Power Amplifier Testing

by William Harrington

Introduction

The purpose of this document is to examine the output power capability of the power amplifier chosen by the Sputnik team for their module.

Materials

- Agilent E8241A Signal Generator
- Tektronix RSA3308A Spectrum Analyzer
- Tektronix PS2520G Programmable Power Supply
- IR Temperature Gauge
- 4 dB 20 watt attenuator
- Various patch cords and N to SMA adapters
- KW0x Breakout board
- Power Amplifier breakout board

Setup

With Signal Generator

Signal generator was hooked up to input for Power Amplifier. Power Amplifier was powered off programmable power supply. Output of Power Amplifier hooked up to 4dB 20W attenuator which was hooked up to Spectrum Analyzer.

With KW0x

KW0x was hooked up to input for Power Amplifier. Power Amplifier was powered off programmable power supply. Output of Power Amplifier hooked up to 4dB 20W attenuator which was hooked up to Spectrum Analyzer.

Procedure

With Signal Generator

1. Used signal generator for generating input power for power amplifier.
2. Programmable power supply used for powering amplifier also gave measurement of current being pulled by PA when on.
3. Output power of power amplifier measured by spectrum analyzer.
4. Temperature of power amplifier measured by IR temp gauge.

With KW0x

1. Used KW0x breakout board for generating input power for power amplifier.
2. Programmable power supply used for powering amplifier also gave measurement of current being pulled by PA when on.
3. Output power of power amplifier measured by spectrum analyzer

Measurements recorded

- Input Power (dBm)
- Current Consumption of Power Amplifier (Amps)
- Temperature \((\,^{\circ}\mathrm{C})\)
- Output Power of the Power Amplifier (dBm)

In [26]:

```
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline
from IPython.display import set_matplotlib_formats

# Graphing helper function
def setup_graph(title='', x_label='', y_label='', fig_size=None):
    fig = plt.figure()
    if fig_size != None:
        fig.set_size_inches(fig_size[0], fig_size[1])
    ax = fig.add_subplot(111)
    ax.set_title(title)
    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
```

With Signal Generator

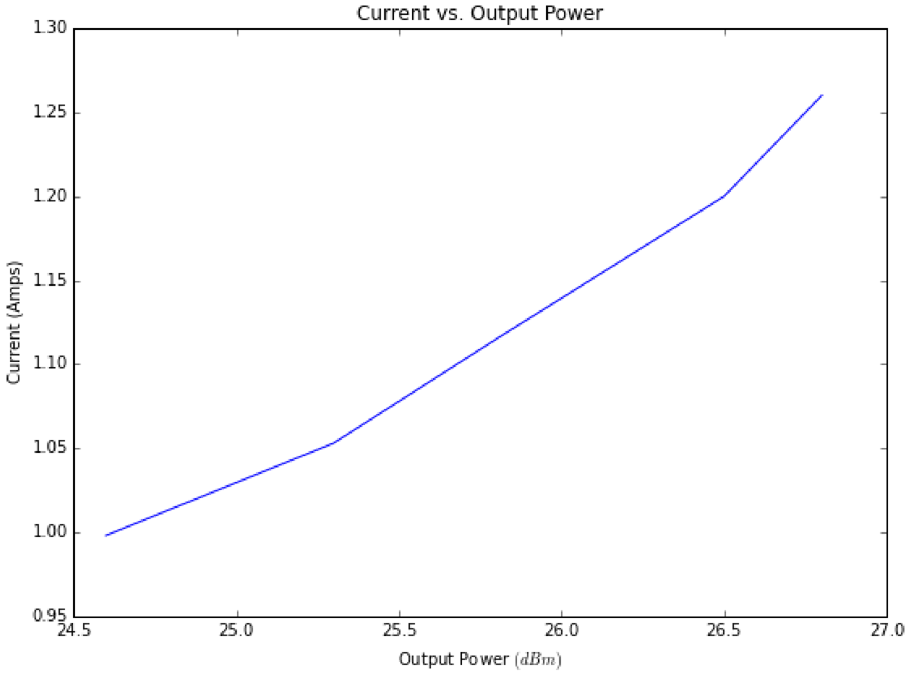
In [27]:

```
# read the data
columns = np.loadtxt('RFPAtest.csv',
                    dtype=float,
                    delimiter=',',
                    unpack=True)

current = columns[0]
temp = columns[1]
input_power = columns[2]
output_power = columns[5]
```

