Temperature Dependence

Josh Wilkins
10/04/17

The objective of this file is to characterize the temperature dependency of the synthesizer. The bandpass filter inside the synthesizer is expected to have the most contribution to this dependency so it will be characterized first.

In [1]:
```
# Imports↔
```

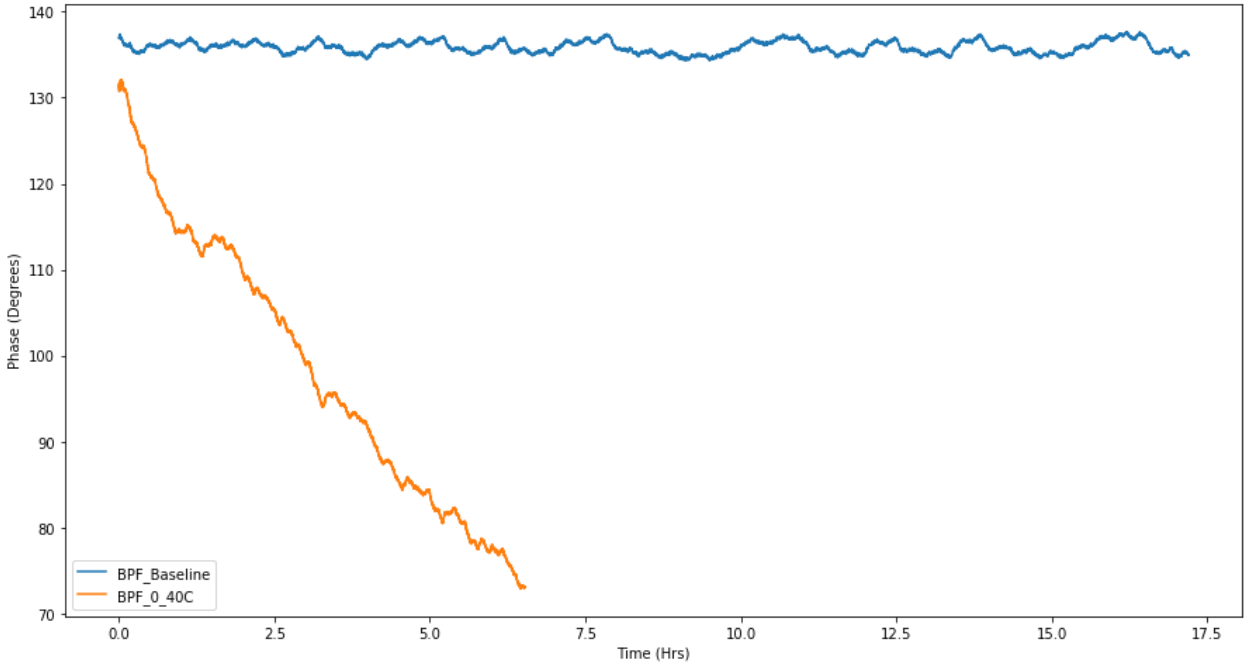
In [2]:
```
# Custom Methods

def phase_plot(file_name):
    # file_name: 'testfile.csv'
    # sample_rate: x Hz

    data = csvData(file_name).getData()
    time = data[:,0]     # In Seconds
    phase = data[:,2]    # In Degrees

    print "Peak to Peak: %.2f°" % (max(phase) - min(phase))

    # Plotting the Phase in degrees vs time in hrs
    plt.plot(time/3600, phase+137, label=(file_name[:-4]))
    plt.xlabel('Time (Hrs)')
    plt.ylabel('Phase (Degrees)')
    plt.grid(which='both')
    plt.legend()

    return phase

def ADev(file_name, phase, sample_rate): # phase in °
    phase /= 360.*6.4e9 # Converting phase to seconds

    # Allan Deviation
    (t2, adev, adev_error, adev_n) = allantools.oadev(phase, rate=sample_rate, taus='all')
    plt.loglog(t2, adev, label=(file_name[:-4]))

    plt.xlabel('Tau')
    plt.ylabel('Allan deviation')
    plt.title('Allan Deviation:')
    plt.grid(which='both')
    plt.legend()
```

