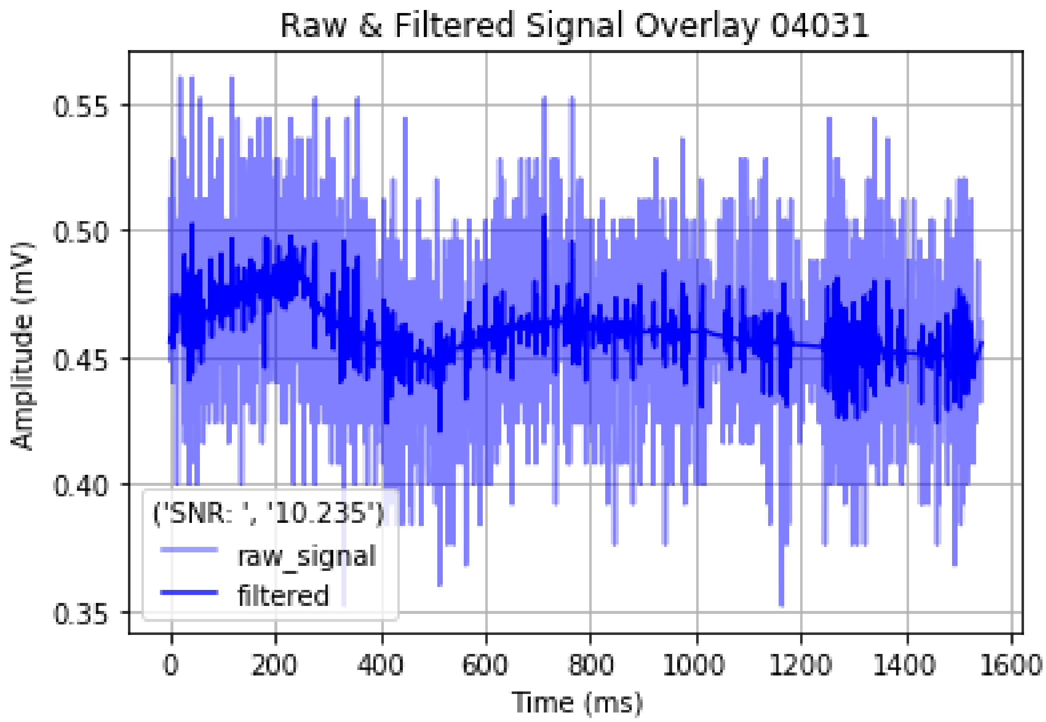
discriminate_waveletSmooth_SNR

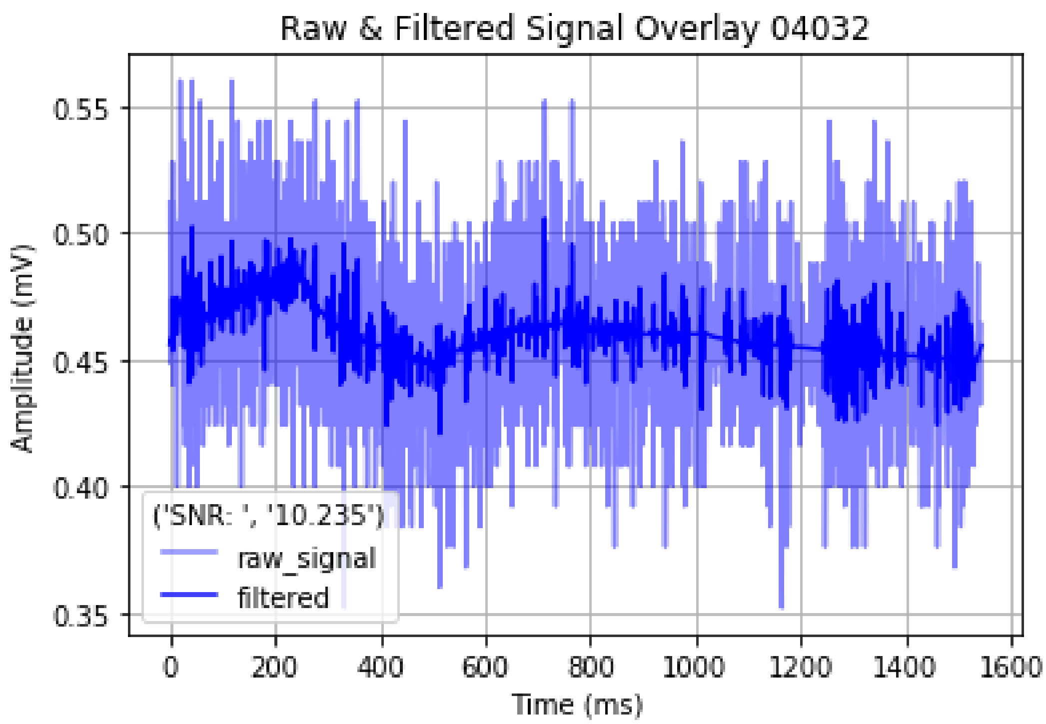
March 10, 2017

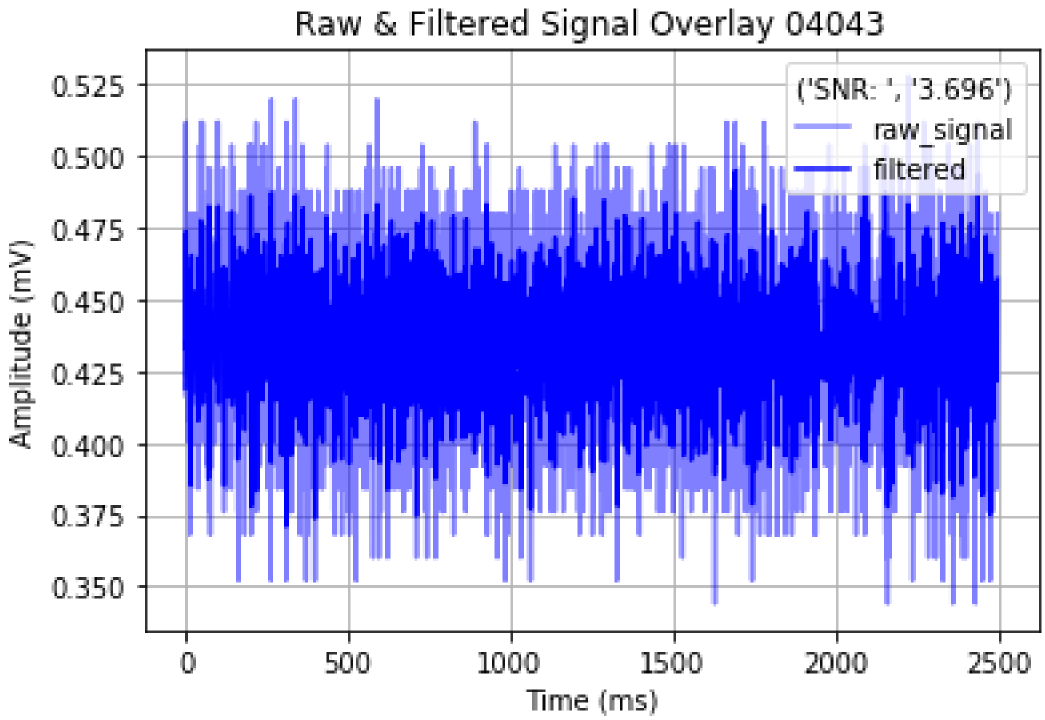
In [61]:
```
%matplotlib inline
#Discriminate wavelet reduction of motion noise
#Authored by: Connor Johnson
#Last modified: 3/10/2017 by Anna

# file processing
import os, ast
# signal processing
import pywt
import numpy as np
# visualization
import matplotlib.pyplot as plt

# Depreciated to avoid fortran dependency
#from statsmodels.robust import mad

def mad(data, axis=None):
    # median absolute deviation
    return np.median(np.absolute(data - np.median(data,axis)), axis)

def waveletSmooth(signal, wavelet, level=1):
    # returns y a rectified and smoothed signal

    # multilevel wavelet decomposition generates coefficients
    coeff = pywt.wavedec(signal, wavelet, mode="per") #by default last axis is used
    # calc a threshold to exclude outliers beyond one median absolute deviation of gaus
    sigma = mad(coeff[-level])
    signal_len = len(signal)
    threshold = sigma * np.sqrt(2*np.log(signal_len))
    # Note: alternative distance metrics can be used to vary the threshold
    coeff[1:] =(pywt.threshold(i , value=threshold, mode="soft") for i in coeff[1:])
    #reconstruct signal
    y = pywt.waverec(coeff, wavelet, mode="per")
    return y

def rms(data):
    # returns the root mean squared amplitude of the data
    baseline = np.median(data)
```

