Isotope Shift: Hydrogen and Deuterium

Brandon Meggerson
Physics Department, University of California,
Santa Barbara, CA 93106-9530

Dated: October 27, 2024

1 Abstract

The isotope shift between Hydrogen and Deuterium for the first six Balmer lines is investigated. The average distance between peaks of the Hydrogen and Deuterium spectra is found by first using a known doublet to find a correlation between separation in time and separation in distance, and then sweeping over Hydrogen-Deuterium spectrum with a spectrometer. I find for n = 6 1.92+-0.06 angstroms, n = 5 1.33+-0.04A, n= 4 1.33+-0.04 , n = 3 1.20 +-0.04, n =2 1.20 +- 0.04, n=1 1.29 +- 0.04

2 Introduction

The Rydberg equation provides a relation between the energy level n, the reduced mass of an atom, and the wavelength of light given off from a transition from a higher energy level to a lower one.

$$\frac{1}{\lambda} = \frac{R_\infty}{1 + \frac{m_e}{M}} \left(\frac{1}{n_1^2} - \frac{1}{n_2^2} \right) \tag{1}$$

This means that a change in the mass of an atom would cause a change in wavelength, as such isotopes such as Hydrogen and Deuterium should display a shift in wavelength. Using a spectrometer such differences in wavelength can be measured.The spectrometer slowly turns a diffraction grating which separates the light by constructive interference. The spectrometer takes in light through a slit and can sweep through a spectrum to get to specific wavelengths. A photomultiplier tube takes faint light sources such as that coming through the slits of the spectrometer and emits photoelectrons when the photons hit a semiconducting material which causes a current to be produced. Said current can be recorded and analysis of the data can be used to find the isotope shifts.

3 Experimental Methods

This experiment uses a spectrometer, a photomultiplier tube (PMT), a picoammeter, and computer with the application Datalogger, a Hg lamp, and a Hydrogen-Deuterium lamp.The PMT is installed in the spectrometer and connected to the picoammeter in order to detect the current going through the PMT slit. The picoammeter is also wired to a computer that uses Datalogger to plot the current vs time in order to produce a spectra. The first measurement taken was the 6263 A doublet on the Hg spectrum. The Hg lamp is placed about 2 inches away from the spectrometer and behind the UV screen. The PMT slit width is set to 20 microns and the Spectrometer slit is set to

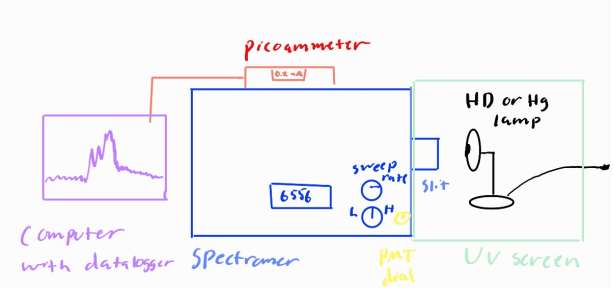


Figure 1: Diagram of Setup Blue:Spectrometer, Purple: Computer, Red: Picoammeter, Yellow: PMT slit knob, Black: Lamp, Teal: UV screen

30 microns. The doublet is recorded at a 5 A/min sweep going up in wavelength. This is repeated five times and each plot of the doublet is saved as a csv file. After plotting the files onto python the distance between the peaks is recorded (time difference) and an average is calculated. The table of emission lines from a mercury arc lamp provided by the National Institute of Standards and Technology is used to find the known distance between the peaks in angstroms so that a correlation between distance and time can be found. Next the Balmer lines of the UV Hydrogen-Deuterium lamp are recorded. The Hg lamp is replaced with the HD lamp and is placed approximately 3 inches away from the spectrometer slit. Starting from the higher wavelengths in the series. A table of the emission line spectrum of Hydrogen from the NIST is used to know the approximate wavelengths that should be swept through. The first six Balmer lines for Hydrogen are 6562A, 4861A, 4340A, 4101A, 3970A, and 3889A. For all of the measurements the spectrometer is swept up at 5A/min and is repeated 5 times. The width of the slits are increased at lower wavelengths in order to resolve the peaks. The range on the picoammeter may also need to be adjusted as the current may overflow which would cause an incorrect graph to be recorded or the current may be too low which would cause the peaks to be unresolvable from the noise. The lower energy wavelengths (3970A, and 3889A) need a lower range of the picoammeter on the scale of 10s of nano amps and the slit widths should be 90 microns on the PMT and 30 microns on the spectrometer. After plotting the data from the Datalogger the time at which each peak occurred is recorded, the data that was recorded in this experiment have oscillations which results in a main peak that is comprised of what appears to be several peaks. The solution to this issue was to record the time of the highest peaks in each region. The time difference between the peaks are multiplied by the Angstroms/time rate that was found from the Hg lamp calibration. The result is the isotope shifts for Hydrogen and Deuterium for the first 6 energy levels in the Balmer series.

4 Raw Data

Each Trial has a graph that the data comes from. They will be included at the end of the paper in an appendix. Each trial has the same sources of error present. There are dark currents that get recorded by the PMT, dark currents are electrons that are released by thermal excitations instead of from a photon striking semiconducting material. These currents are noise that could appear like a peak. The picoammeter is zeroed to suppress some of the noise but it can still be recorded. Lower energy levels require higher slit widths which could increase the uncertainty due to photons that are not from the lamps entering the slits. This uncertainty is difficult to quantify but it is present. The uncertainty listed is based on the measurement being reproduced and is under the assumption that

the distribution is Gaussian. The formula

$$\sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(\delta t - \delta t_a vg)^2}$$

gives the uncertainty for each average change in time.

Table 1: Hg Lamp Calibration $\delta t = 6.4 \pm 0.540$

Trial	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	41	47	6
2	34	40	6
3	25	31	6
4	47	54	7
5	29	36	7

Table 2: n=6 $\delta t = 20.2 \pm 0.410$

Trial	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	21	41	20
2	44	64	20
3	31	51	20
4	31	52	20
5	50	70	20
6	50	70	20

Table 3: n=5 $\delta t = 14.0 \pm 2.34$

Trial	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	52	62	10
2	33	47	14
3	30	45	15
4	30	45	15
5	25	41	16

Table 4: n=4 $\delta t = 14 \pm 0.816$

Trial	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	42	55	13
2	26	41	15
3	30	44	14
4	32	46	14
Data Lost /	/	/	/

Table 5: n=3 $\delta t = 12.6 \pm 0.548$

Trial	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	37	50	13
2	53	65	12
3	55	67	12
4	38	51	13
5	54	67	13

Table 6: n=2 $\delta t = 12.6 \pm 0.894$

Trial Uncertainty	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	58	71	13
2	33	45	12
3	22	34	12
4	44	56	12
5	42	56	14

Table 7: n=1 $\delta t = 13.6 \pm 2.51$

Trial Uncertainty	Peak 1 Time (s)	Peak 2 time (s)	Difference (s)
1	26	39	13
2	16	34	18
3	25	37	12
4	28	41	13
5	23	35	12

5 Results

The relation between distance and time found is $0.045 \pm 0.003 \mathring{A}$. This is simply found by dividing the known distance between the doublet peaks by the time between them from the Hg graph. Multiplying this rate by the average time between the two peaks in the HD graphs will give the average isotope shift between Hydrogen and Deuterium.

Table 8: Hg Lamp Calibration

Energy Level	Isotope Shift in Angstroms
1	$1.292 \pm 0.2419 \mathring{A}$
2	$1.97 \pm 0.0930 \mathring{A}$
3	$1.97 \pm 0.06434 \mathring{A}$
4	$1.33 \pm 0.08817 \mathring{A}$
5	$1.33 \pm 0.2262 \mathring{A}$
6	$1.92 \pm 0.0720 \mathring{A}$

The uncertainty in the isotope shift was found by propagating the uncertainty from the average change in time and the rate of Angstroms per second. Angstroms per second in itself propagates error from the change in time for the HG lamp data and the Angstrom resolution and was found to be approximately 0.003. With this information a formula can be used to find the uncertainty for the isotope shift at each level.

$$\sqrt{(\delta t_a vg \cdot 0.0003)^2 + (0.045 \cdot \sigma t_a vg)^2} = \sigma_S hift$$

The accepted values for the isotope shift between Hydrogen and Deuterium is between 1 and 2 Angstroms. All of the isotope shifts found are within the accepted range but large uncertainty variations call for additional data to be taken in order to decrease the uncertainty and find the source of higher errors in specific energy levels. The higher uncertainties are likely a result of human error, I approximated the time of the peaks by eye when reading the graphs so there is a chance I could have been off particularly at energy level 5. There is also the possibility of short overflows on the picoammeter when I was recording data, I only have one set of eyes so the picoammeter could have briefly overflowed while I was watching the Datalogger which would increase the uncertainty of the location of the peaks.

6 Discussion

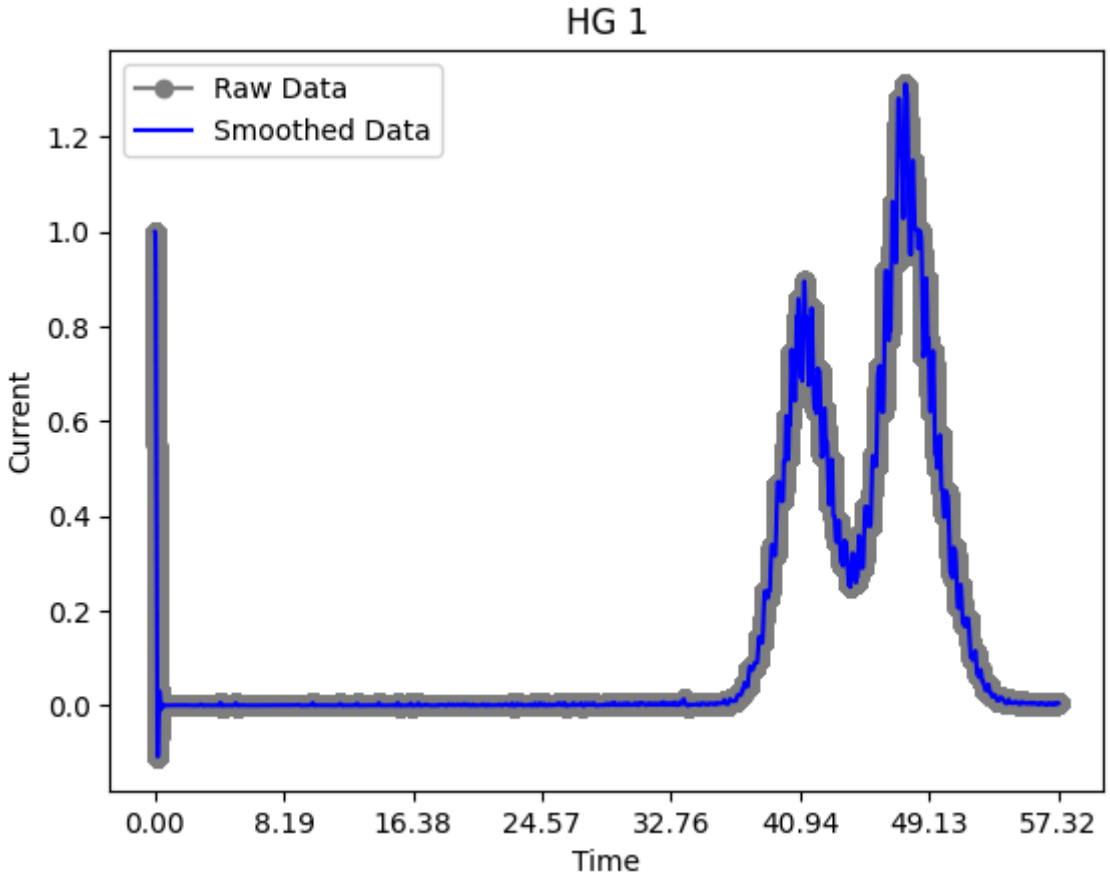
The general isotope shift between Hydrogen and Deuterium met my expectations but the uncertainty for the energy levels leads me to believe there were larger systematic errors that could be improved. The range of 1 to 2 Angstroms is quite high for this type of measurement so it is not surprising all of the values found in this experiment fell within that range but the uncertainty should decrease as the energy level increases which does not occur. As previously discussed, it could be that a small overflow occurred, this could be resolved by doing the experiment with multiple people and having one person keep an eye on the picoammeter. It could also be that the peaks that I found were slightly off, this could be resolved by writing a program that specifically gives each peak. This method was attempted but there were problems restraining the region for each peak search, a more talented programmer could find a way to solve this issue. Lower energy levels have peaks

that are smaller which means they are more difficult to resolve from the noise. The noise on the last two energy levels is approximately 3/4 the height of the peak, this is not ideal and so lower the noise, by conducting this experiment in a dark room for example, would make it easier to resolve and give more accurate results.