Current

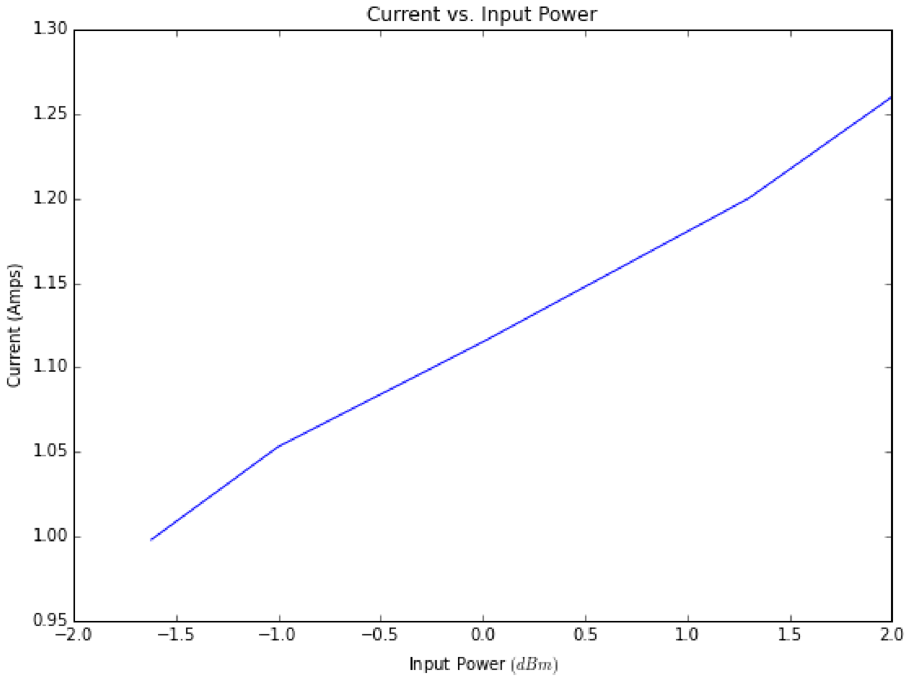
In [28]:

```
setup_graph('Current vs. Output Power',
            r'Output Power $(dBm)$',
            'Current (Amps)',
            (9,6.5))
plt.plot(output_power, current)
plt.show()
```



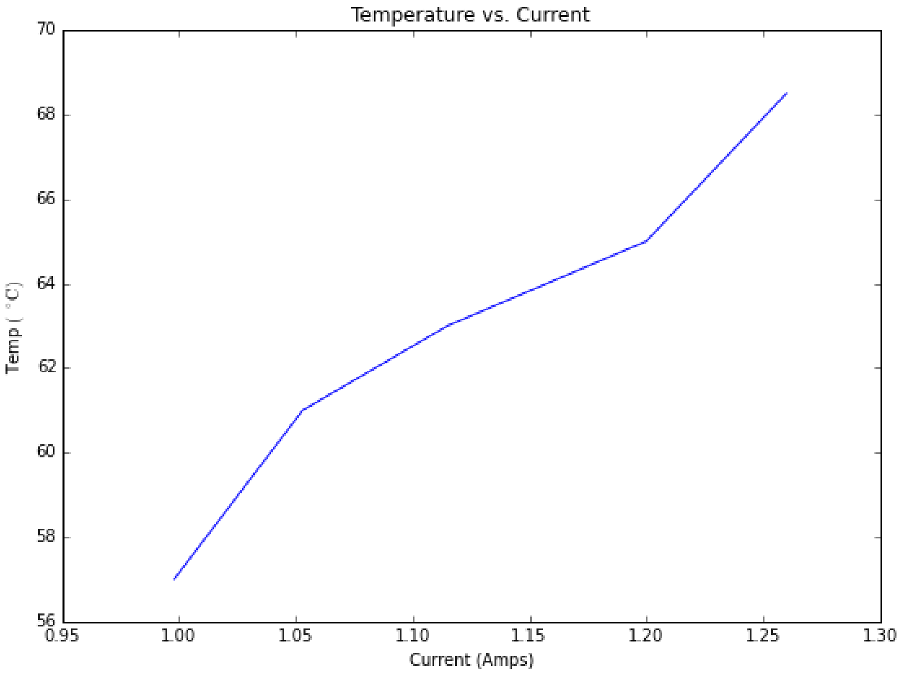
In [29]:

```
setup_graph('Current vs. Input Power',
            r'Input Power $(dBm)$',
            'Current (Amps)',
            (9,6.5))
plt.plot(input_power, current)
plt.show()
```