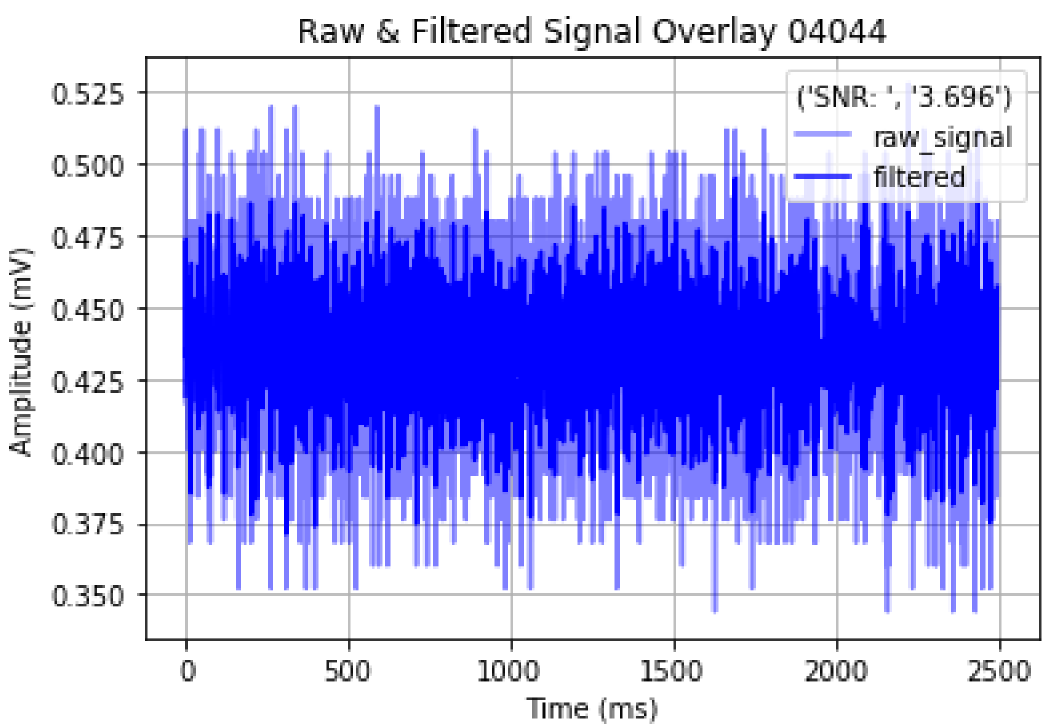
```python
    return np.sqrt(((data - baseline)**2).mean())

def signal_to_noise(signal, noise):
    # returns the signal to noise ratio
    # assumption: Equal impedance

    Asignal = rms(signal)
    Anoise = rms(noise)

    if Anoise == 0:
        SNR = float('nan')
    else:
        SNR = (Asignal/Anoise)**2
    return SNR


raw_signal_path = """raw_signal/"""

for filename in os.listdir(raw_signal_path):
        signal = []
        if (not filename.endswith('.ipynb_checkpoints')):

            # fetch signal from file
            with open(raw_signal_path + filename) as fin:
                signal = ast.literal_eval(fin.read())

                # format list of tuples (time, amplitude) as numpy array
                dt=np.dtype('float,float')
                signal = np.array(signal, dtype=dt)

                raw = [amplitude[1] for amplitude in signal]

            fin.close()

            # Eight part gaussian wavelet decomposition on amplitudes
            # f(t) = y(t) + e(t), where y(t) is the signal and e(t) is the noise
            wavelet_type = 'db2' # Daubechies wavelet mapping
            # smoothing level maxes out at 7
            filt = waveletSmooth(raw, wavelet_type, level=7)
            SNR = signal_to_noise(raw, filt)

            # visualize
            p1, = plt.plot(raw, color="b", alpha=0.5, label='raw_signal')
            p2, = plt.plot(filt, color="b", label='filtered')
            SNR_report = ('SNR: ', '%.3f'%(SNR))
            plt.legend([p1, p2], ['raw_signal', 'filtered'], loc=0, title=SNR_report)
            plt.title('Raw & Filtered Signal Overlay ' + filename )
            plt.xlabel('Time (ms)')
```

