wiringoutputresist

February 24, 2020

0.1 WiringOutputResist

PHY224H1S | 2020 Winter

Jeff Shen | 1004911526

Stacy Ossipov | 1004877779

24 Feb 2020

[51]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
from scipy.optimize import curve_fit
```

[52]:
```
def model(m, b, x):
    return m * x + b

def rcs(pred, target, uncertainty, n_params):
    return np.square((pred - target) / uncertainty).sum() / (pred.size -␣
 ↪n_params)
```

[53]:
```
# read data (using dataframe because it's easier)
data = pd.read_csv('data2.csv')
data.current /= 1000 #converting to amps
data.ucurrent /= 1000
```

[54]:
```
data
```

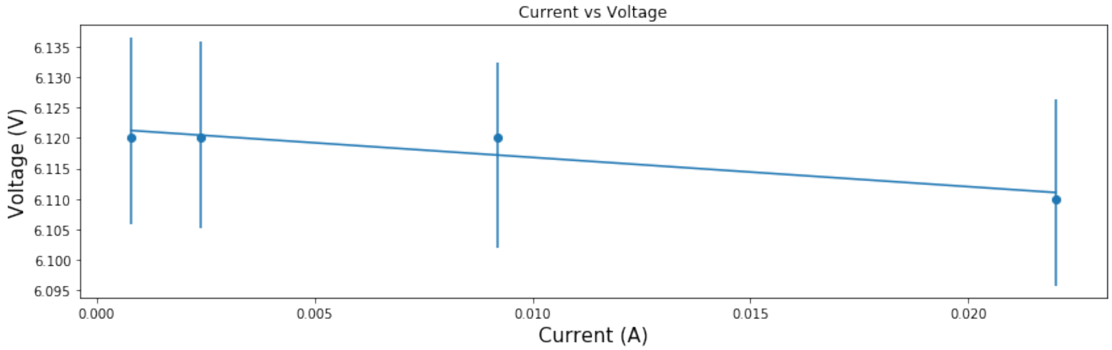
[54]:

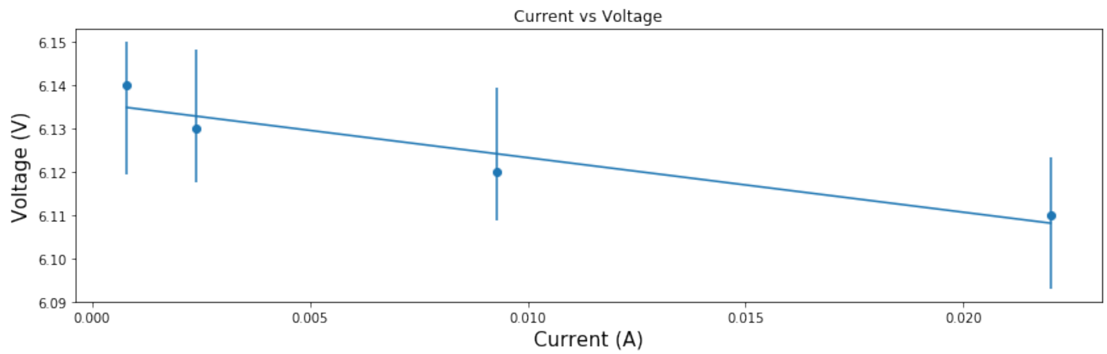
	voltage	current	uvoltage	ucurrent
0	6.11	0.0220	0.015275	0.000165
1	6.12	0.0092	0.015300	0.000100