7 Appendix

In [1]:
```
#These are just rough plots from the data collected from datalogger

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.signal import find_peaks

# Load the CSV file
data = pd.read_csv('HGD1.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')



plt.xlabel('Time')
plt.ylabel('Current')
plt.title('HG 1')

num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

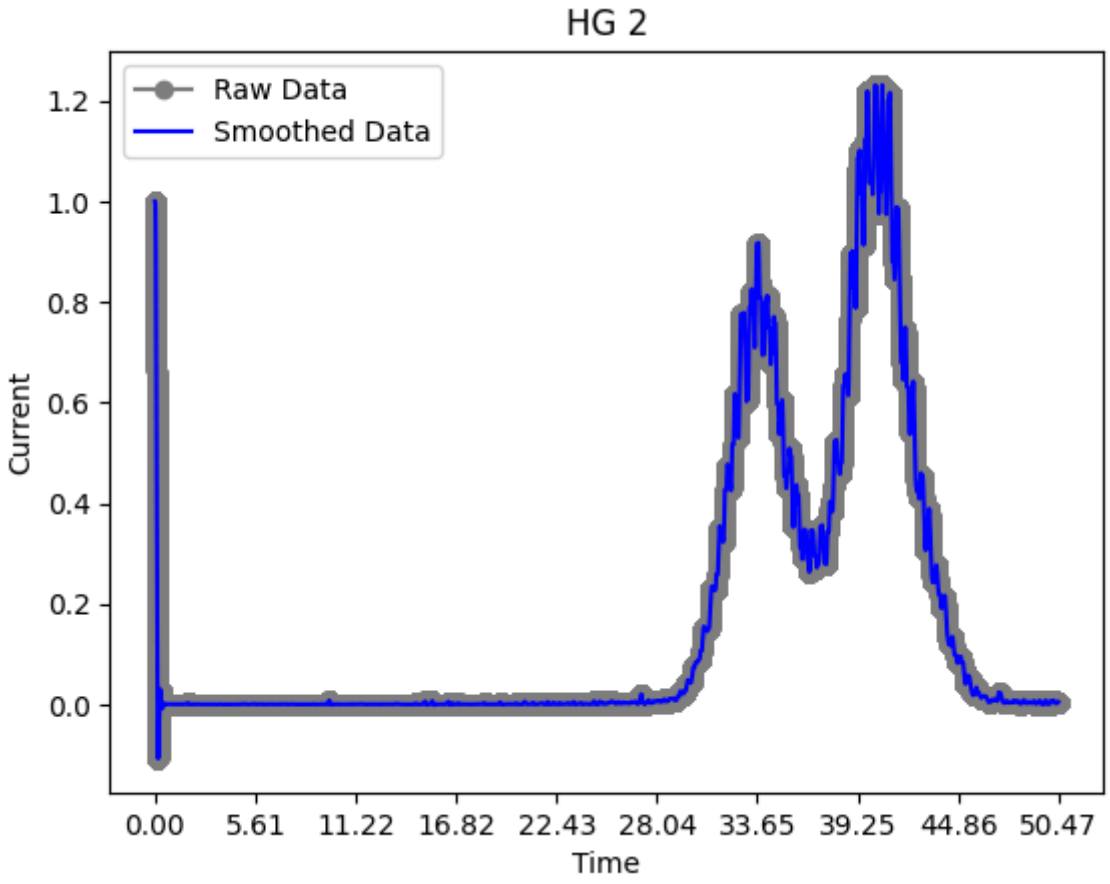
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('HGD2.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('HG 2')
```

```
num_ticks = 10  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

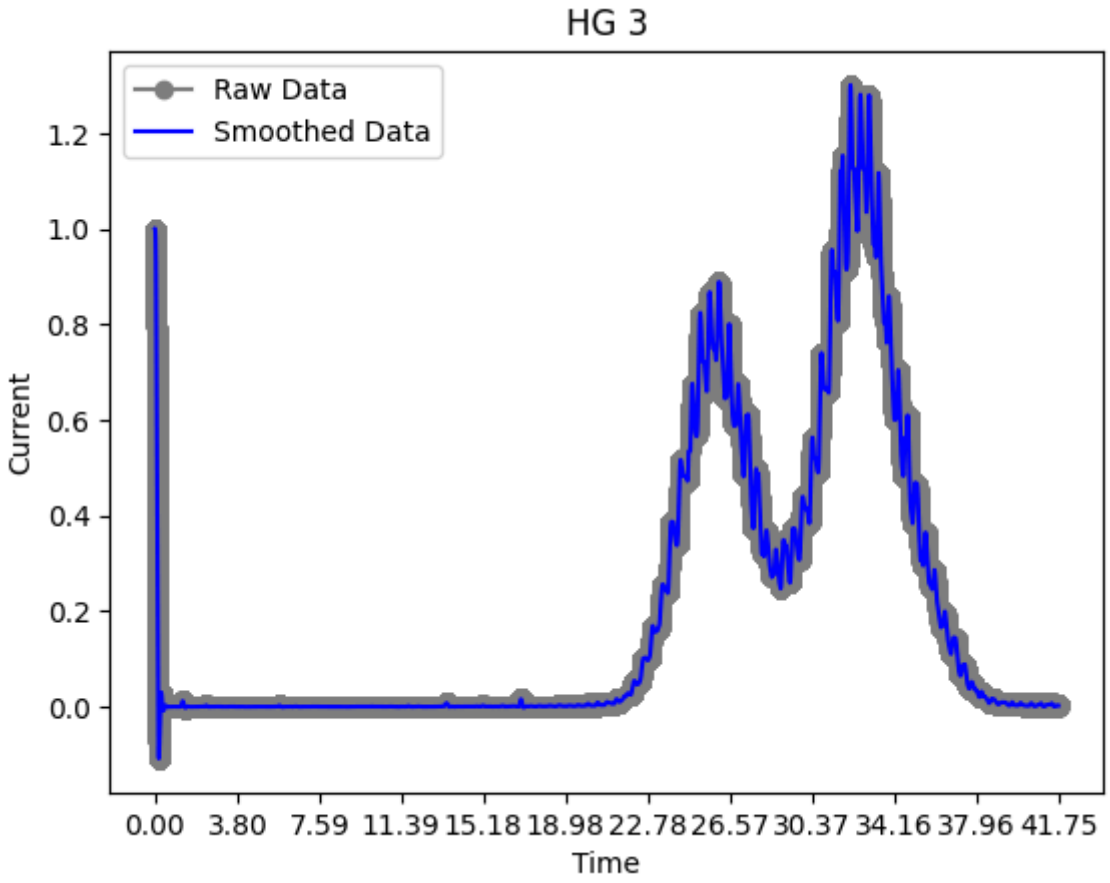
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('HGD3.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('HG 3')

num_ticks = 12  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

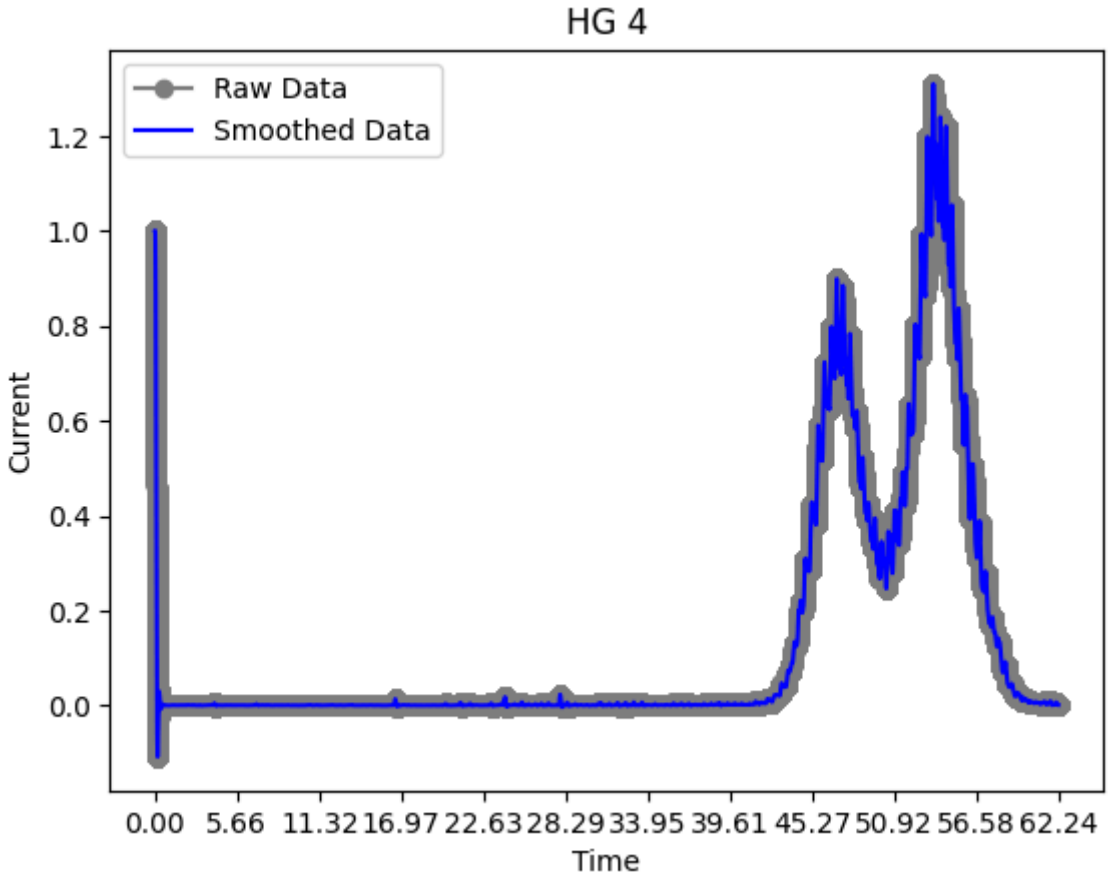
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# Load the CSV file
data = pd.read_csv('HGD4.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('HG 4')

num_ticks = 12  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

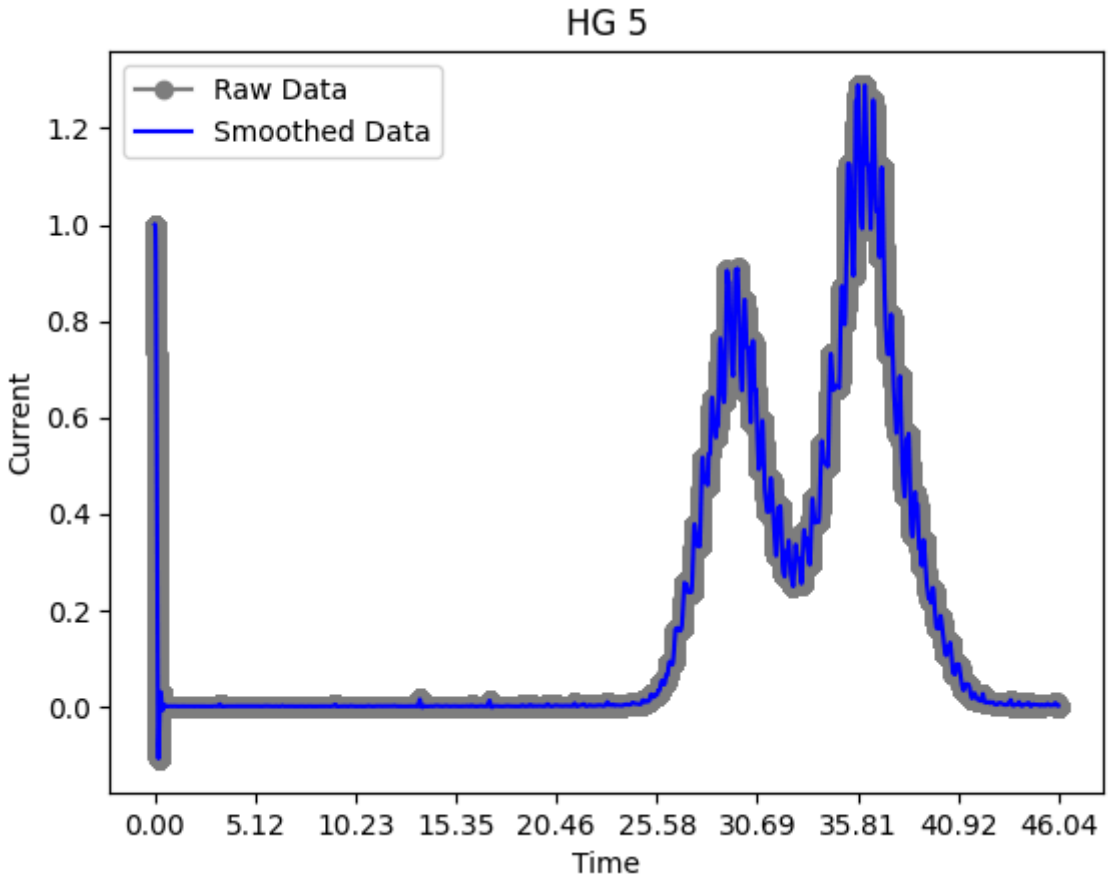
```
        0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('HGD5.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('HG 5')
```

```
num_ticks = 10  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

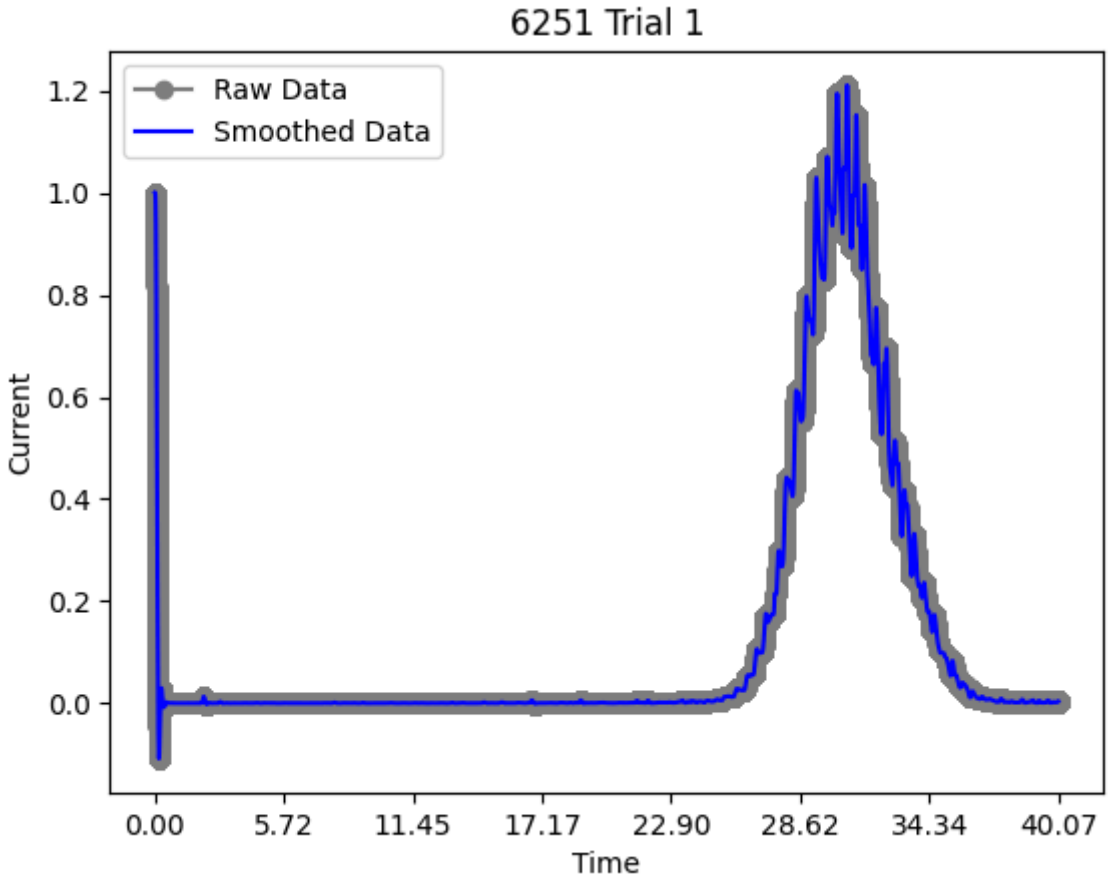
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6251T1.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('6251 Trial 1')

num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

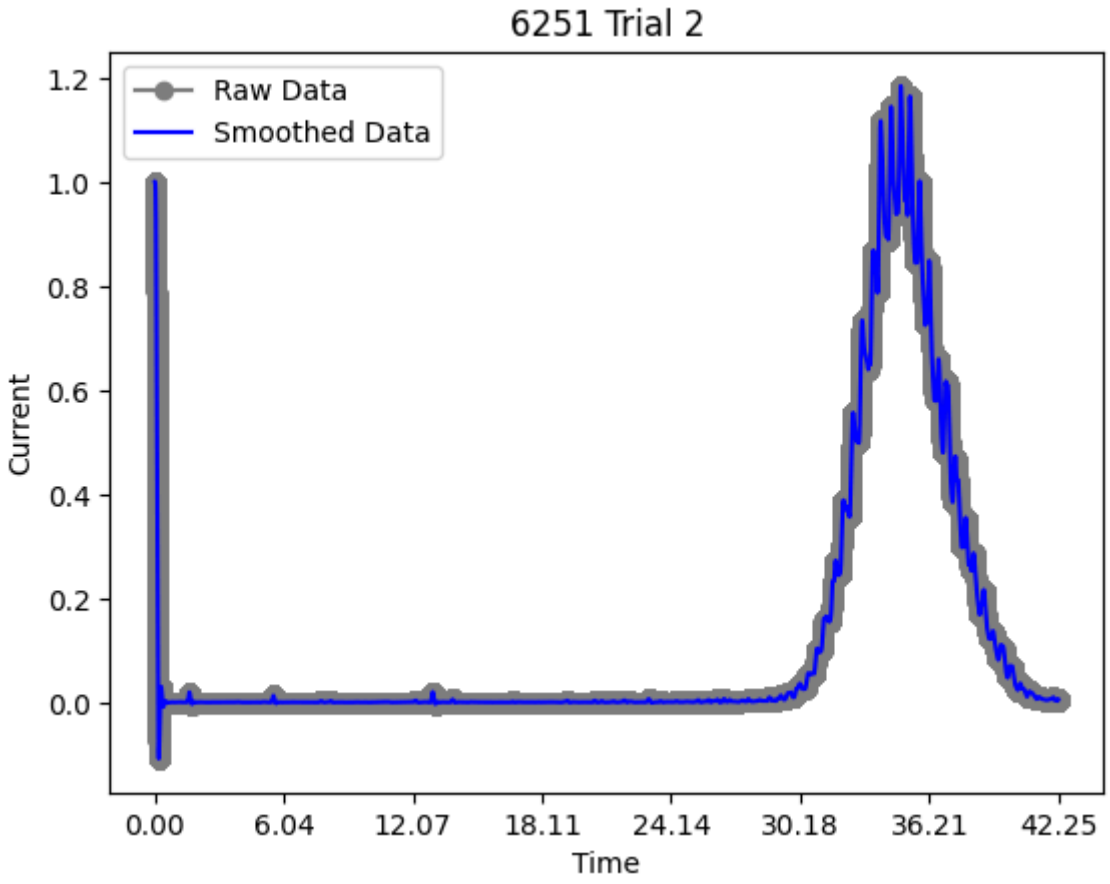
```
          0         1
0   0.00010  0.999987
1   0.00018  0.999987
2   0.00026  0.999987
3   0.00034  0.999987
4   0.00042  0.999987
```