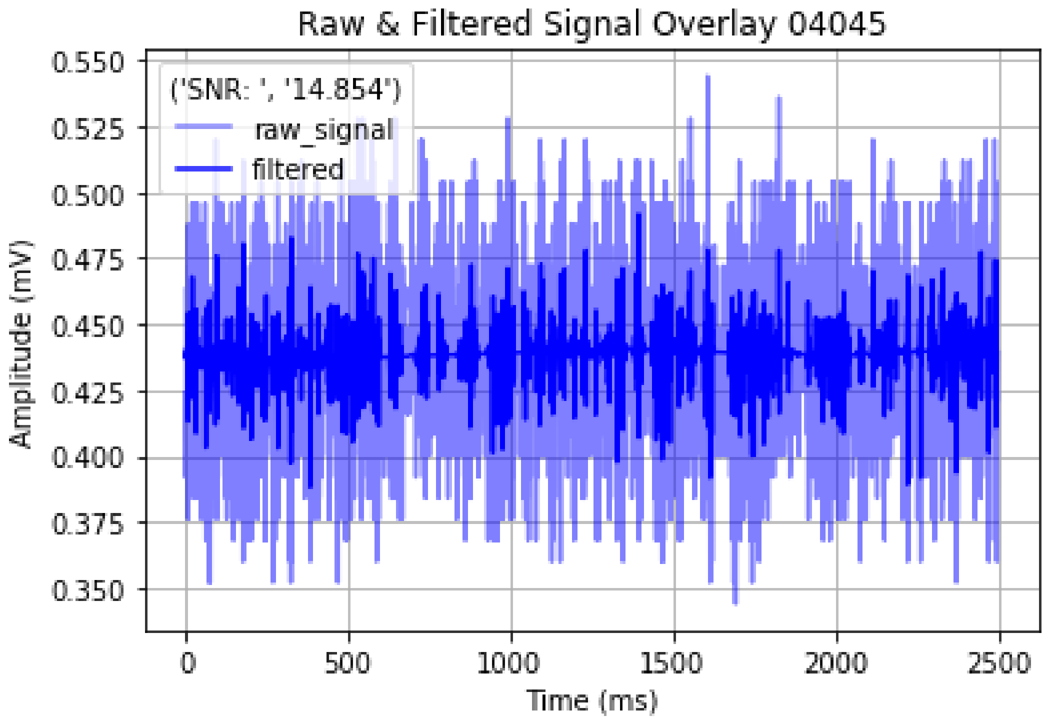
```
plt.ylabel('Amplitude (mV)')
plt.grid()
plt.show()
```

