A Quick Demonstration of MCSA Using Raw Data in Python

Done in a Jupyter Notebook

Howard W Penrose, Ph.D., CMRP - President MotorDoc LLC, info@motordoc.com


```
### (c)2022, MotorDoc LLC   www.motordocai.io
```


In [2]:

```
#Here is where we import the tools that will be used for our rough pass.
import numpy as np
import csv
import matplotlib.pyplot as plt
import scipy as sp
from scipy.fft import fft, fftfreq
from scipy.fft import rfft, rfftfreq
import pandas as pd
```

In [23]:

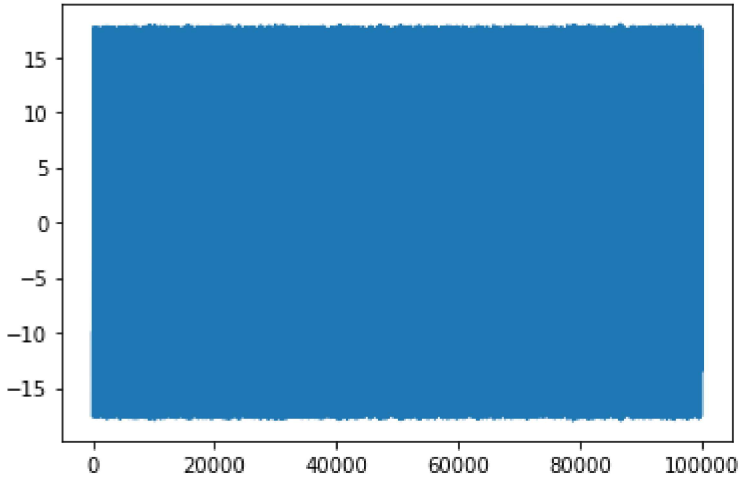
```
#This is the data that was used from a csv file.  There are 100,000 rows of data taken at 10,000 samples per second
#for 10 seconds.
df = pd.read_csv(r"C:ol.....[this part deleted]
df.head()
```

Out[23]:

	Va	Vb	Vc	Aa	Ab	Ac
0	-394.499725	254.709045	195.858688	-10.017016	17.856047	-8.399207
1	-402.846680	235.570038	217.189835	-10.709504	17.899864	-7.772016
2	-403.015320	222.332916	228.319138	-11.359893	17.884399	-7.156853
3	-403.268250	211.540878	238.942551	-12.010282	17.917047	-6.539113
4	-410.603455	193.666565	245.856201	-12.616853	17.923922	-5.917936

In [21]:

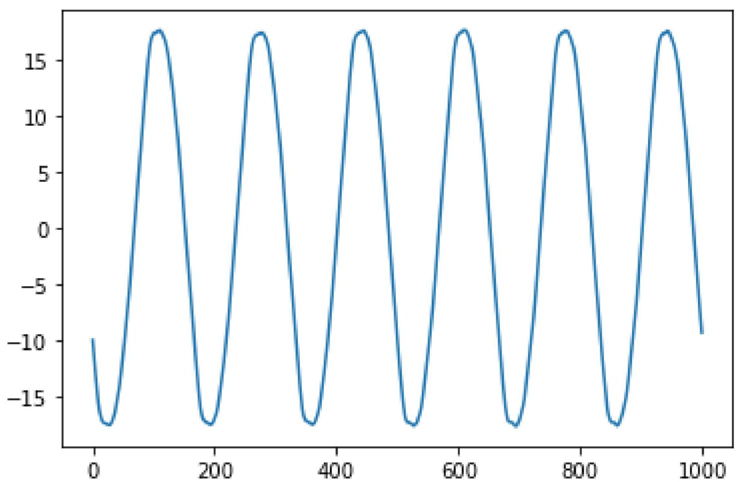
```
#We are going to look at the raw data from just one phase of current - not much to see here.
plt.plot(df['Aa'])
plt.show()
```



In [55]:

```
#So, while we prep for performing an FFT analysis, let's see what it looks like.  Wow - a perfect sine wave.
#There cannot be information there, can there?
Sec = 10 #seconds of data collected
normalized_data = np.double(df['Aa'])
plt.plot(normalized_data[:1000])
plt.show
```