In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# Load the CSV file
data = pd.read_csv('6251T2.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('6251 Trial 2')

num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

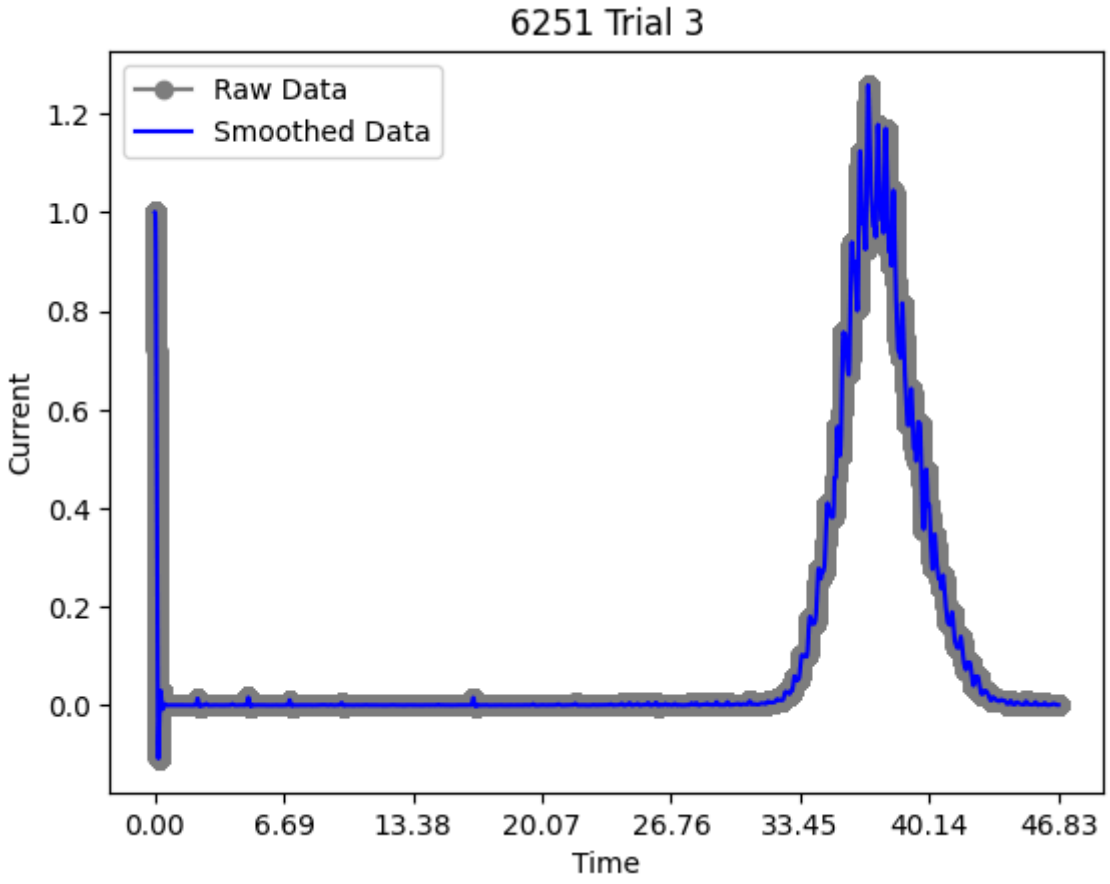
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [8]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6251T3.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('6251 Trial 3')
```

```
num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

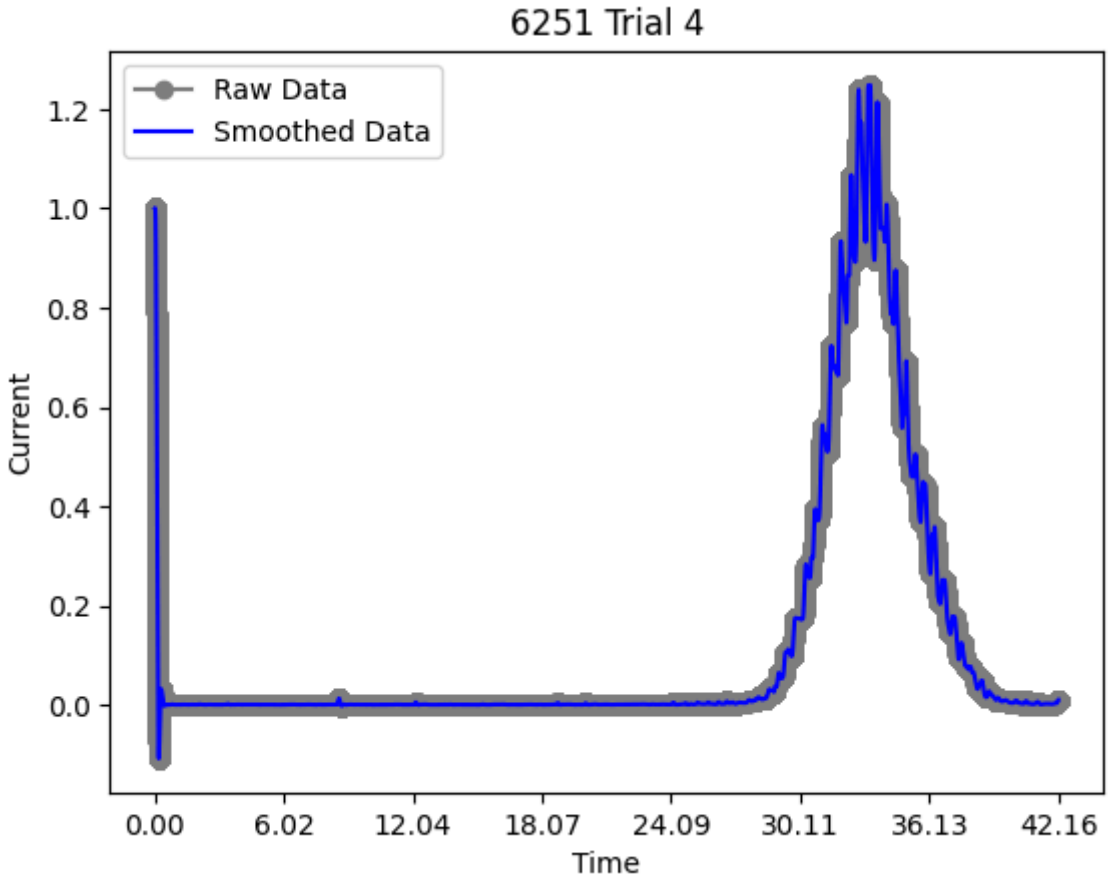
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [9]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6251T4.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('6251 Trial 4')

num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

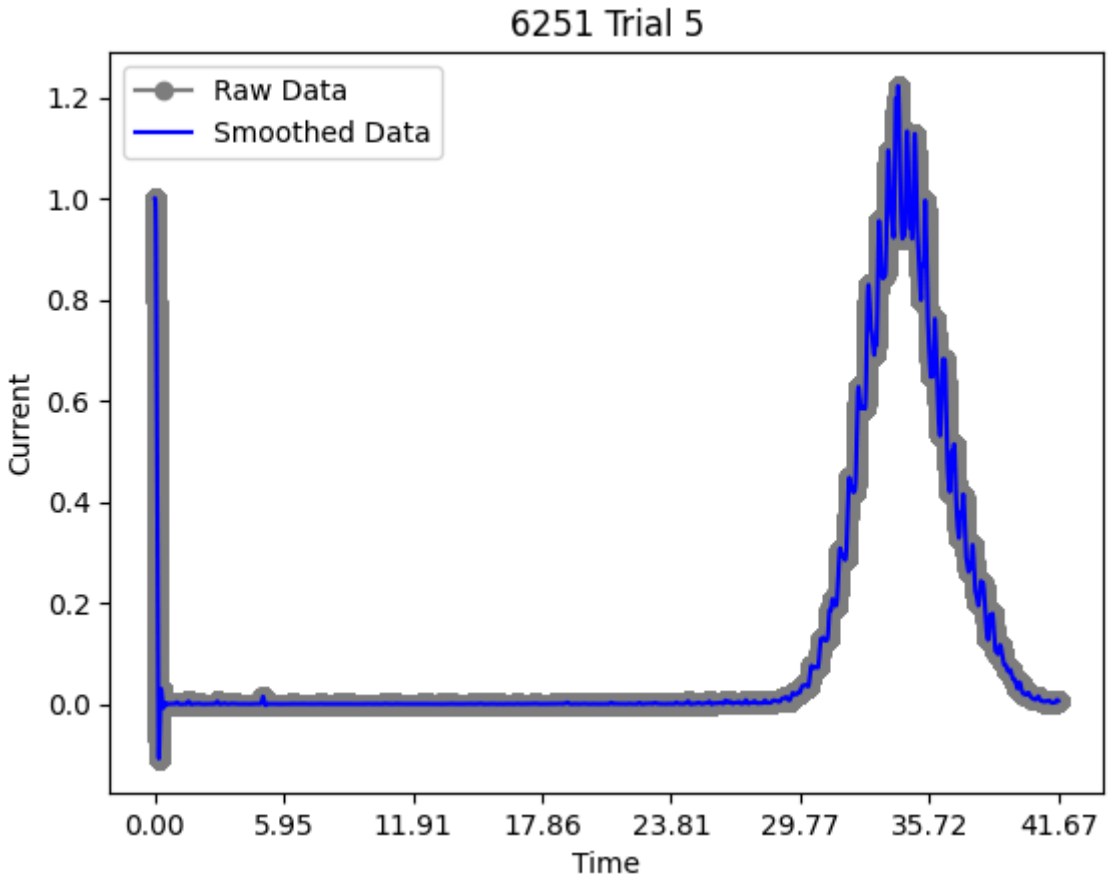
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# Load the CSV file
data = pd.read_csv('6251T5.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('6251 Trial 5')

num_ticks = 8  # Set the number of ticks you want
plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

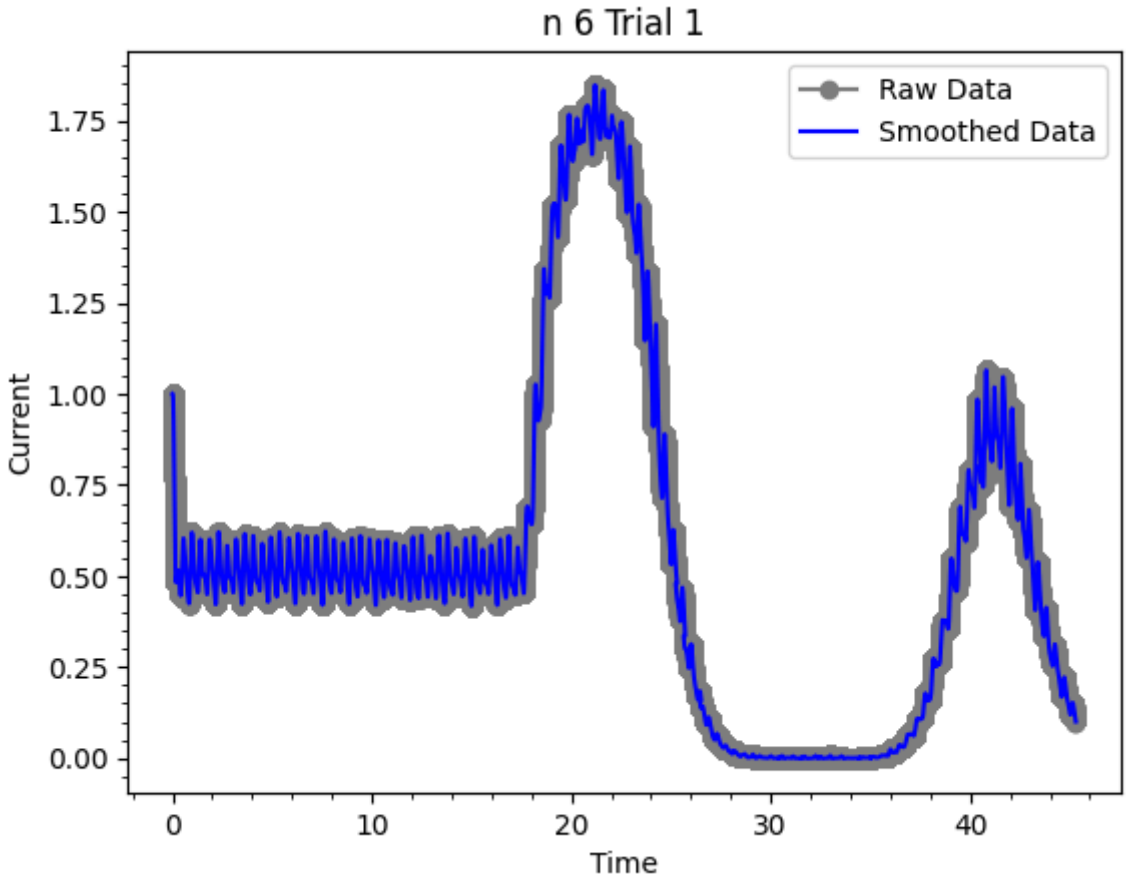
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [11]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6562D2T1BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 1')
```

```
#num_ticks = 8  # Set the number of ticks you want
#plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.minorticks_on()
plt.legend()
plt.show()
```

```
        0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

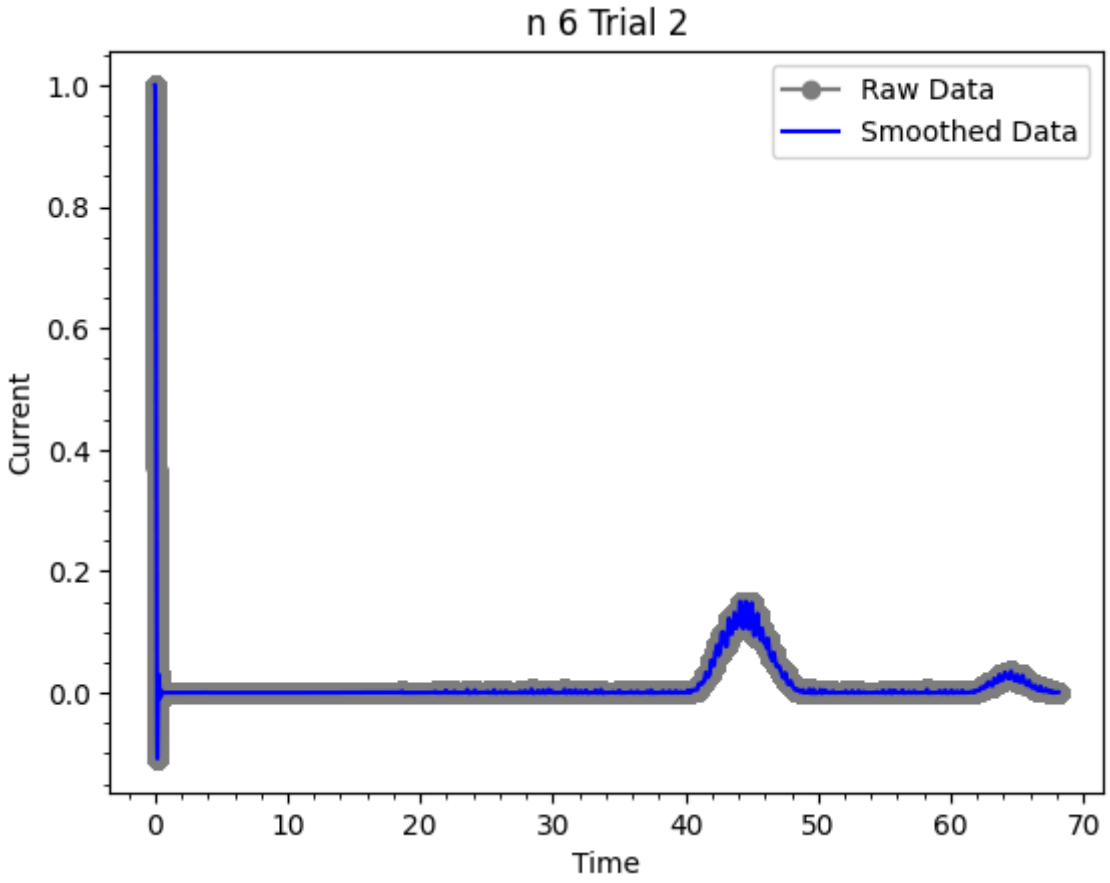


In [12]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6562D2T2BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
```