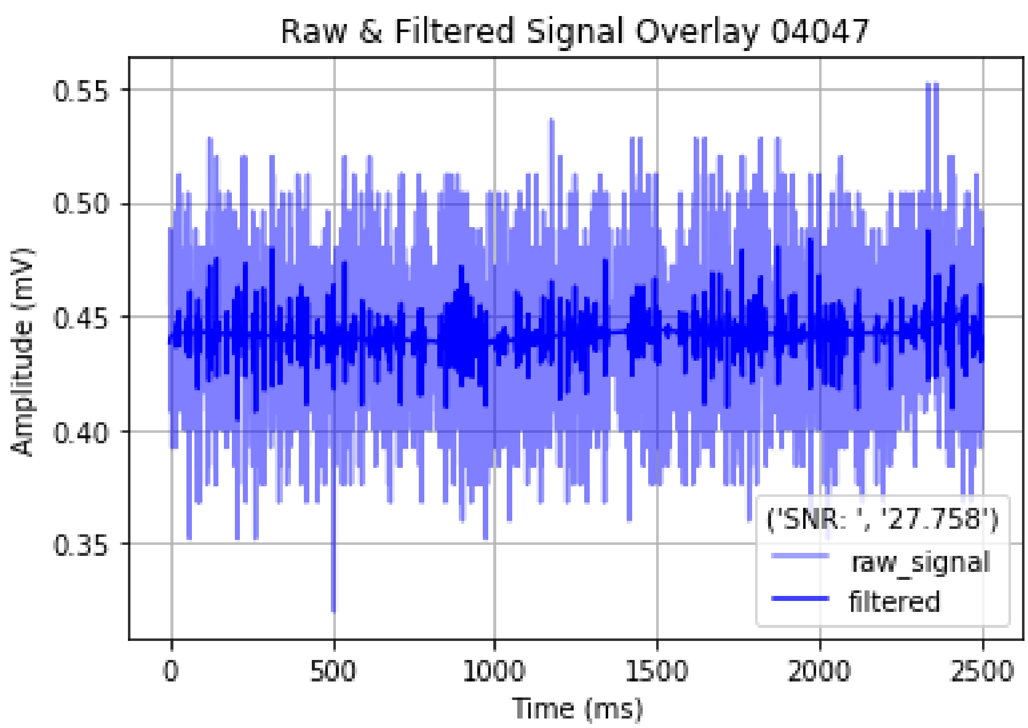


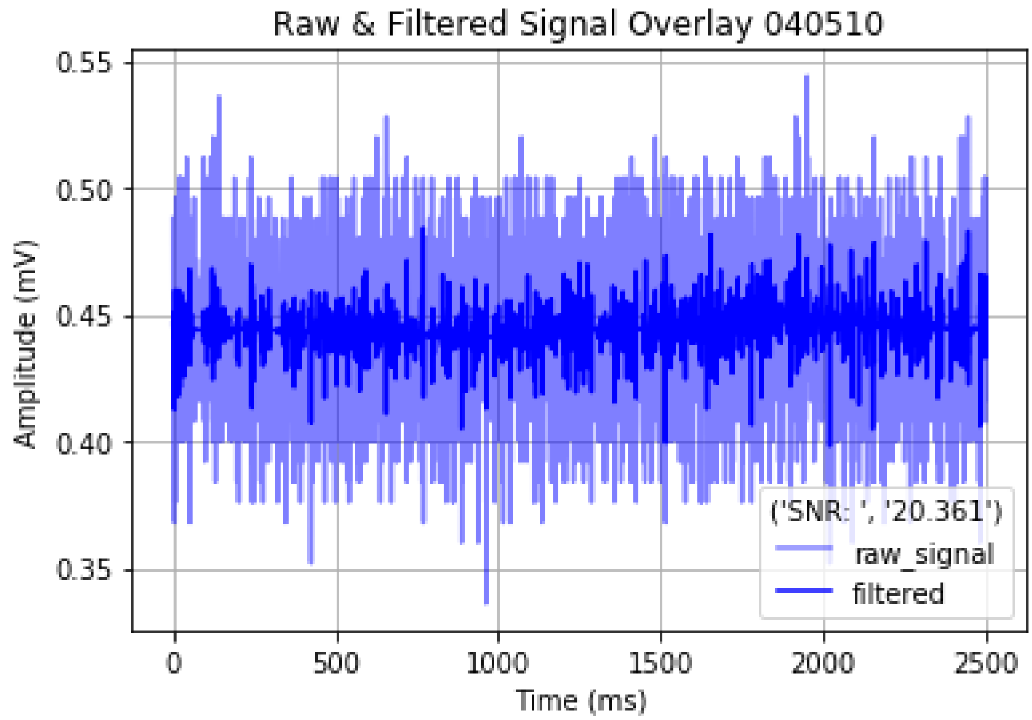


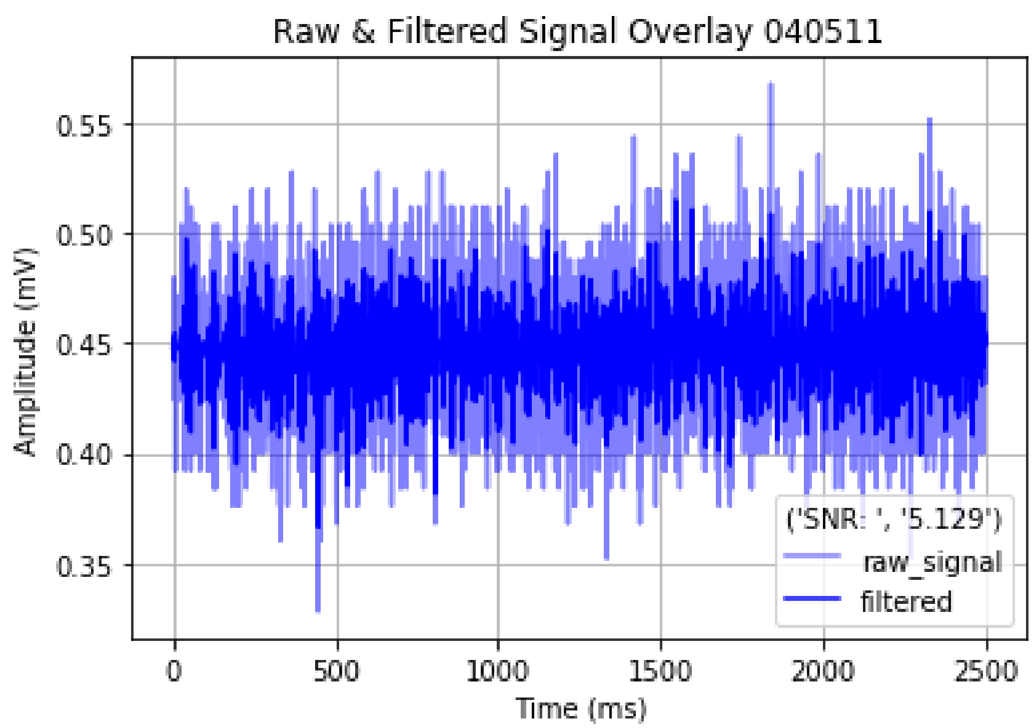

Raw & Filtered Signal Overlay 04043
('SNR: ', '3.696')
raw_signal
filtered
Amplitude (mV)
Time (ms)
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0