2	6.12	0.0024	0.015300	0.000100
3	6.12	0.0008	0.015300	0.000100
4	6.11	0.0220	0.015275	0.000165
5	6.12	0.0093	0.015300	0.000100
6	6.13	0.0024	0.015325	0.000100
7	6.14	0.0008	0.015350	0.000100
8	9.97	0.0359	0.024925	0.000269

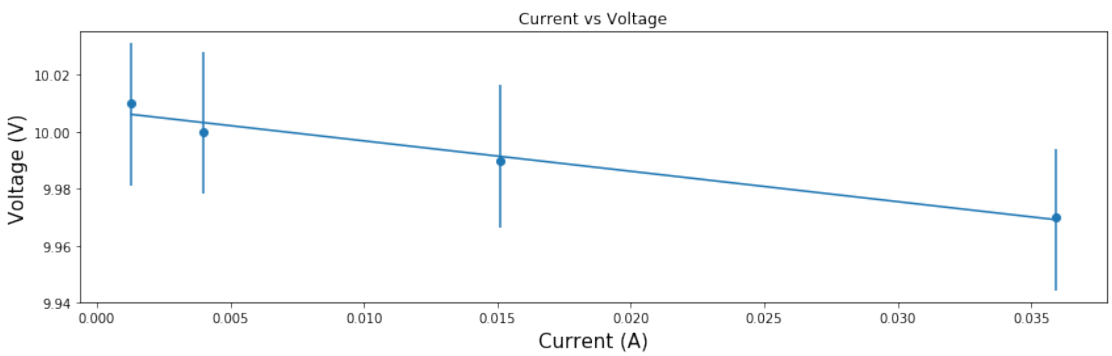
```
9      9.99   0.0151  0.024975  0.000113
10    10.00   0.0040  0.025000  0.000100
11    10.01   0.0013  0.025025  0.000100
12    15.03   0.0541  0.037575  0.000406
13    15.07   0.0228  0.037675  0.000171
14    15.11   0.0061  0.037775  0.000100
15    15.12   0.0020  0.037800  0.000100
16    19.26   0.0693  0.048150  0.000520
17    19.29   0.0292  0.048225  0.000219
18    19.31   0.0078  0.048275  0.000100
19    19.32   0.0026  0.048300  0.000100
```

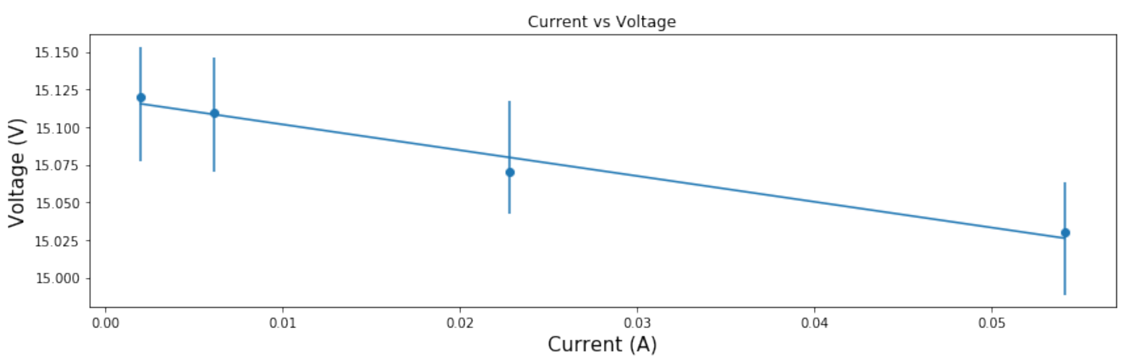
[55]:
```python
# making multiple plots