```
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 2')

plt.minorticks_on()
#plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.legend()
plt.show()
```

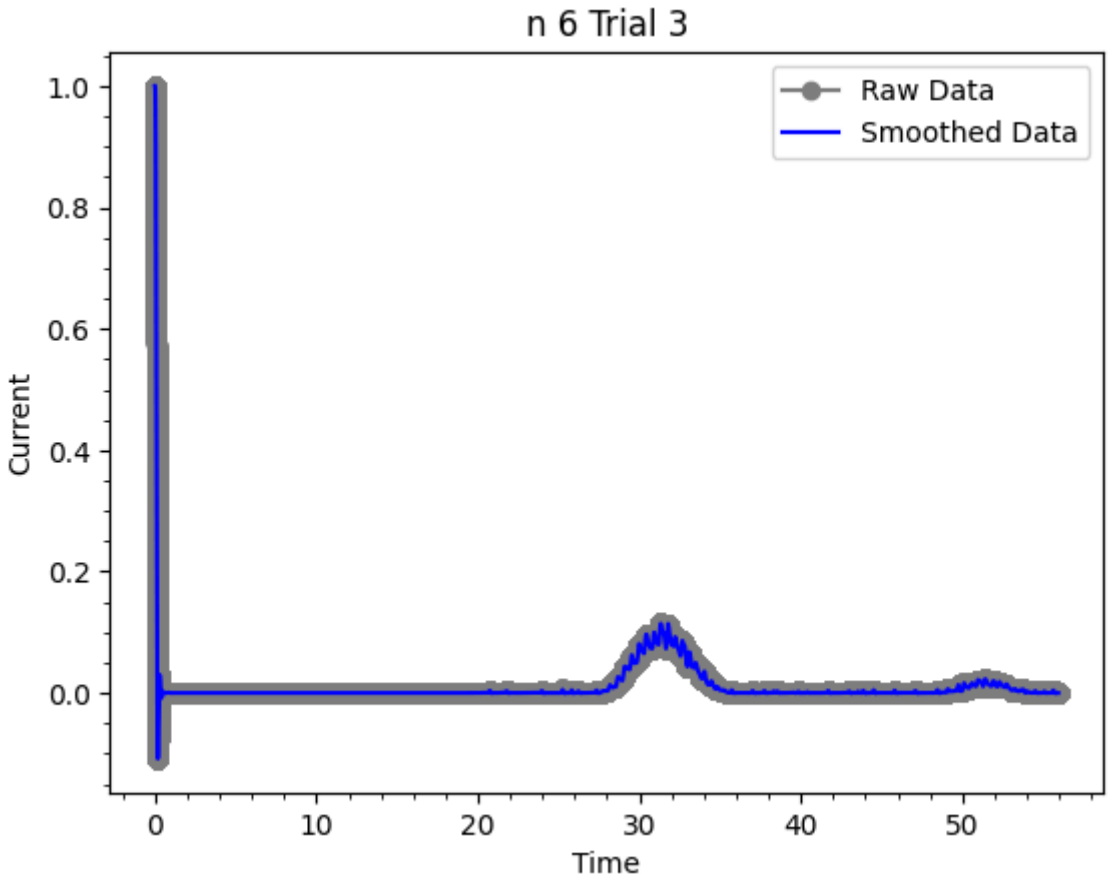
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [13]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# Load the CSV file
data = pd.read_csv('6562D2T3BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 3')

#num_ticks = 8  # Set the number of ticks you want
#plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.minorticks_on()
plt.legend()
plt.show()
```

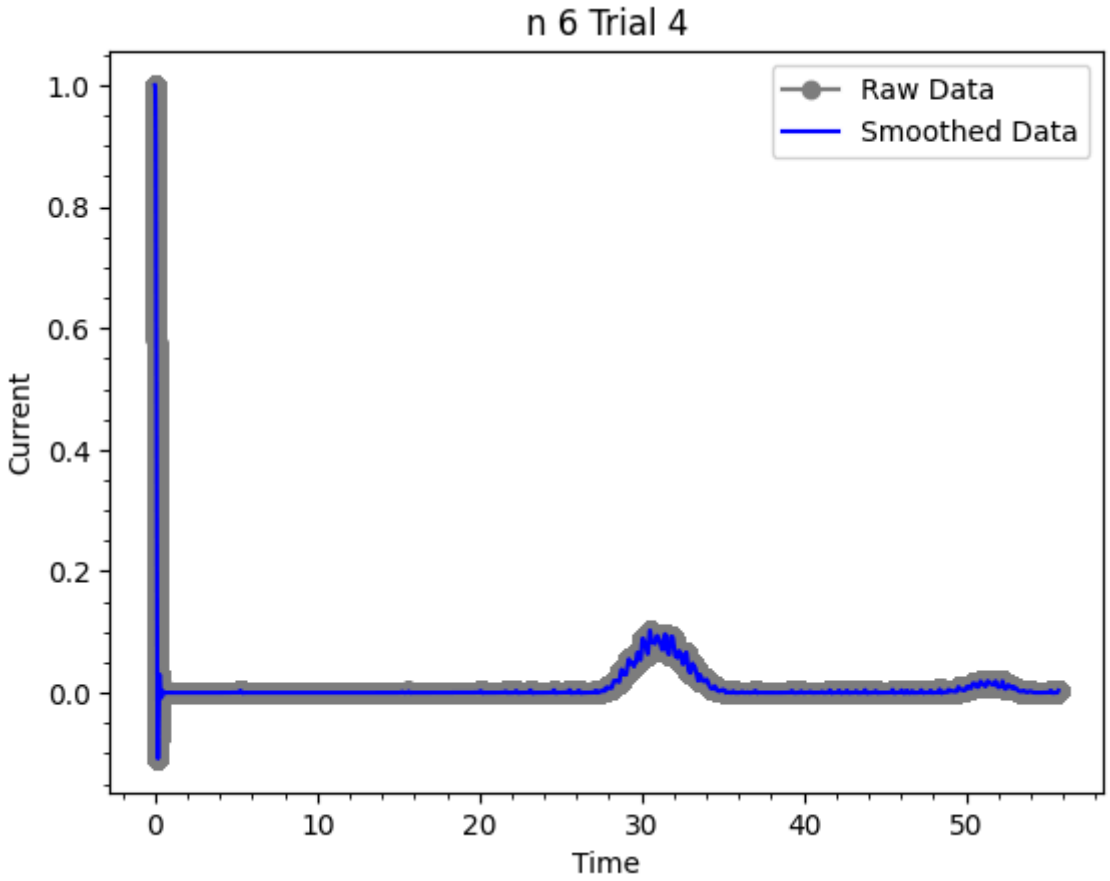
```
          0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6562D2T4BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 4')
```

```
#num_ticks = 8  # Set the number of ticks you want
#plt.xticks(np.linspace(x.min(), x.max(), num_ticks))
plt.minorticks_on()
plt.legend()
plt.show()
```

```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

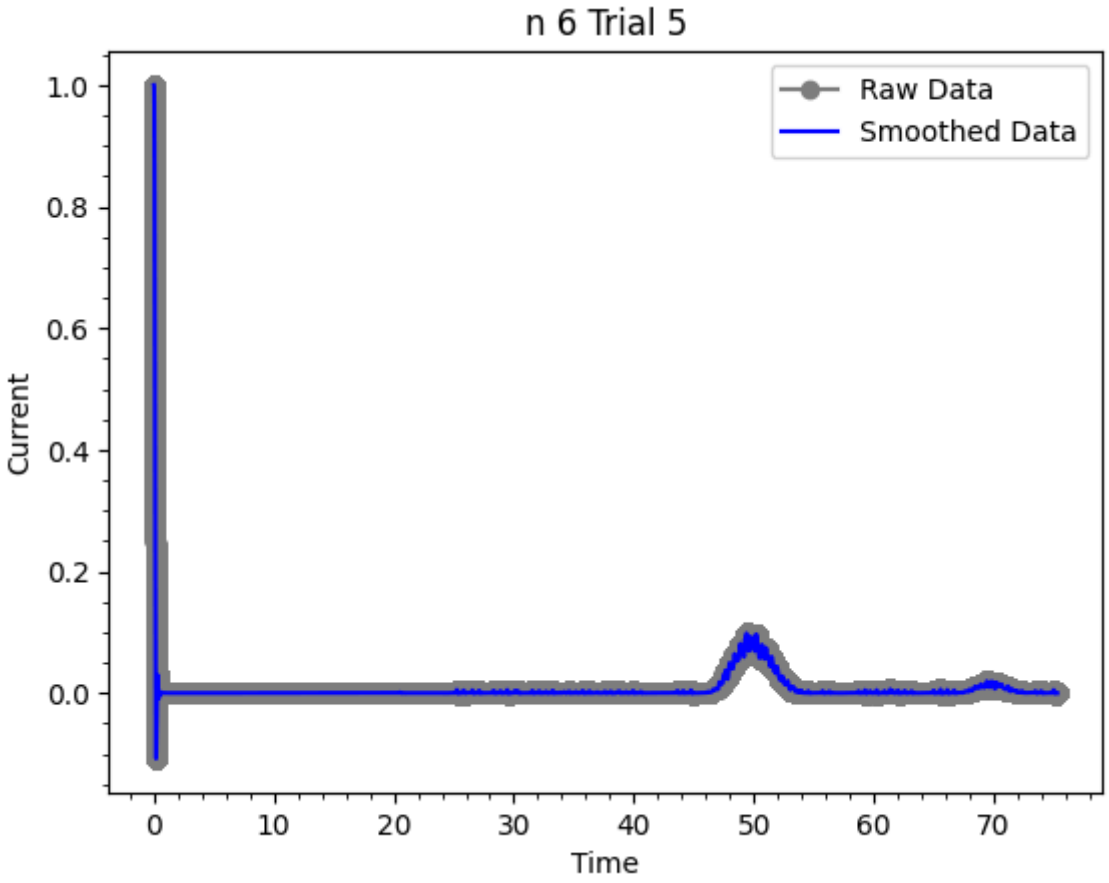


In [15]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('6562D2T5BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
```

```
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 5')

plt.minorticks_on()
plt.legend()
plt.show()
```

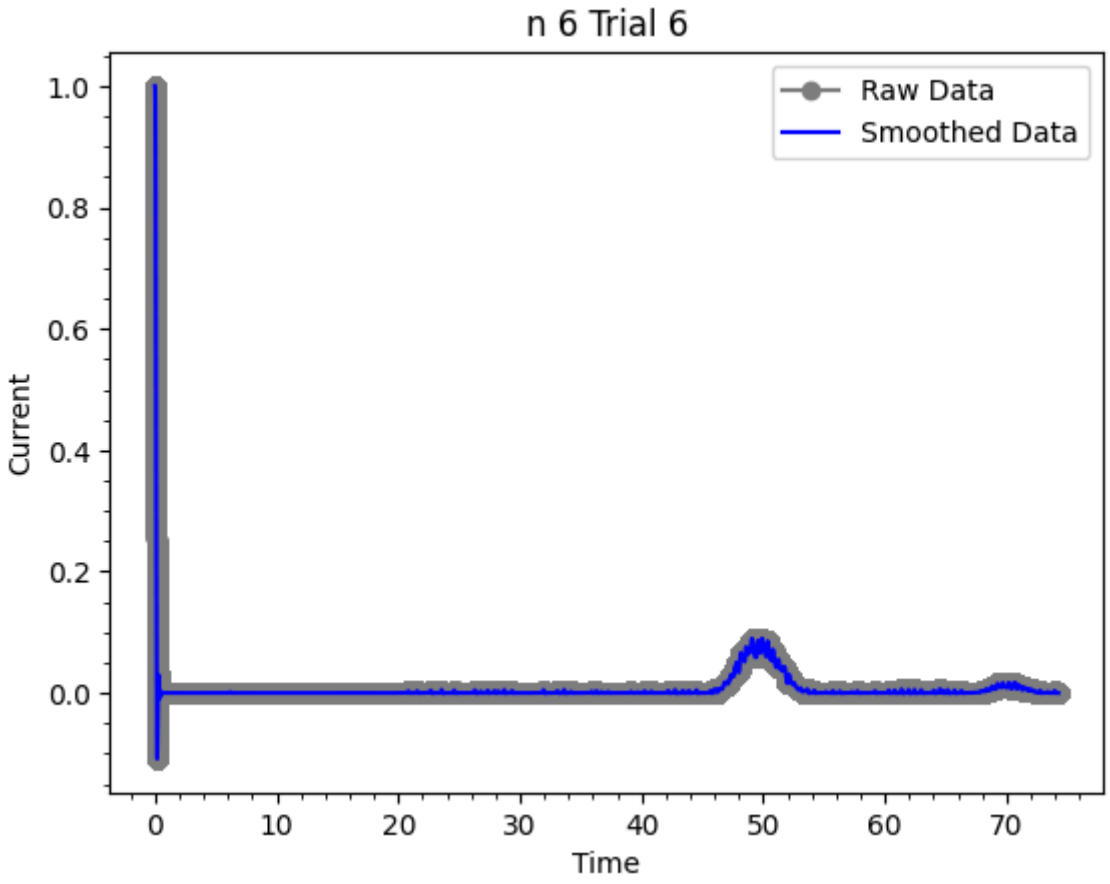
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# Load the CSV file
data = pd.read_csv('6562D2T6BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 6 Trial 6')