500
1000
1500
2000
2500

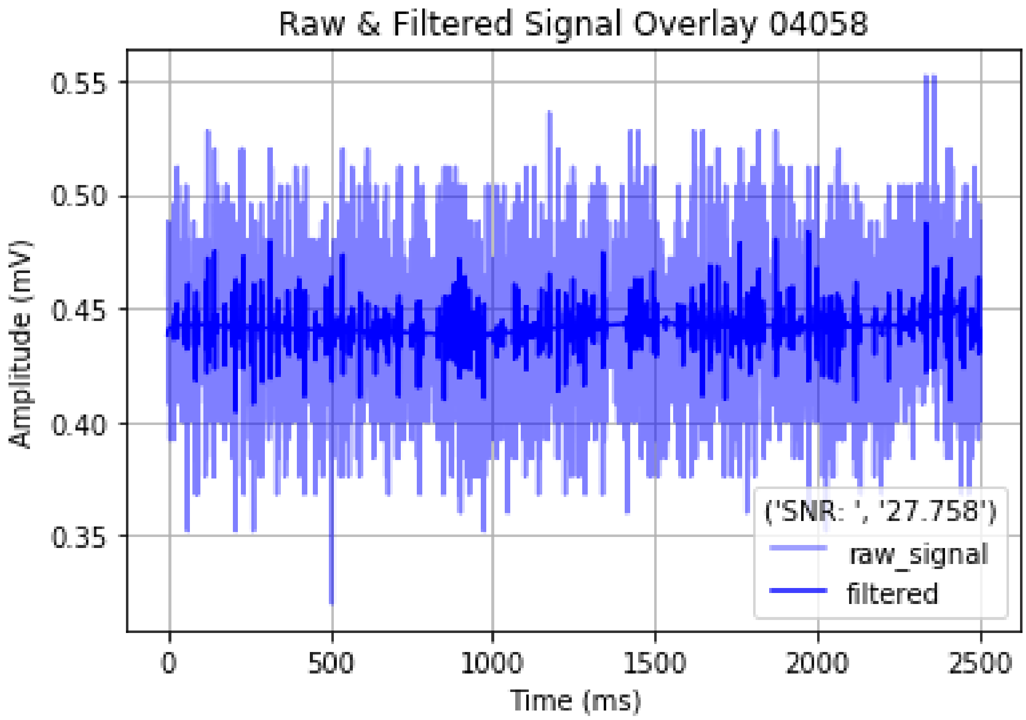

Raw & Filtered Signal Overlay 04044
('SNR: ', '3.696')
raw_signal
filtered
Amplitude (mV)
Time (ms)
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0
500
1000
1500
2000
2500