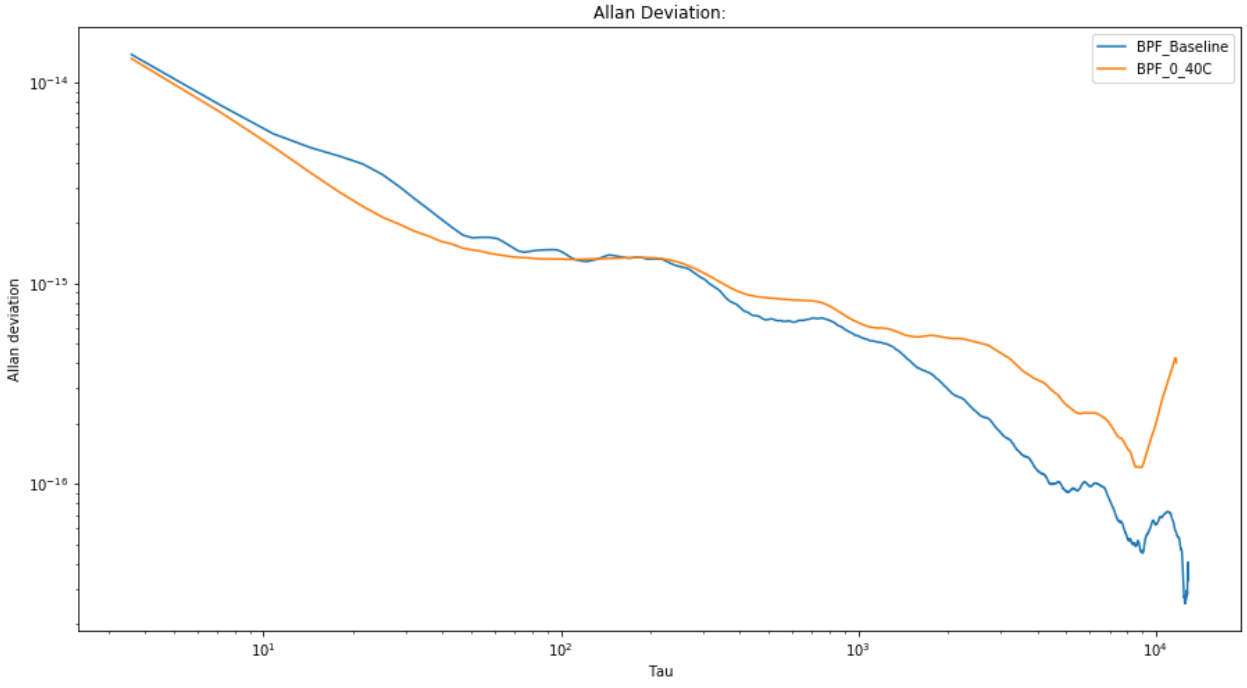
BPF Results

The phase of the BPF was measured over a long span of time and then plotted below. From the peak to peak variation of the phase, an approximate phase to temperature conversion can be obtained. The allan deviation was also plotted to verify results.

In [3]:
```
# BPF Results
Baseline = phase_plot('BPF_Baseline.csv') # BPF Baseline
TempVaried = phase_plot('BPF_0_40C.csv')    # Temperature varied BPF
plt.figure()
ADev('BPF_Baseline.csv', Baseline, 0.278)
ADev('BPF_0_40C.csv', TempVaried, 0.278)
```

```
Peak to Peak: 3.39°
Peak to Peak: 59.15°
```





For the second test, the BPF was put into a temperature chamber and its temperature varied linearly from 0 to 40°C. From the resulting peak to peak variation in the phase of 59.15°, the conversion from phase to temperature can be approximated by about 1.5° per °C.