In [30]:

```
setup_graph('Temperature vs. Current',
            'Current (Amps)',
            r'Temp $(\,^{\circ}\mathrm{C})$',
            (9,6.5))
plt.plot(current, temp)
plt.show()
```



With KW0x

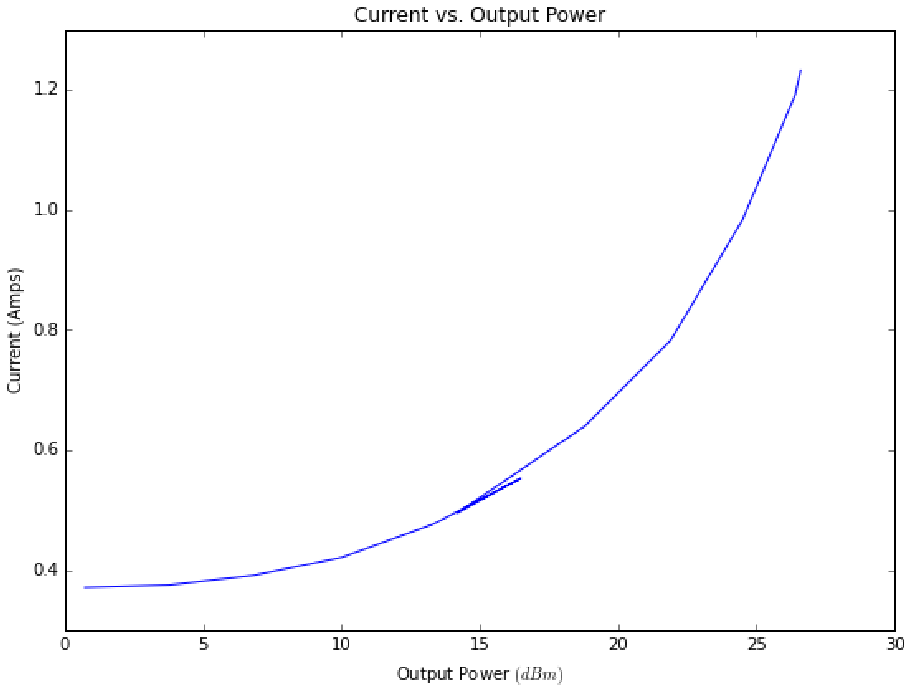
In [31]:

```
# read the data
columns = np.loadtxt('KW0xPAtest.csv',
                    dtype=float,
                    delimiter=',',
                    unpack=True)

current = columns[0]
temp = columns[1]
input_power = columns[2]
output_power = columns[5]
second_harm = columns[6]
```