plt.minorticks_on()
plt.legend()
plt.show()
```

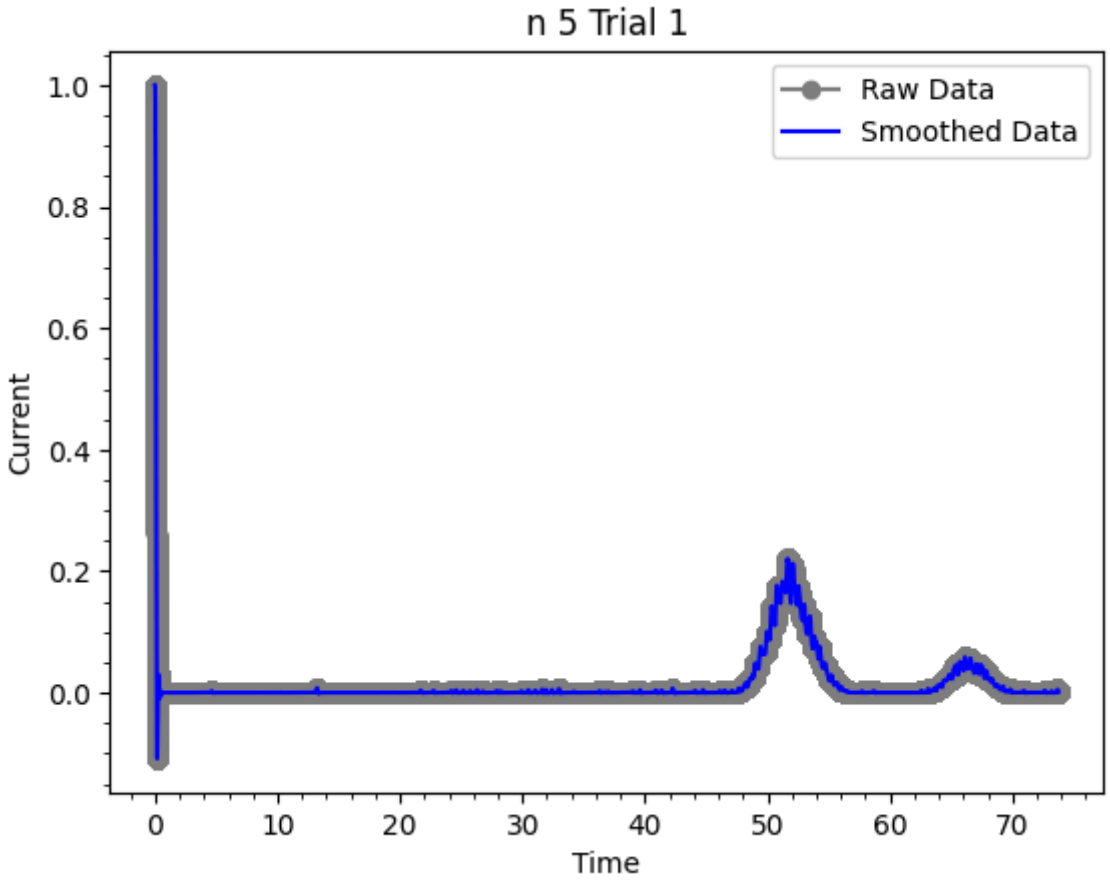
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [17]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4861D2T1BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 5 Trial 1')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

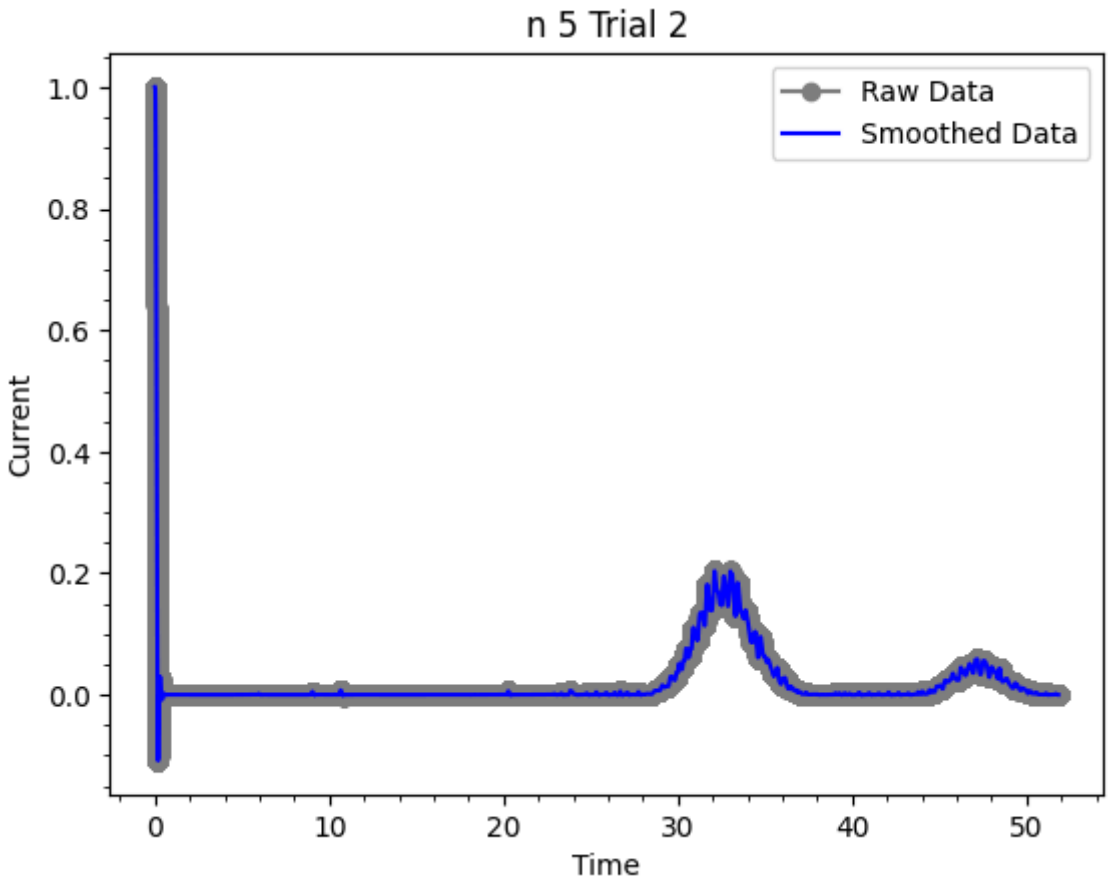
```
          0         1
0   0.00010  0.999987
1   0.00018  0.999987
2   0.00026  0.999987
3   0.00034  0.999987
4   0.00042  0.999987
```



In [18]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4861D2T2BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 5 Trial 2')

plt.minorticks_on()
plt.legend()
plt.show()
```

```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

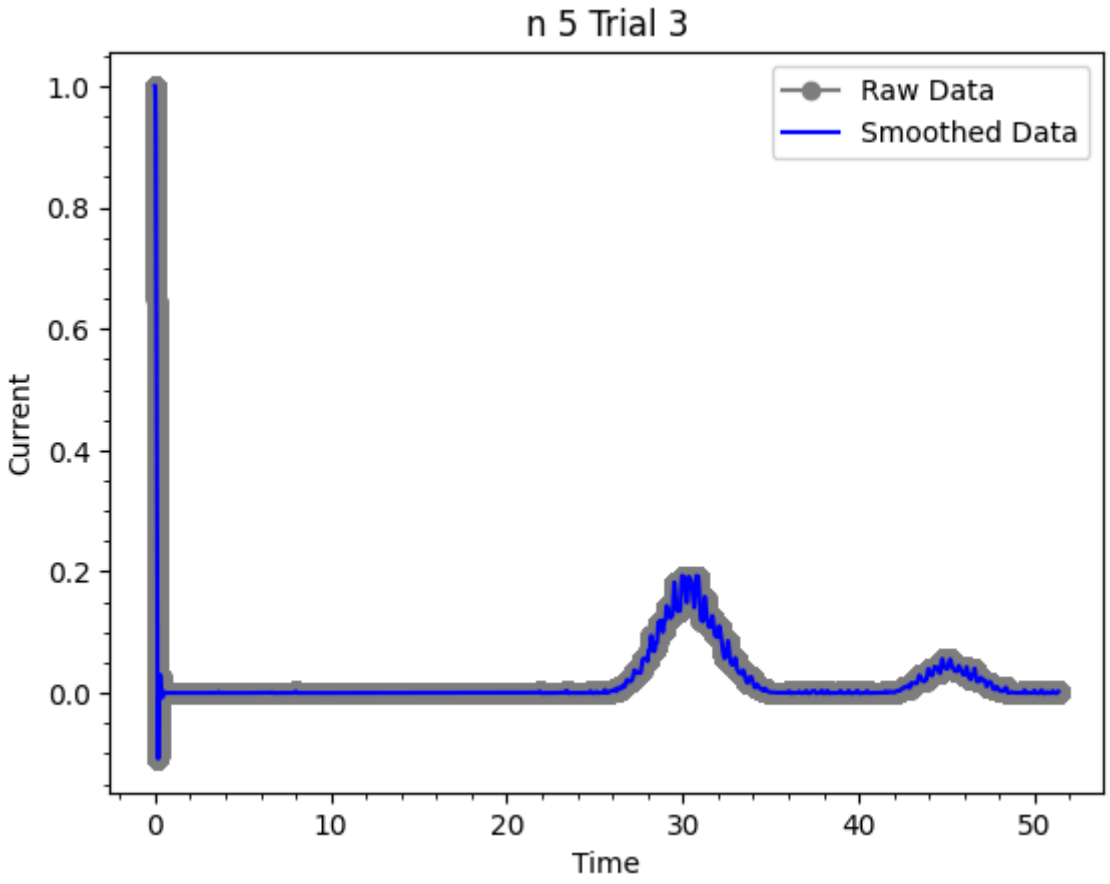


In [19]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4861D2T3BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 5 Trial 3')

plt.minorticks_on()
plt.legend()
plt.show()
```

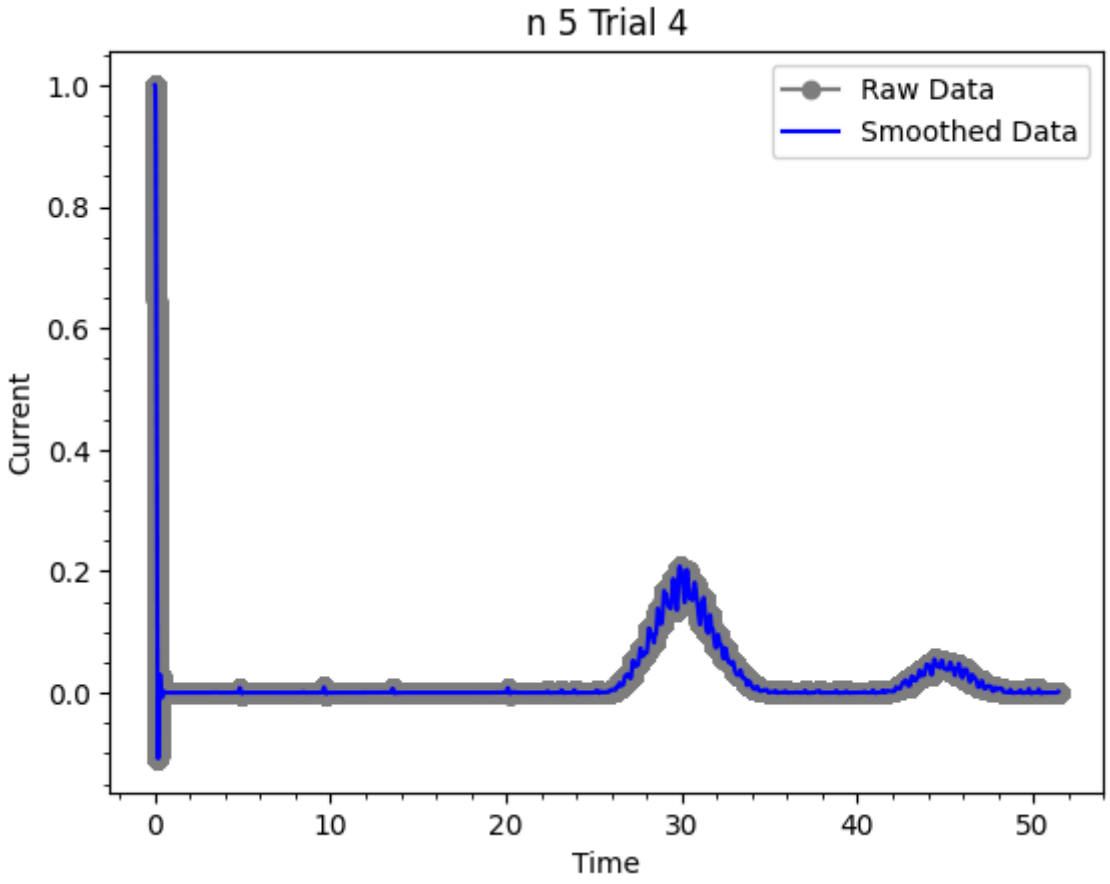
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [20]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4861D2T4BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 5 Trial 4')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```

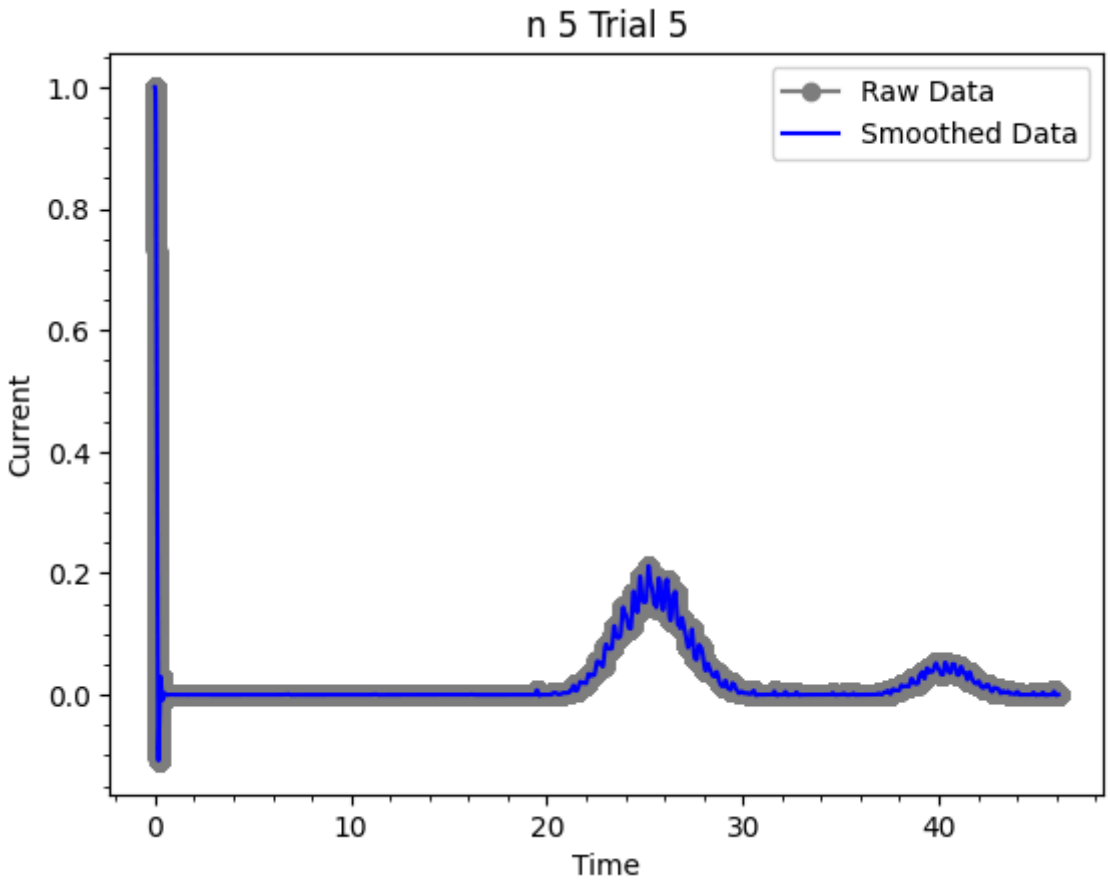


In [21]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4861D2T5BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 5 Trial 5')

plt.minorticks_on()
plt.legend()
plt.show()
```

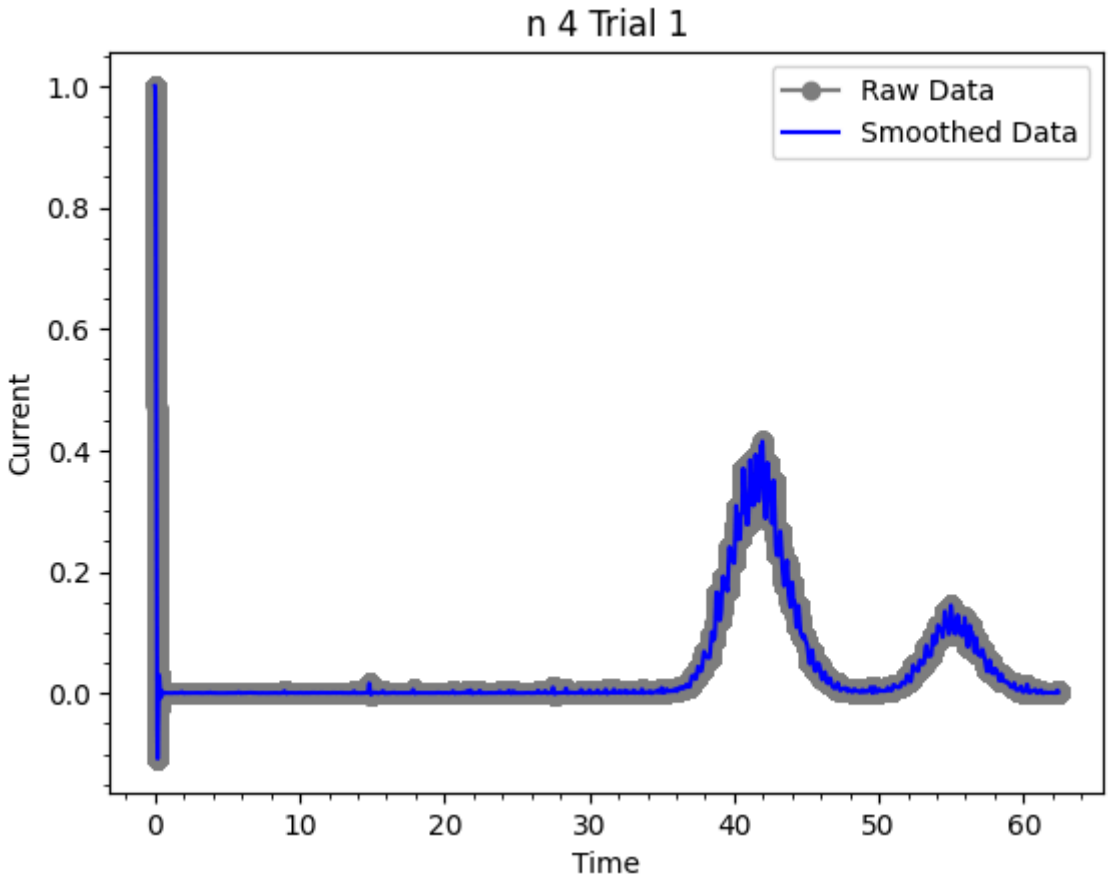
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4340D2T1BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 4 Trial 1')

plt.minorticks_on()
plt.legend()
plt.show()
```