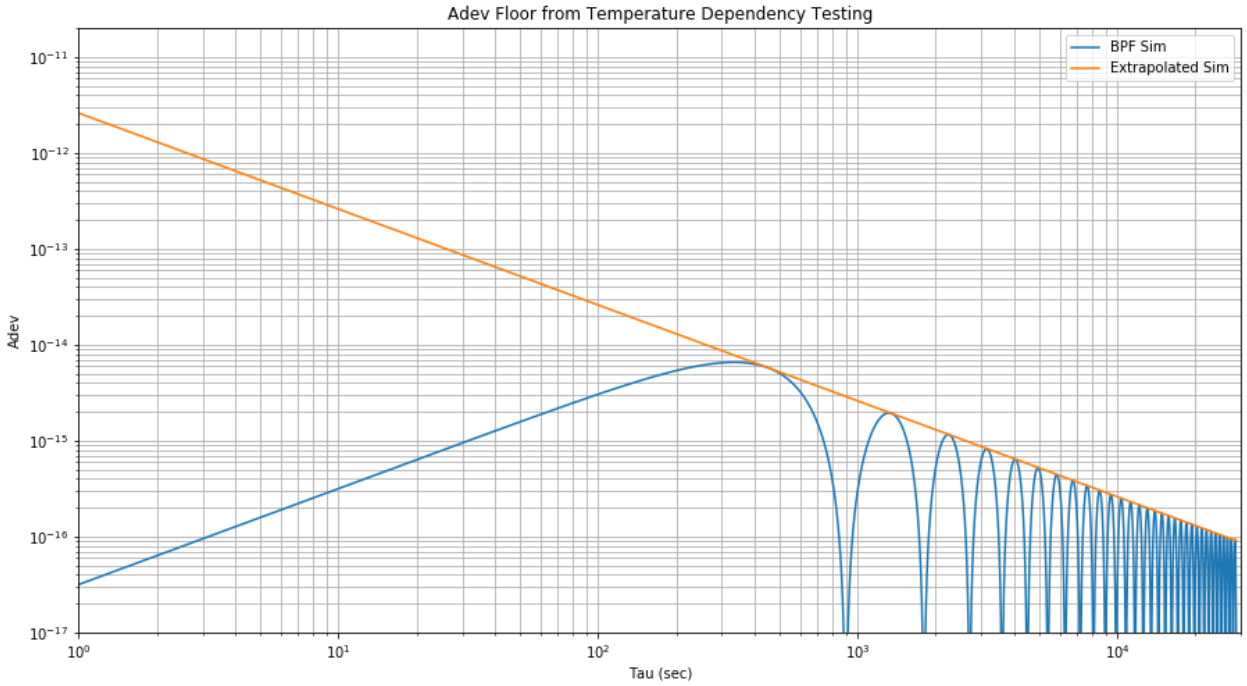
Approximate Adev Floor of Synth

A simulated allan deviation plot was created using a sin wave with an equivalent amplitude of 4°C peak to peak and a 15 minute cycle time. This is a rough approximation of the effect the A/C might have on the synthesizer box. It should be noted that the synthesizer itself has a thermal resistance due to its mass that will slow the effect of ambient temperature changes.

In [4]:
```
# Simulating worst case temperature deviation in a room due to A/C↔
```

In [5]:
```
# Adev Floor↔
```

Out[5]: <matplotlib.legend.Legend at 0xe5eac18>



The green line bounds the Adev floor; It represents a 4°C peak to peak change in temperature at a tau interval. Note that the slope of the green line corresponds to the peak to peak variation in temperature and the Tau would represent the interval at which this variation would occur.

Fractional Freq Error vs Temperature Rate

Given a temperature rate in °C/min, the corresponding fractional freq error is found. This would be used to validate temperature dependency specs of the synthesizer.

In [6]:
```
# Fractional Freq Error vs Temperature Rate↔
```