Current

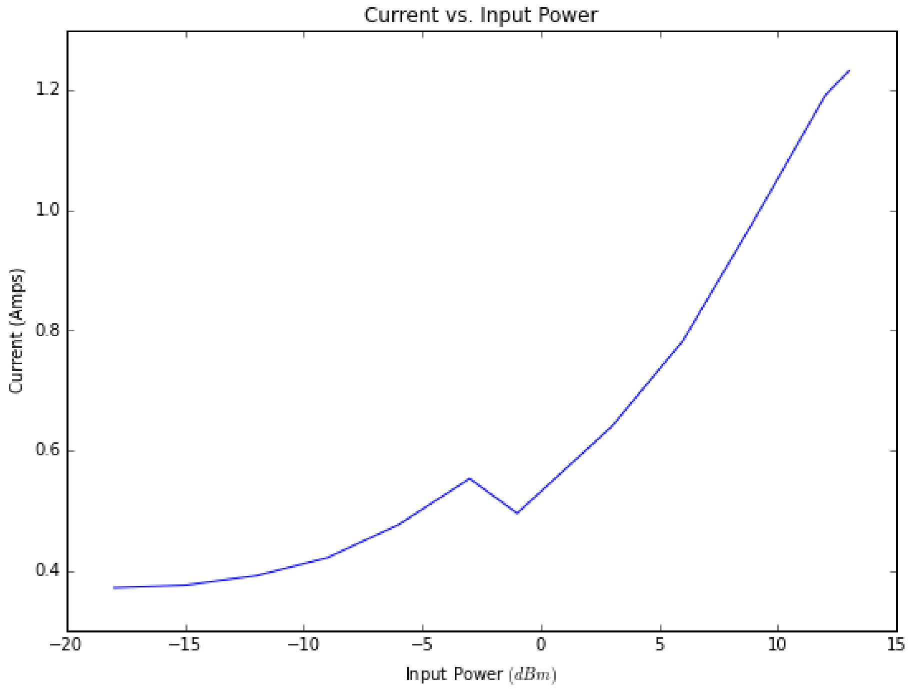
In [32]:

```
setup_graph('Current vs. Output Power',
            r'Output Power $(dBm)$',
            'Current (Amps)',
            (9,6.5))
plt.plot(output_power, current)
plt.show()
```



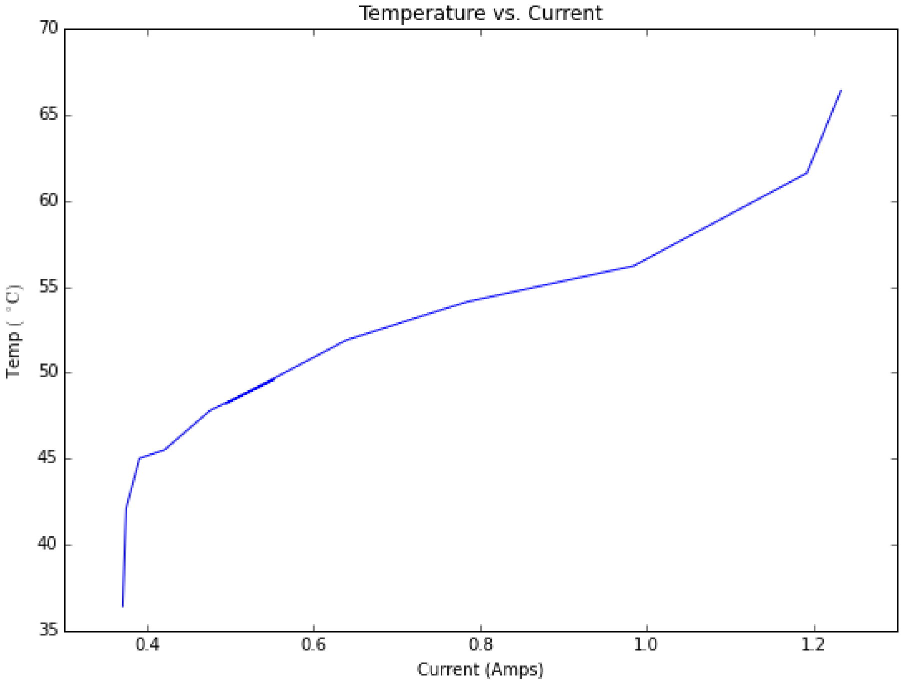
In [33]:

```
setup_graph('Current vs. Input Power',
            r'Input Power $(dBm)$',
            'Current (Amps)',
            (9,6.5))
plt.plot(input_power, current)
plt.show()
```