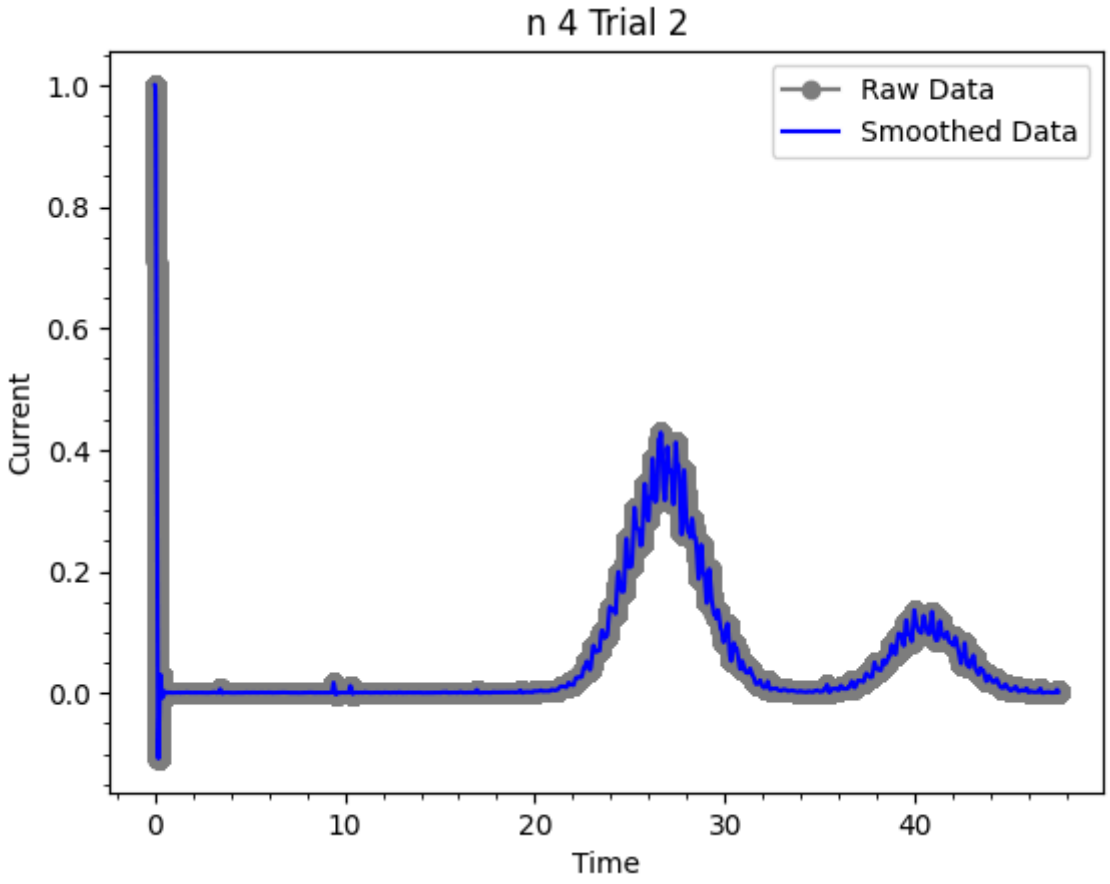
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4340D2T2BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 4 Trial 2')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

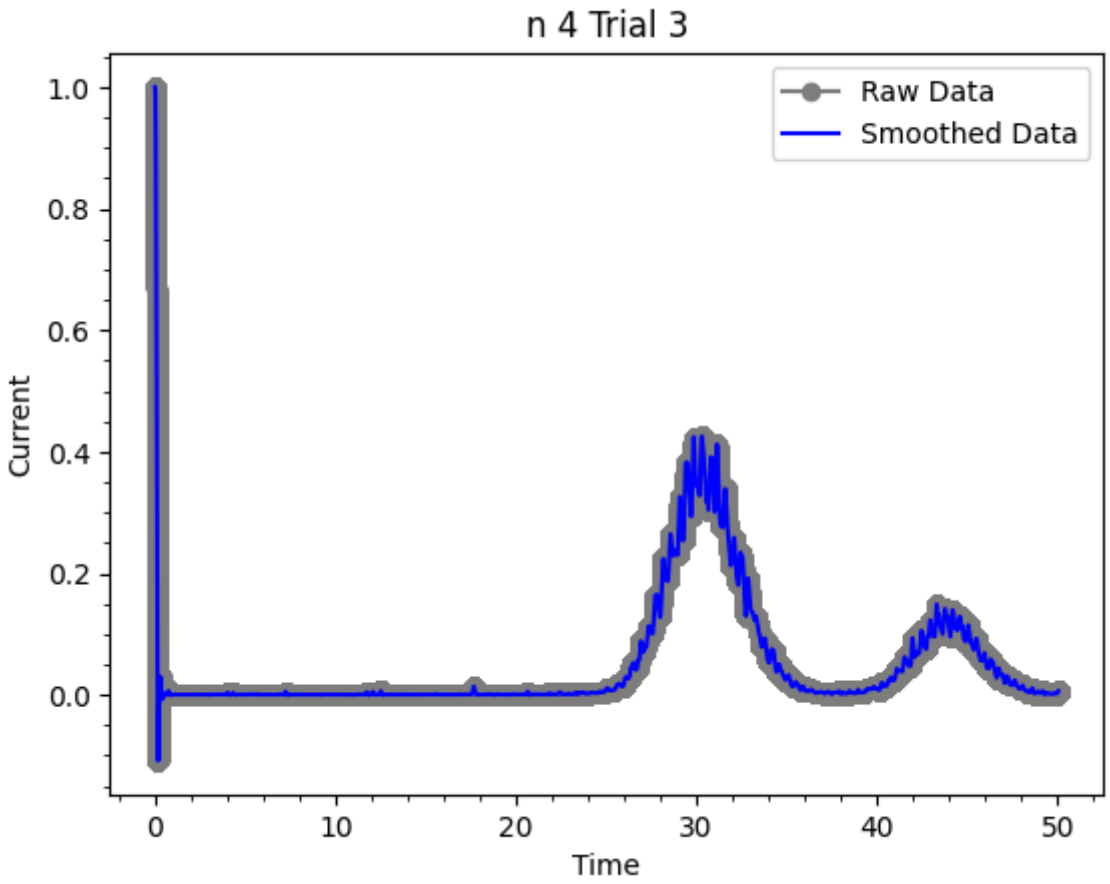


In [24]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4340D2T3BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 4 Trial 3')

plt.minorticks_on()
plt.legend()
plt.show()
```

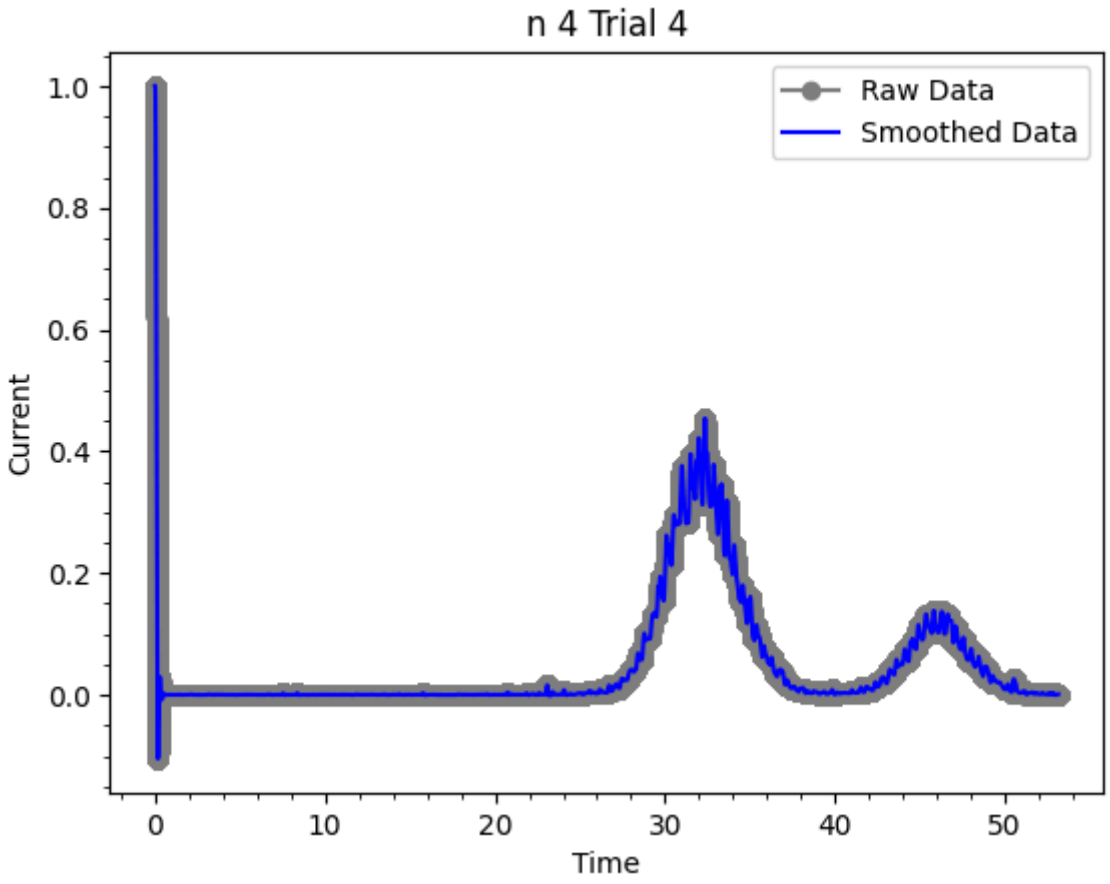
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [25]:
```
#One of my trials was lost due to me uploading the wrong file
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
```

```
data = pd.read_csv('4340D2T5BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 4 Trial 4')

plt.minorticks_on()
plt.legend()
plt.show()
```

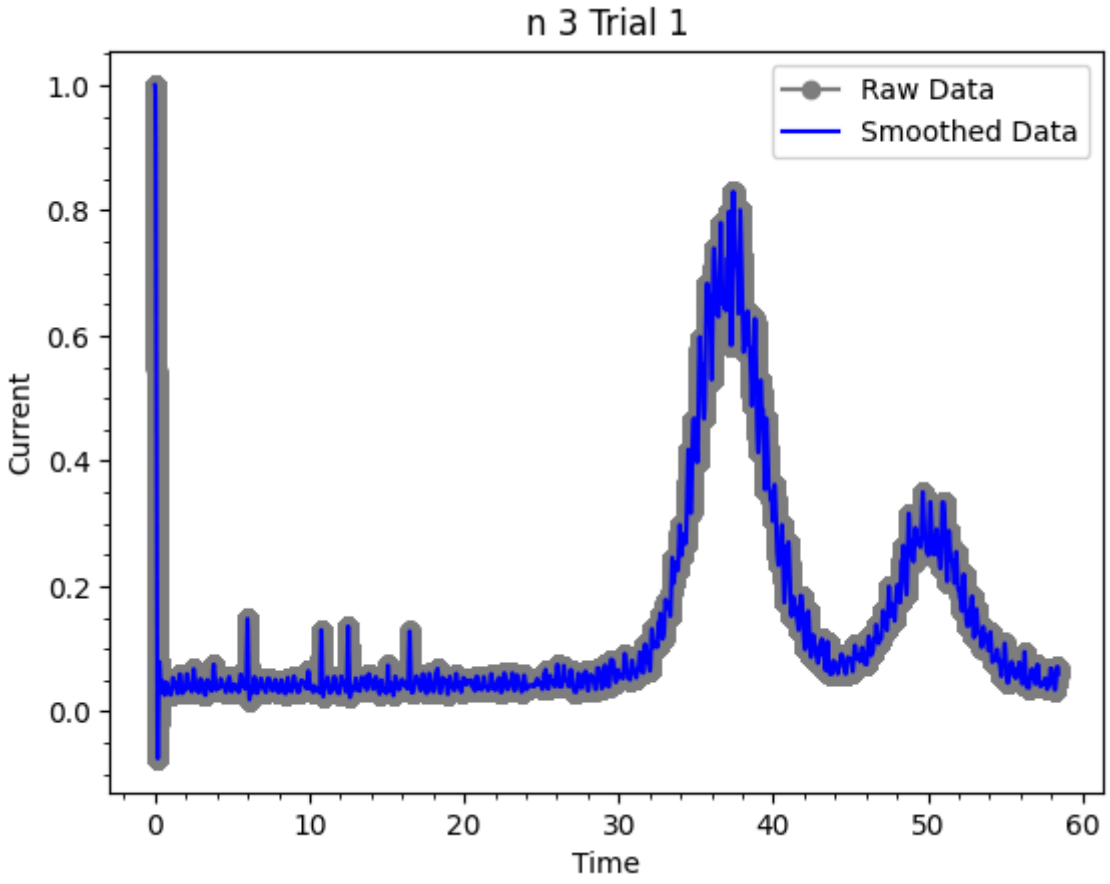
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [26]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4101D2T1BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 3 Trial 1')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
          0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

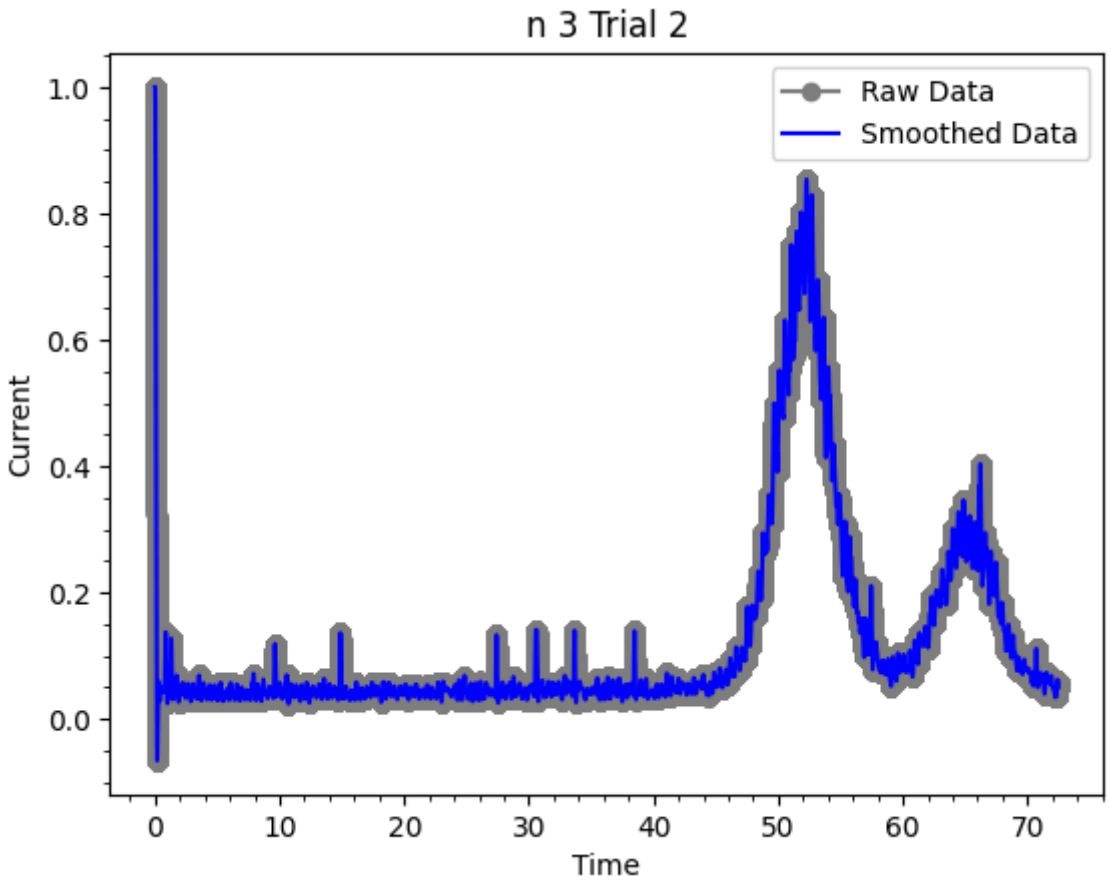


In [27]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4101D2T2BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 3 Trial 2')

plt.minorticks_on()
plt.legend()
plt.show()
```