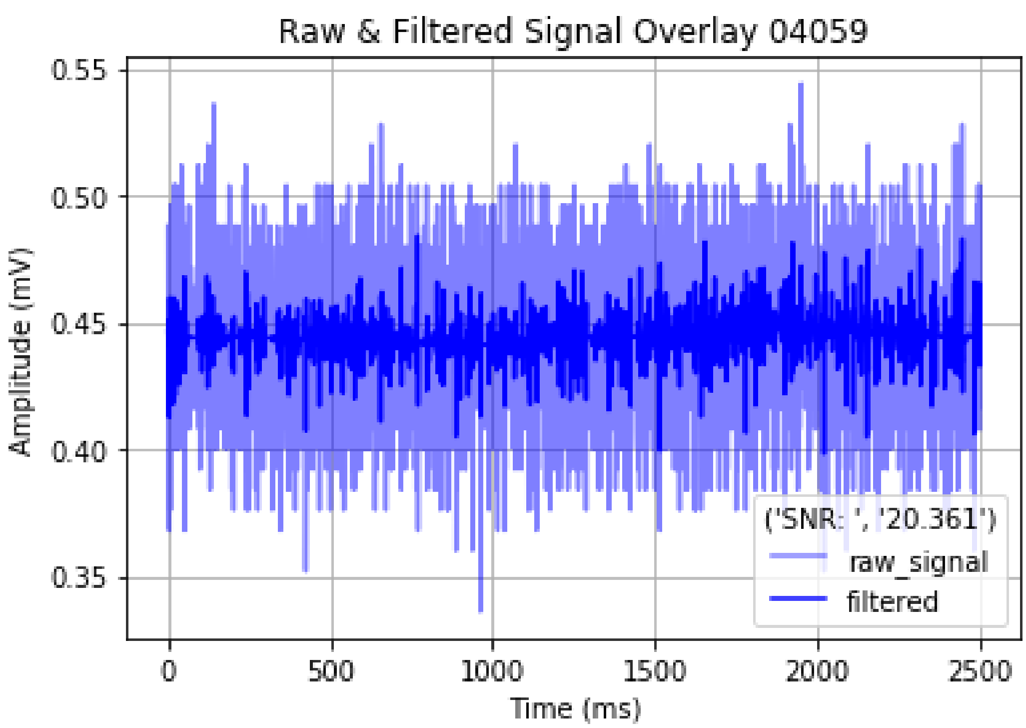

Raw & Filtered Signal Overlay 04045
('SNR: ', '14.854')
raw_signal
filtered
Amplitude (mV)
Time (ms)
0.550
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0
500
1000
1500
2000
2500

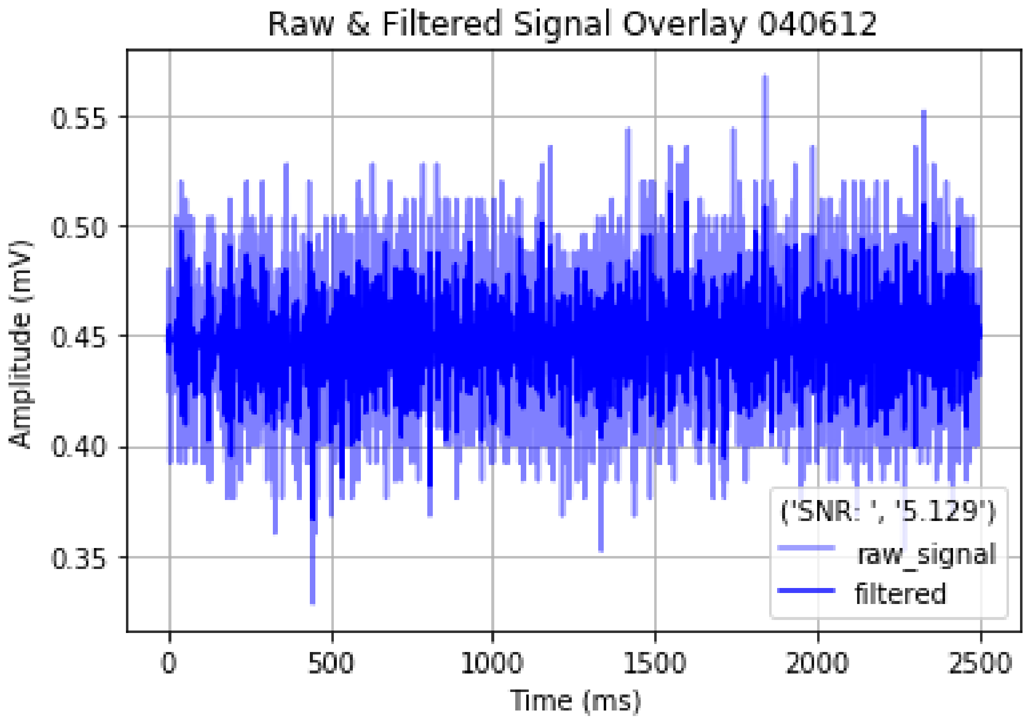

Raw & Filtered Signal Overlay 04047
('SNR: ', '27.758')
raw_signal
filtered
Amplitude (mV)
Time (ms)
0.55
0.50
0.45
0.40
0.35
0
500
1000
1500
2000
2500