In [34]:

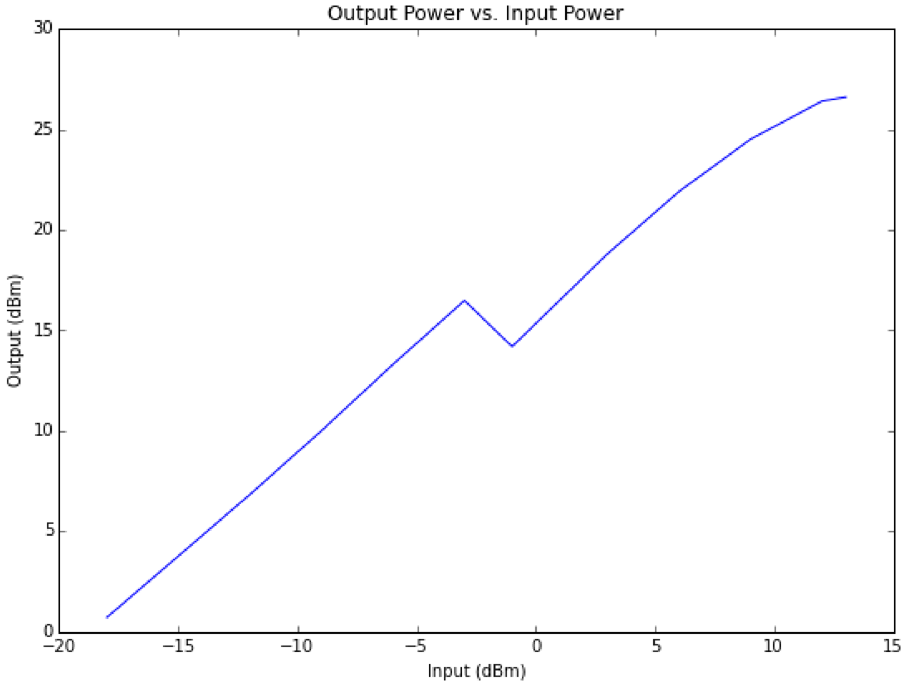
```
setup_graph('Temperature vs. Current',
            'Current (Amps)',
            r'Temp $(\,^{\circ}\mathrm{C})$',
            (9,6.5))
plt.plot(current, temp)
plt.show()
```



Power

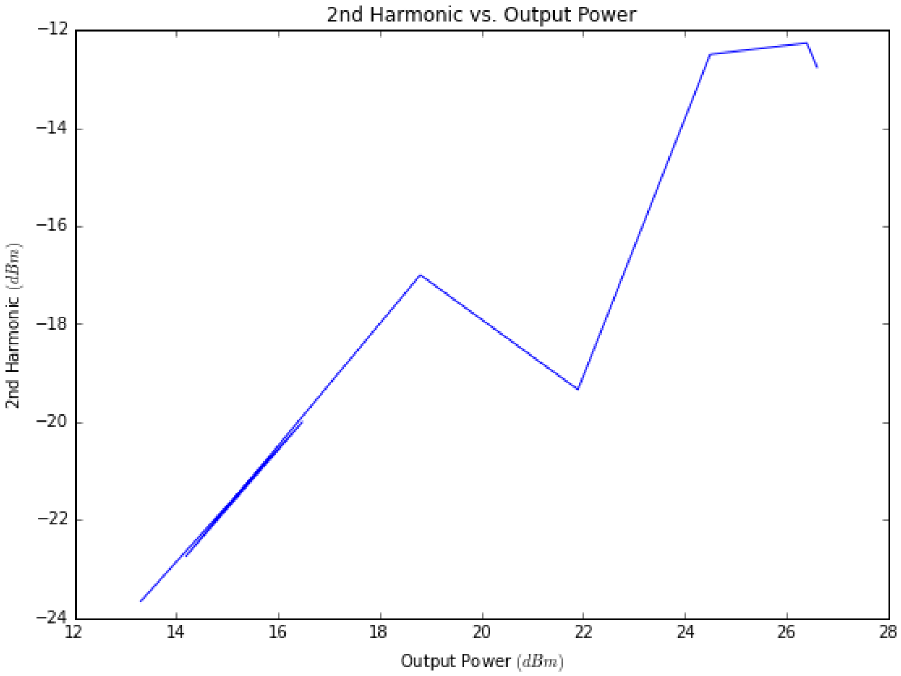
In [35]:

```
setup_graph('Output Power vs. Input Power',
            'Input (dBm)',
            'Output (dBm)',
            (9,6.5))
plt.plot(input_power, output_power)
plt.show()
```