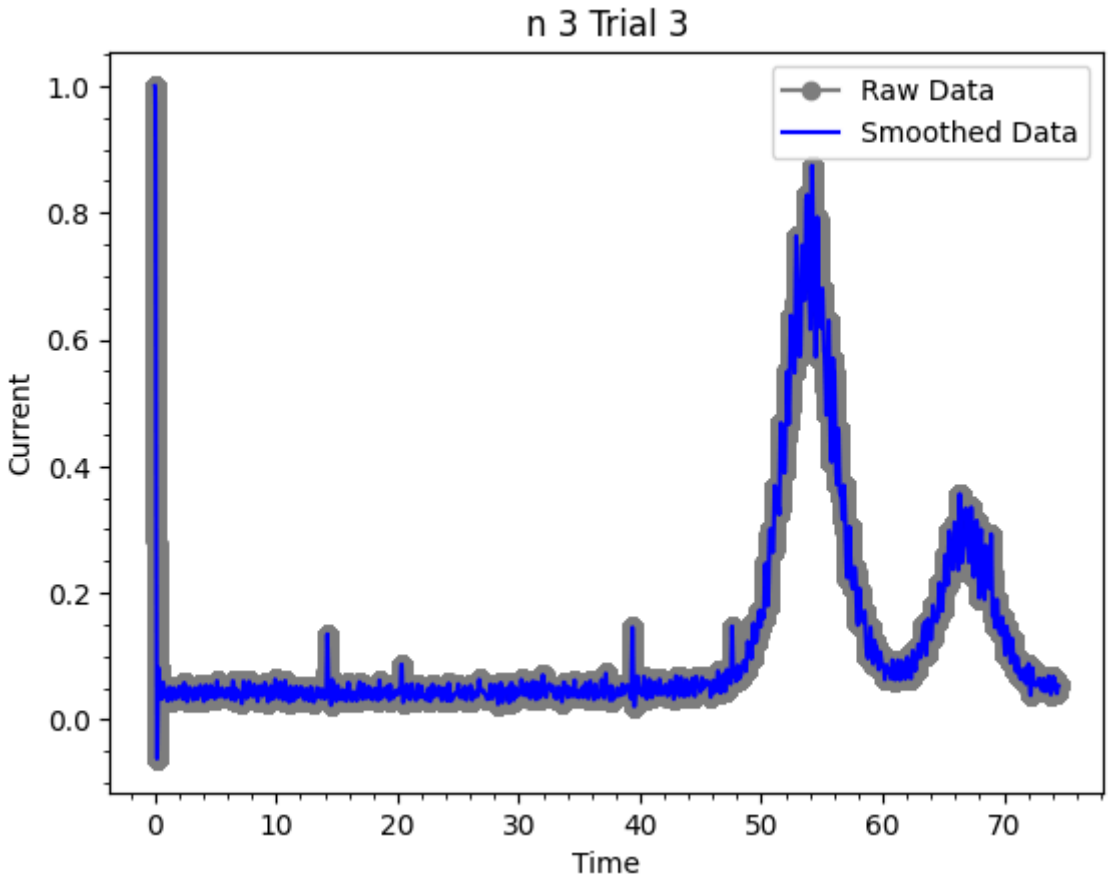
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [28]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4101D2T3BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 3 Trial 3')

plt.minorticks_on()
plt.legend()
plt.show()
```

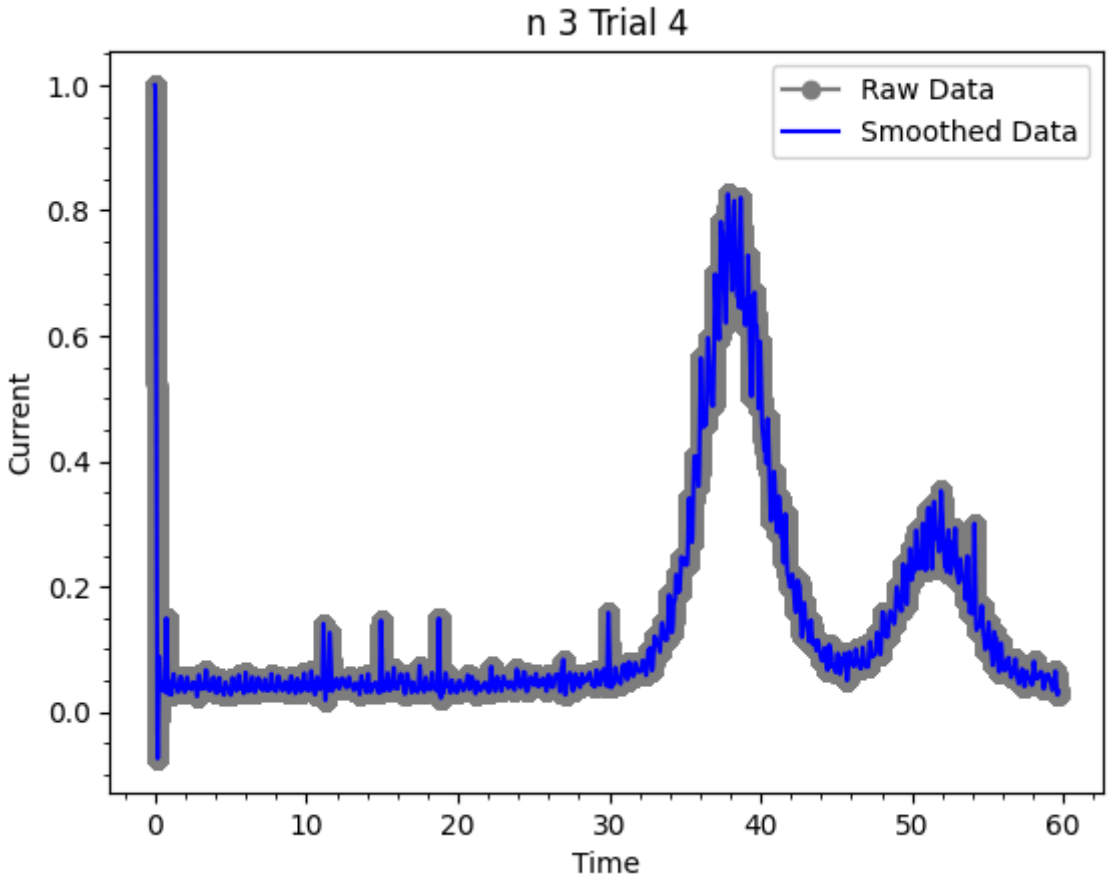
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [29]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4101D2T4BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 3 Trial 4')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```

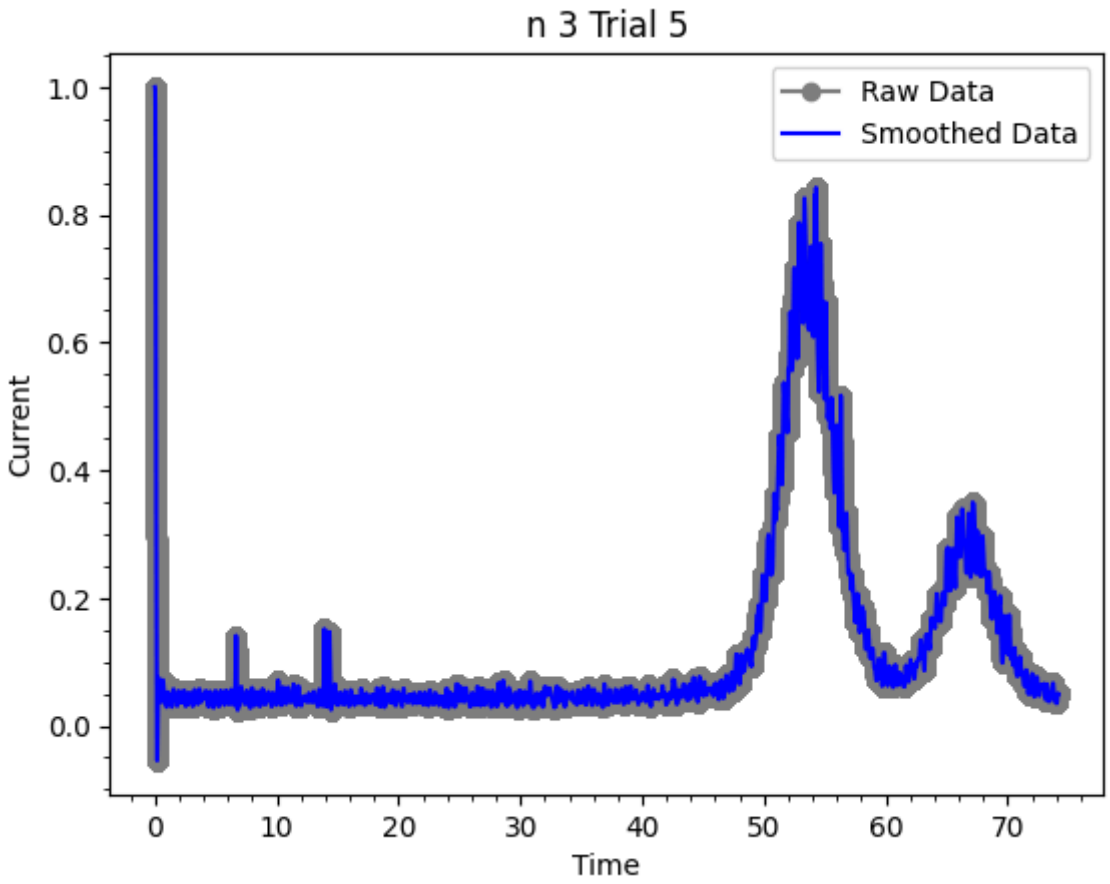


In [30]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('4101D2T5BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 3 Trial 5')

plt.minorticks_on()
plt.legend()
plt.show()
```

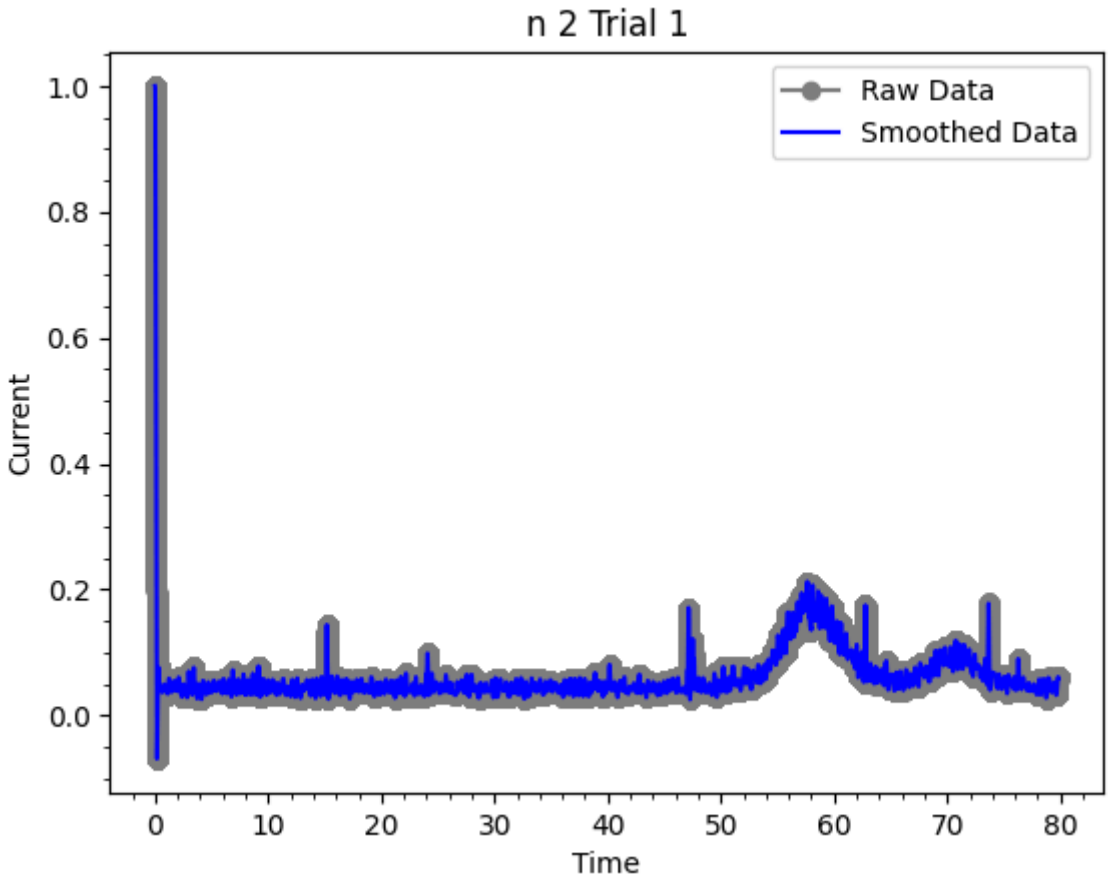
```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```



In [31]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3970D2T1BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 2 Trial 1')

plt.minorticks_on()
plt.legend()
plt.show()
```

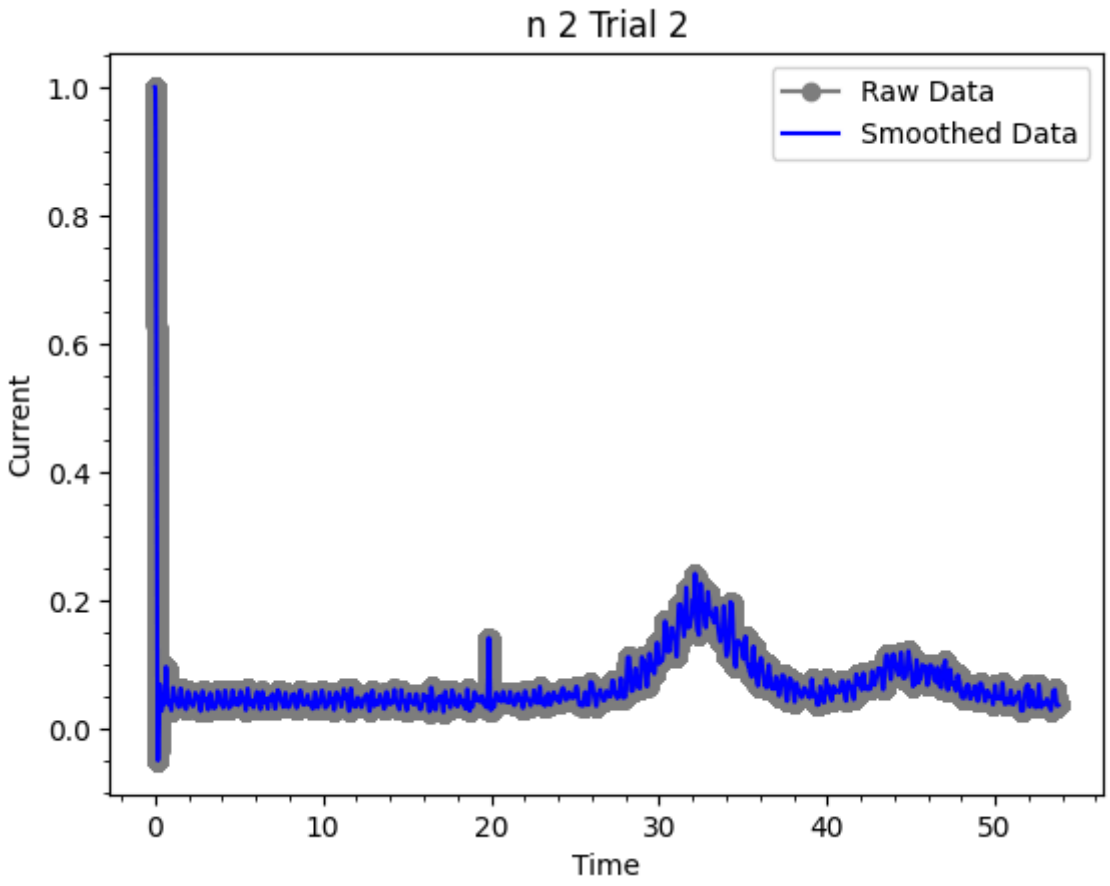
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [32]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3970D2T2BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 2 Trial 2')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
          0         1
0  0.000096  0.999987
1  0.000176  0.999987
2  0.000256  0.999987
3  0.000336  0.999987
4  0.000416  0.999987
```

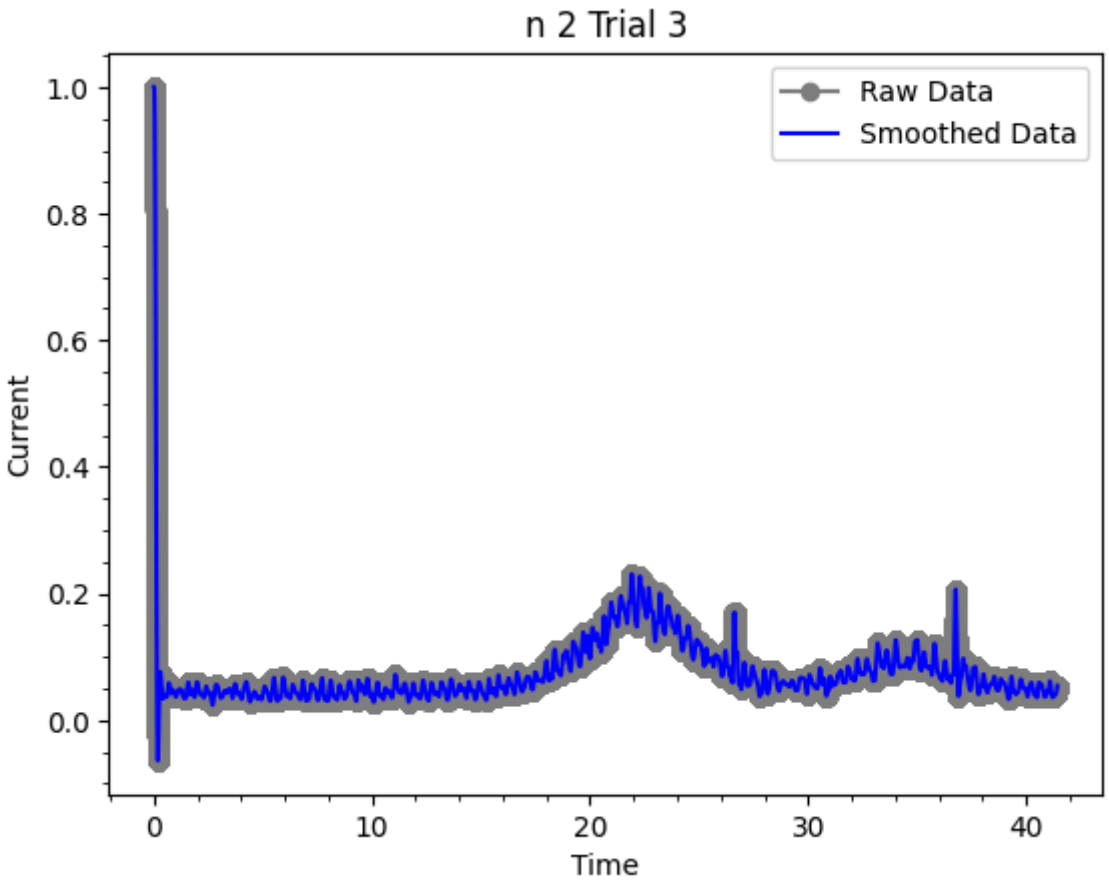


In [33]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3970D2T3.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 2 Trial 3')

plt.minorticks_on()
plt.legend()
plt.show()
```

