseband Signal (600 MHz)



Second Harmonic (1200 MHz)



Third Harmonic (1800 MHz)

Combined table representation of the above images

Frequency [MHz]	Peak Label	Amplitude [dBm]
600	Baseband	-9.1
601	Baseband	-6.1
1200	HD2	-59.0
1201	IM2	-50.2
1202	HD2	-53.3
1800	HD3	-78.1
1801	IM3	-66.7
1802	IM3	-63.6
1803	HD3	-70.2

From the table we can see that the relative peak heights of the harmonics match what is expected from theory almost exactly!!

Add a second amplifier in cascade with the first with a 1dB attenuator between them. Measure the cascaded IIP3 and compare to calculations.

Reverted spectrum analyzer to setup used for measuring IP2 and IP3 (tones have same power). Added another amplifier in series with the first and 1 dB attenuator between them. Also added 30 dB of attenuation at the end to keep the output power in a reasonable range.

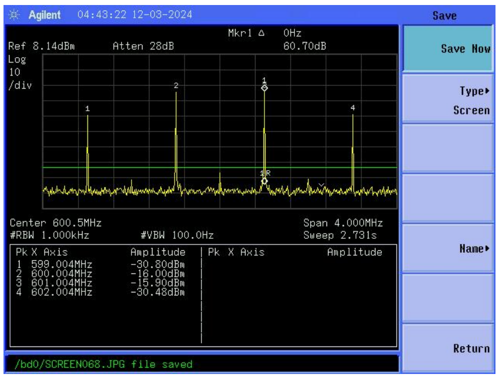
Kept track of link budget and intermodulation terms using a spreadsheet (available here) and copied below:

Stage	Description	600.000E+6 (Baseband)	601.000E+6 (Baseband)	1.801E+09 (IM3)	1.802E+09 (IM3)
0	input	-50	-50	0	0
1	amplifier	-17.5	-17.5	-112.8	-112.8

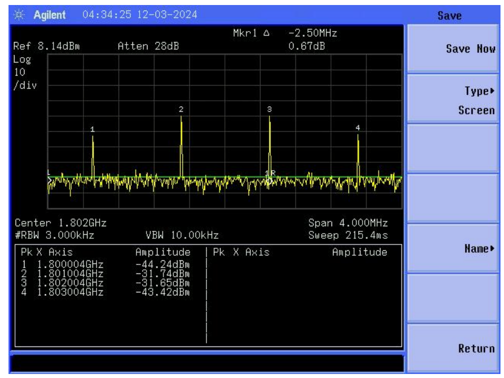
2	1db attenuator	-18.5	-18.5	-113.8	-113.8
3	amplifier	14	14	-18.3	-18.3

Because the final IM3 term is the sum (in linear space) of the IM3 from the first and second amplifiers, it gets converted to a max in log space. The input to the second amplifier is significantly larger than the input to the first, so its IM3 term is much bigger and dominates.

This results in the following updated spectrum (example captured with -50 dBm input power & 30 db attenuation)):



Baseband (600 MHz)

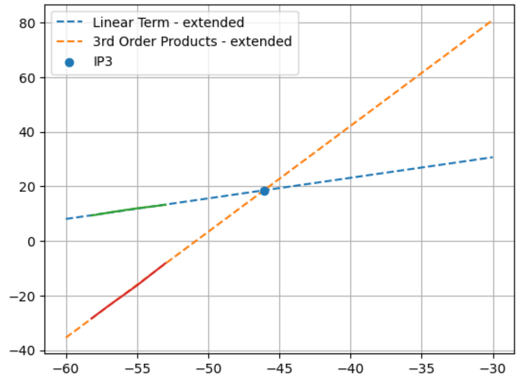


Third Harmonic (1800 MHz)

Frequency [MHz]	Peak Label	Measured Amplitude [dBm]	Predicted Amplitude [dBm]
600	Baseband	14	14
601	Baseband	14.1	14
1800	HD3	-14.2	
1801	IM3	-1.7	-18.3
1802	IM3	-1.7	-18.3
1803	HD3	-13.4	

The predicted amplitudes for IM3 are significantly less than the expected ones (-18.3 dBm vs -1.7 dBm). From talking to Prof Spencer, this is likely due to the outputted power being slightly larger than we expected or an incorrect number for OIP3.

Collecting data at multiple input powers and we can make a full power plot to get




```
+-----------+---------+-------------+
| Term      |   Slope |   Intercept |
+===========+=========+=============+
| Linear    |    0.75 |       53.25 |
+-----------+---------+-------------+
| 3rd Order |    3.88 |      197.22 |
+-----------+---------+-------------+6
```

This gives a cascaded IIP3 of -46.08 dBm and a cascaded OIP3 of 18.54 dBm.

To find the expected cascaded IIP3 from calculations, we can plug in different power values into the spreadsheet and test when the power of the IM3 harmonic is equal to the fundamental. Testing a few different values until the power at 600 MHz is equal to the power at 1801 and 1802 MHz, we get the following

Stage	Description	600.000E+6 (Baseband)	601.000E+6 (Baseband)	1.801E+09 (IM3)	1.802E+09 (IM3)
0	input	-33.5	-33.5	0	0
1	amplifier	-1	-1	-63.3	-63.3
2	1db attenuator	-2	-2	-64.3	-64.3
3	amplifier	30.5	30.5	31.2	31.2

This results in a calculated IIP3 of -33.5 dBm and a calculated OIP3 of 31.2 dBM. These are both much higher than the measured value, again likely due to the same inaccuracies with OIP3 that we saw in the previous part of this problem.