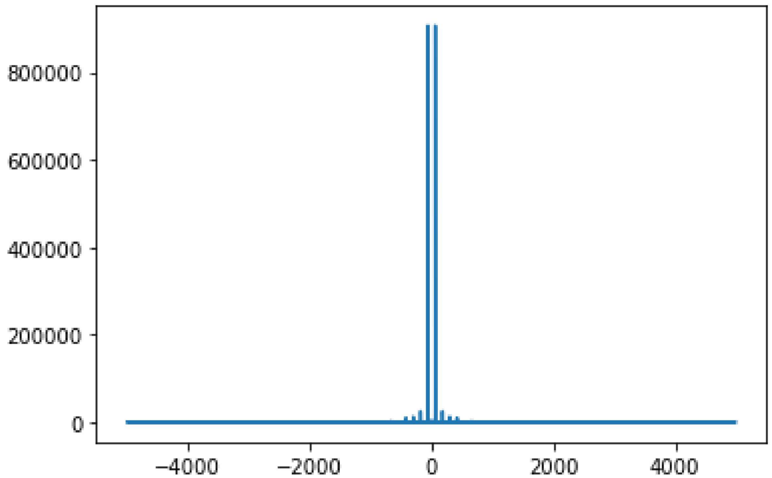
Out[55]: <function matplotlib.pyplot.show(close=None, block=None)>



In [56]:

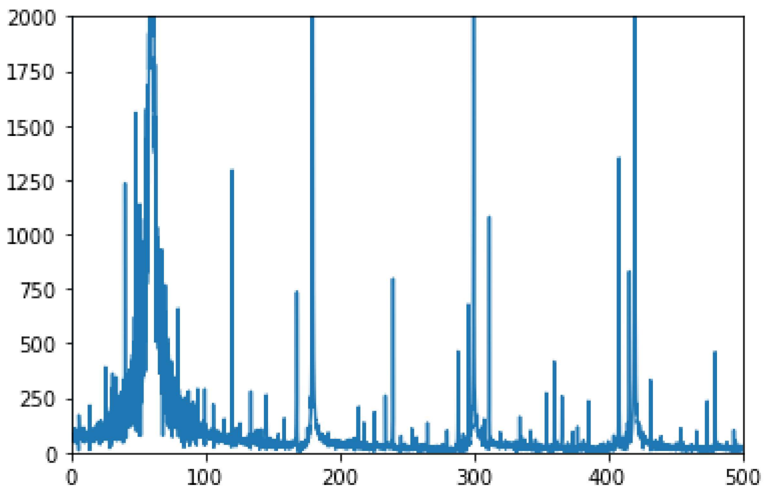
```
#Well, lets do a little data science and run a rough look.  Now, don't worry about amplitudes.  This is not
#what we will be analyzing.  However, note that there is a mirror on either side and the X axis is rediculous.
```

```
#As it stands, there's not much here to work with, is there?  I mean, the peaks are just horribly off, aren't they?
Sample_Rate = 10000 #samples per second
N = Sample_Rate*Sec #number of samples

yf = fft(normalized_data)
xf = fftfreq(N,1/Sample_Rate)

plt.plot(xf,np.abs(yf))
plt.show()
```



In [90]:

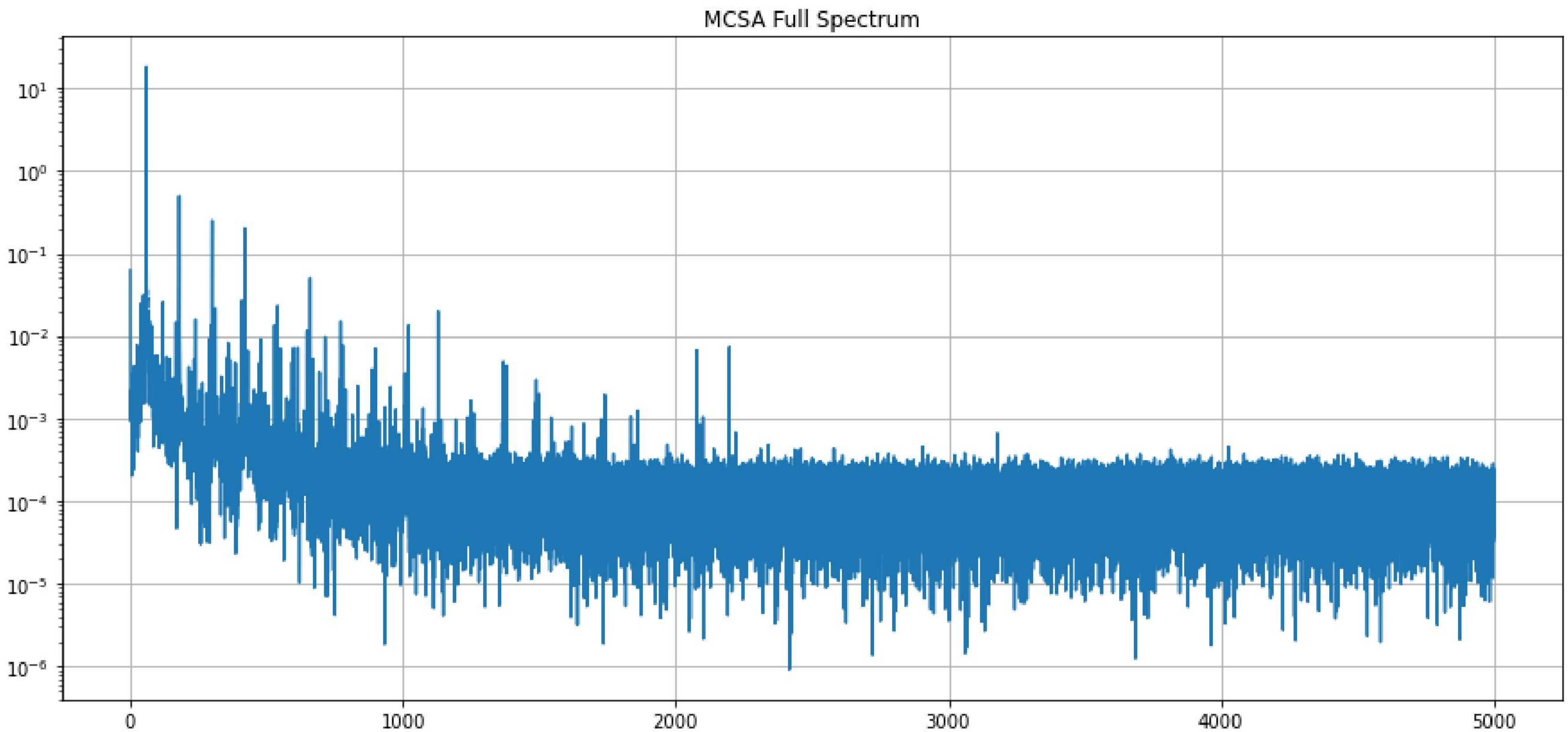
```
#What happens if we zoom in and ignore the rediculously high line frequency peak - or peaks.
Sample_Rate = 10000
N = Sample_Rate*10
yf = rfft(normalized_data)
xf = rfftfreq(N,1/Sample_Rate)

plt.plot(xf,np.abs(yf))
plt.xlim([0,500])
plt.ylim([0,2000])
plt.show()
```