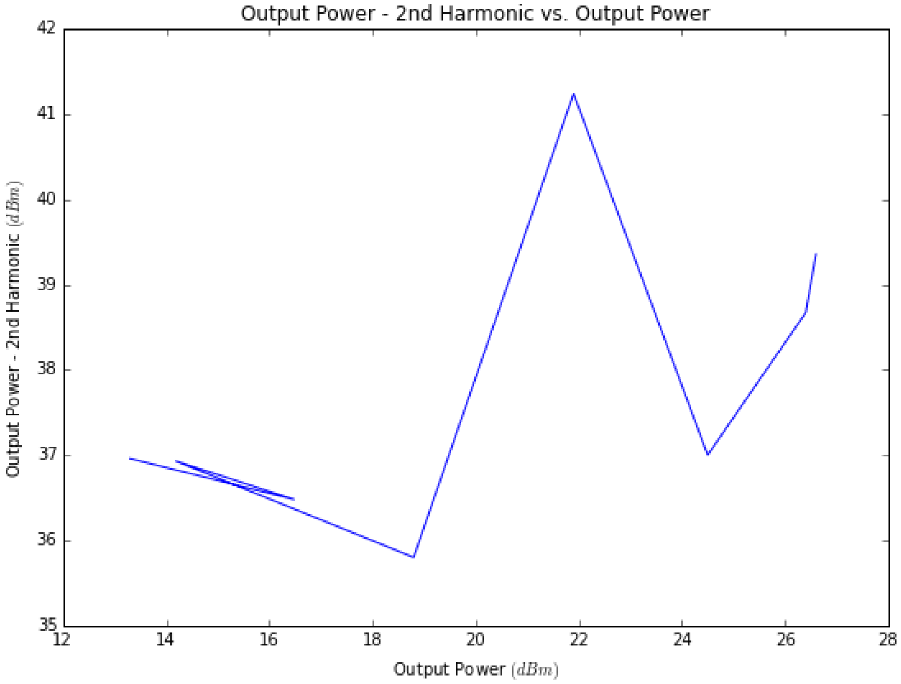
In [36]:

```
# get rid of 0 measurements
second_harm = second_harm[4:]
output_power = output_power[4:]

setup_graph('2nd Harmonic vs. Output Power',
            r'Output Power $(dBm)$',
            r'2nd Harmonic $(dBm)$',
            (9,6.5))
plt.plot(output_power, second_harm)
plt.show()
```



In [37]:

```
delta = []
for i in range(len(output_power)):
    delta.append(output_power[i]-second_harm[i])


setup_graph('Output Power - 2nd Harmonic vs. Output Power',
            r'Output Power $(dBm)$',
            r'Output Power - 2nd Harmonic $(dBm)$',
            (9,6.5))
plt.plot(output_power, delta)
plt.show()
```


Output Power - 2nd Harmonic vs. Output Power
Output Power - 2nd Harmonic (dBm)
Output Power (dBm)
42
41
40
39
38
37
36
35
12
14
16
18
20
22
24
26
28