fig, ax = plt.subplots(5, figsize=(12, 20))
x = 4

resistances = np.zeros(int(data.shape[0] / x))
errors = np.zeros(int(data.shape[0] / x))
fits = np.zeros(int(data.shape[0] / x))

# loop over all resistors
for i in np.arange(0, data.shape[0], x):

    current = data.current[i:i+x]
    voltage = data.voltage[i:i+x]
    ucurrent = data.ucurrent[i:i+x]
    uvoltage = data.uvoltage[i:i+x]

    # fit curve
    popt, pcov = curve_fit(f=model, xdata=current, ydata=voltage,␣
↪sigma=uvoltage, absolute_sigma=True)
    # reduced chi squared values
    fit_score = rcs(model(current, *popt), voltage, uvoltage, 2)

    # save slope (resistance) and corresponding errors

    resistances[int(i/x)] = popt[1]
    errors[int(i/x)] = np.sqrt(np.diag(pcov))[1]
    fits[int(i/x)] = fit_score

    print('\n')

    # plot results of fit with error bars
    ax[int(i/x)].errorbar(current, model(current, *popt), yerr=uvoltage,␣
↪marker='', ls='-')
    ax[int(i/x)].scatter(current, voltage)
```

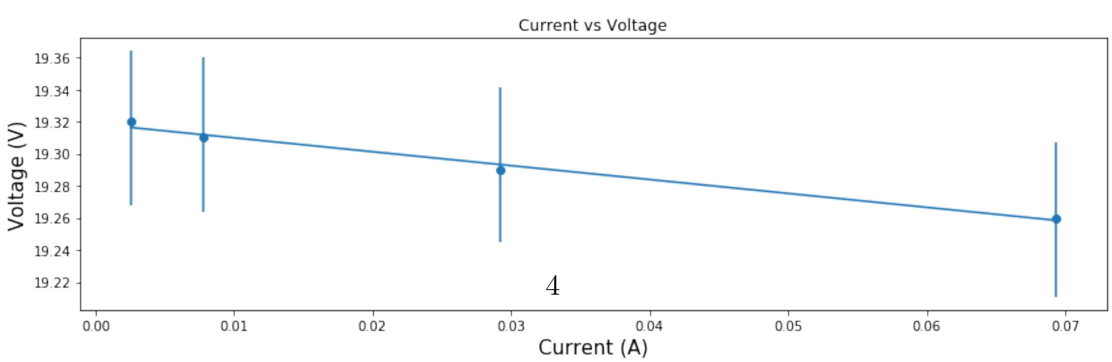
```
    ax[int(i/x)].set_xlabel('Current (A)', fontsize=15)
    ax[int(i/x)].set_ylabel('Voltage (V)', fontsize=15)
    ax[int(i/x)].set_title('Current vs Voltage')

fig.tight_layout(h_pad=3)
```


Current vs Voltage
Voltage (V)
6.135
6.130
6.125
6.120
6.115
6.110
6.105
6.100
6.095
0.000
0.005
0.010
0.015
0.020
Current (A)


Current vs Voltage
Voltage (V)
6.15
6.14
6.13
6.12
6.11
6.10
6.09
0.000
0.005
0.010
0.015
0.020
Current (A)


Current vs Voltage
Voltage (V)
10.02
10.00
9.98
9.96
9.94
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
Current (A)


Current vs Voltage
Voltage (V)
15.150
15.125
15.100
15.075
15.050
15.025
15.000
0.00
0.01
0.02
0.03
0.04
0.05
Current (A)


Current vs Voltage
Voltage (V)
19.36
19.34
19.32
19.30
19.28
19.26
19.24
19.22
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Current (A)

[56]:

```
print('Resistances')
for i in resistances:
    print(f'{-i:.3f}') # take the negative slope to be the resistance

print('\nCorresponding Errors')
for j in errors:
    print(f'{j:.3f}')

print('\nReduced Chi Squared of Fits')
for k in fits:
    print(f'{k:.3f}')
```

```
Resistances
0.480
1.261
1.068
1.713
0.866

Corresponding Errors
0.915
0.915
0.916
0.918
0.918

Reduced Chi Squared of Fits
0.023
0.117
0.022
0.047
0.006
```

[57]:

```
# calculating the mean resistance for the power supply (over 4 different␣
↪voltage values)
print(f'{-resistances[1:].mean():.3f}')
print(f'{errors[1:].mean():.3f}')
```

```
1.227
0.917
```

[ ]:

```
```