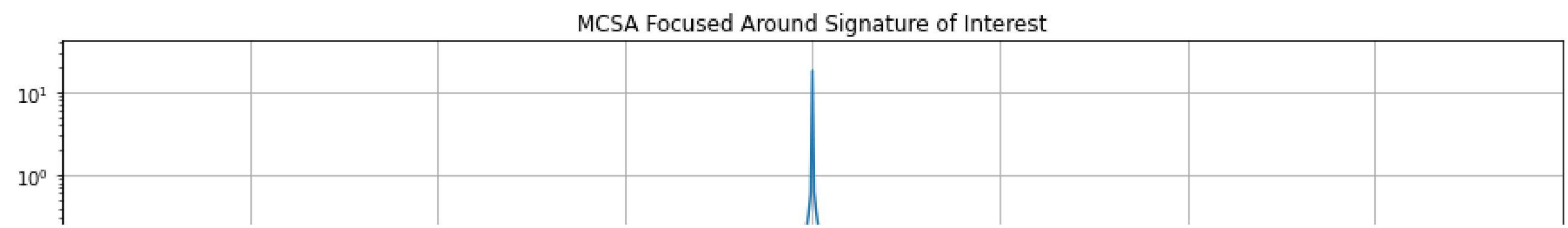


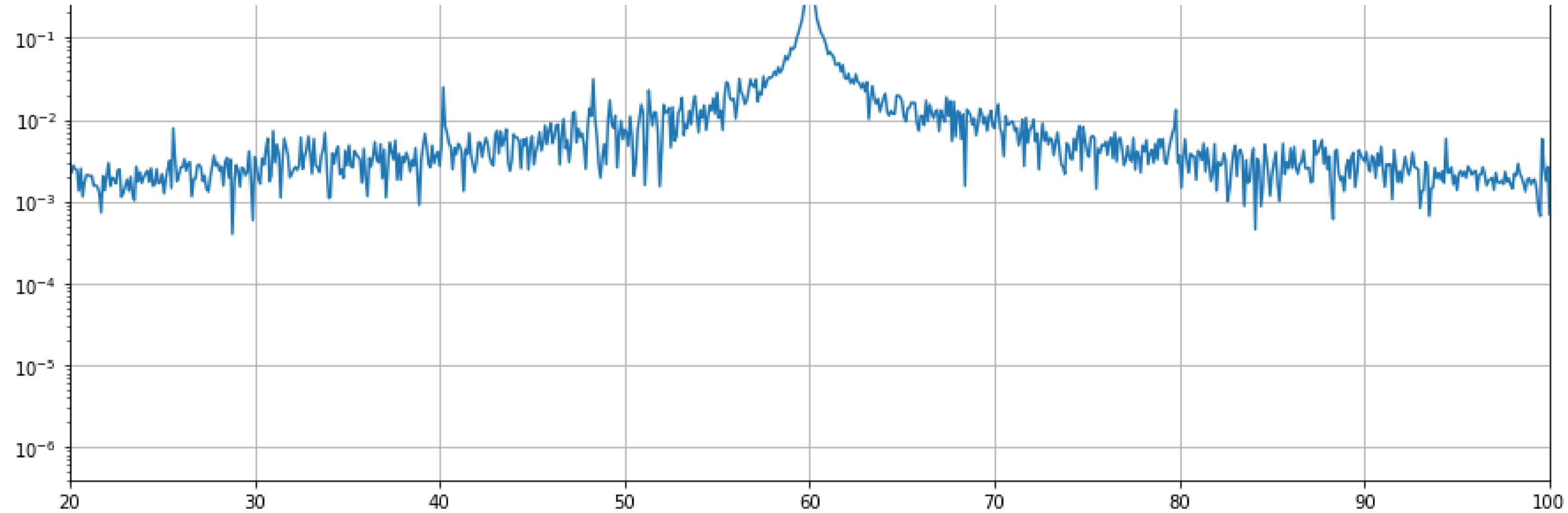
In [94]:

```
#OK, now lets look at the real logarithmic graph in dB.  These are not the values we expect, but it gets us in the right
#direction.  What's wrong with this graph?  The dB values are a factor of 10 instead of 20, which is used for current,
#leaving us well under what we should see.

#The motor is real.  It is a 6-pole, 1200 RPM (20Hz) motor with a bit of an issue.  It is an 1175 RPM, 10HP, 460Vac, 14.3
#machine that is misaligned.  As you can see below, there is a peak just above 40Hz and below 80 Hz, which are +/- runnin
#speed around the line frequency of 60Hz.

#Now we are off to a good start.  What else can we do with this data?  Visit next week as we start working on resolution,
#windows, and a bunch of other things to get this signature to look like what we see on an analyzer.
T = 1/10000 #sampling frequency (sec)
x = np.linspace(0.0,N*T,N)
y = normalized_data

yf = abs(fft(y)) #perform fft returning magnitude
xf = np.linspace(0.0,1.0/(2.0*T), N//2)#determine frequency bins

freqs = fftfreq(N,T)

#ax1 = plt.subplot(211)
#ax1.plot(x,y)
ax2 = plt.figure(figsize=(15,15))
#plt.grid()

ax2 = plt.subplot(212)
yf2 = 2/N*np.abs(yf[0:N//2]);
ax2.semilogy(xf,yf2)
plt.grid()
```

```
ax2.set_title("MCSA Focused Around Signature of Interest")
ax2.set_xlim([20,100])

ax3 = plt.subplot(211)
yf2 = 2/N*np.abs(yf[0:N//2]);
ax3.semilogy(xf,yf2)
plt.grid()
ax3.set_title("MCSA Full Spectrum")
#ax3.set_xlim([20,100])
plt.show()
```







For more information follow the Motor Diagnostics and Motor Health Newsletter by signing up on any of the MotorDoc LLC official websites: www.motordoc.com www.empathcms.com or www.motordocai.io

```
In [ ]:
```

Note: DO NOT USE for product development. There are significant portions of the design that are missing from what would be used in commercial products or MotorDocAI development. The purpose is for gaining an understanding of how MCSA, Vibration, ESA, and other related technologies are developed and managed.

For full development, please feel free to contact us. See TheRAMReview.com for a description/story related to this document.