Raw & Filtered Signal Overlay 040510
Amplitude (mV)
0.55
0.50
0.45
0.40
0.35
0
500
1000
1500
2000
2500
Time (ms)
('SNR: ', '20.361')
raw_signal
filtered

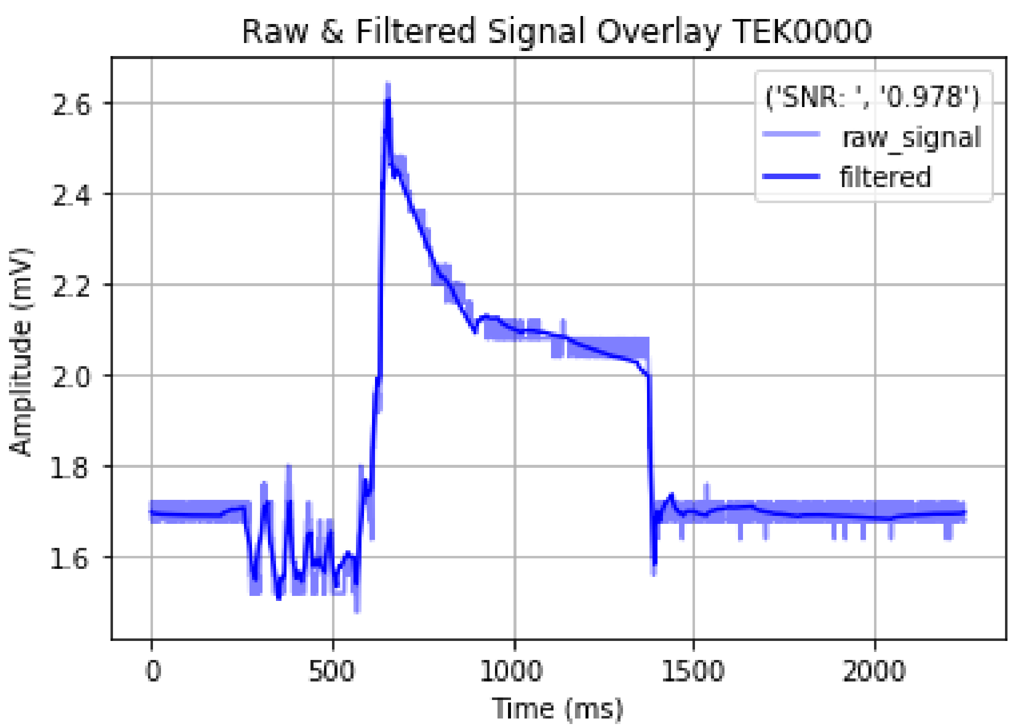

Raw & Filtered Signal Overlay 040511
Amplitude (mV)
0.55
0.50
0.45
0.40
0.35
0
500
1000
1500
2000
2500
Time (ms)
('SNR: ', '5.129')
raw_signal
filtered


Raw & Filtered Signal Overlay 04058
0.55
0.50
0.45
0.40
0.35
Amplitude (mV)
0
500
1000
1500
2000
2500
Time (ms)
('SNR: ', '27.758')
raw_signal
filtered


Raw & Filtered Signal Overlay 04059
0.55
0.50
0.45
0.40
0.35
Amplitude (mV)
0
500
1000
1500
2000
2500
Time (ms)
('SNR: ', '20.361')
raw_signal
filtered



Raw & Filtered Signal Overlay 040612
Amplitude (mV)
Time (ms)
('SNR: ', '5.129')
raw_signal
filtered



Raw & Filtered Signal Overlay TEK0000
Amplitude (mV)
Time (ms)
('SNR: ', '0.978')
raw_signal
filtered

```
In [ ]:
```