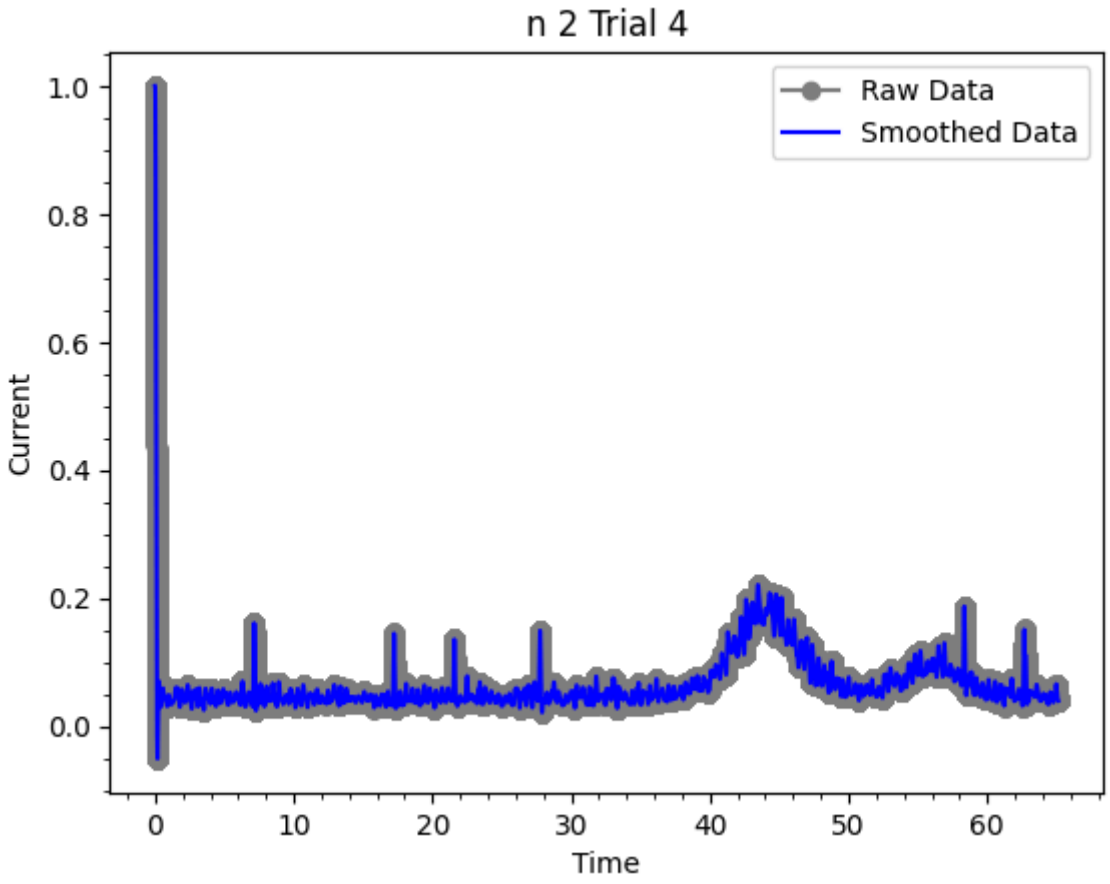
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [34]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3970D2T4BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 2 Trial 4')

plt.minorticks_on()
plt.legend()
plt.show()
```

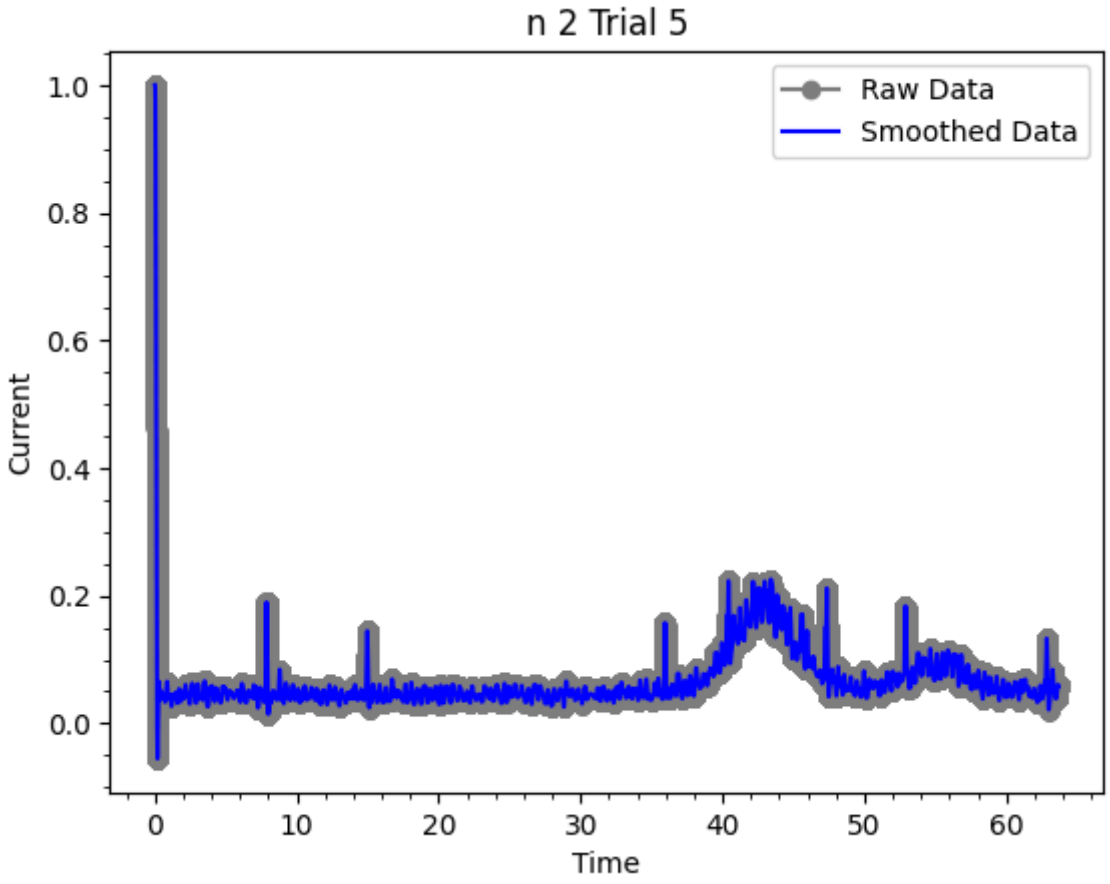
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [35]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3970D2T5BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 2 Trial 5')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
          0         1
0   0.00010  0.999987
1   0.00018  0.999987
2   0.00026  0.999987
3   0.00034  0.999987
4   0.00042  0.999987
```

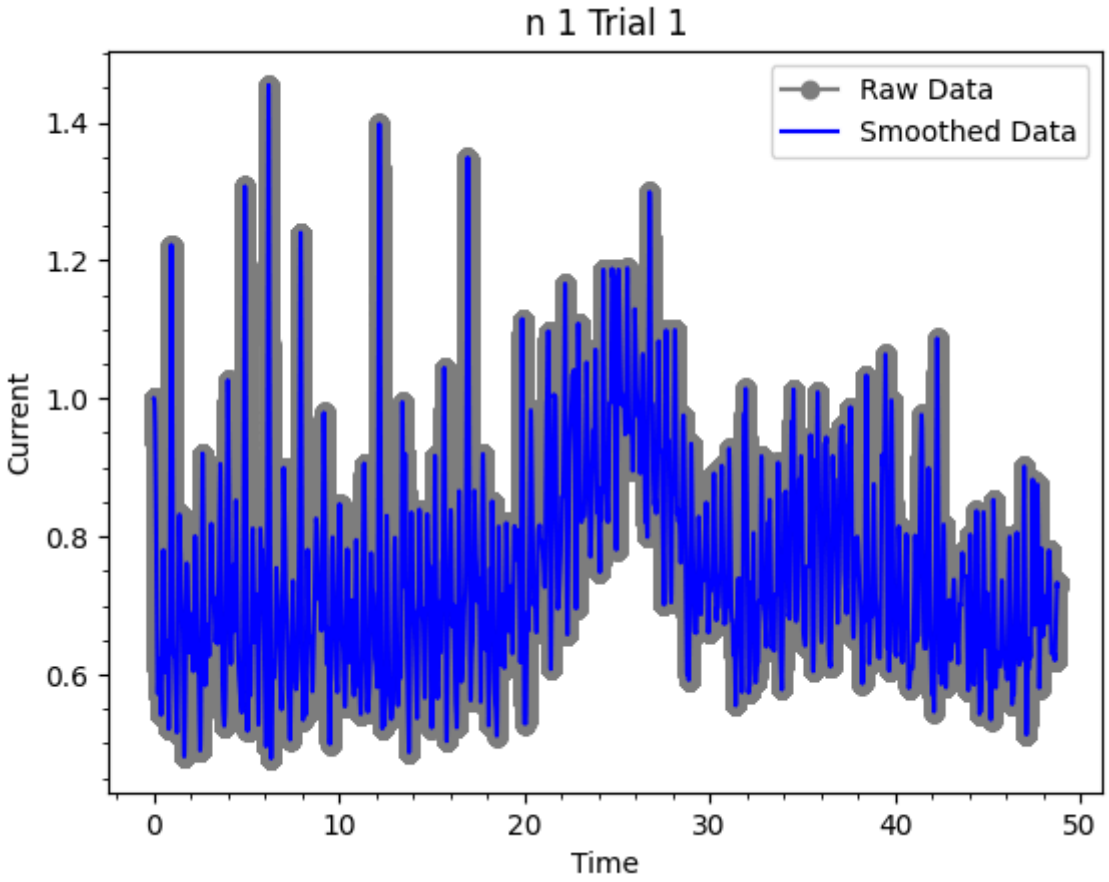


In [36]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3889D2T1BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 1 Trial 1')

plt.minorticks_on()
plt.legend()
plt.show()
```

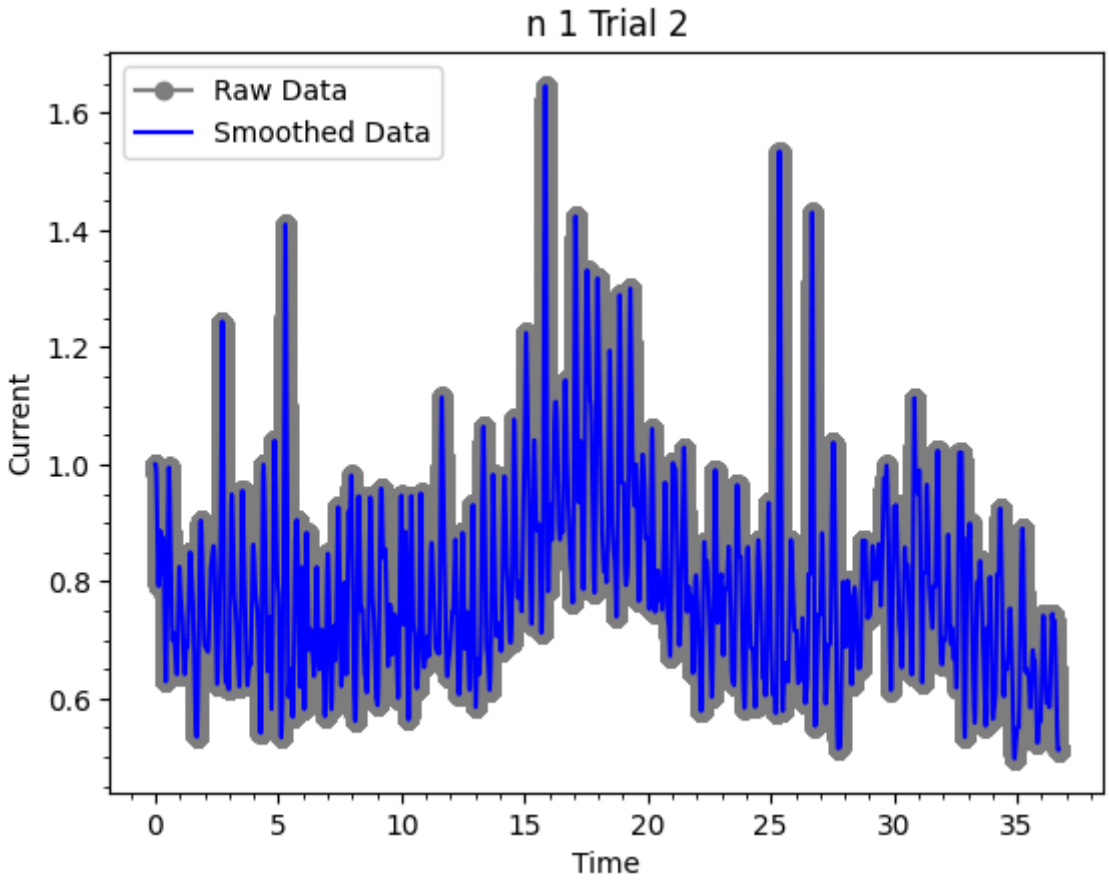
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [37]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3889D2T2BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 1 Trial 2')

plt.minorticks_on()
plt.legend()
plt.show()
```

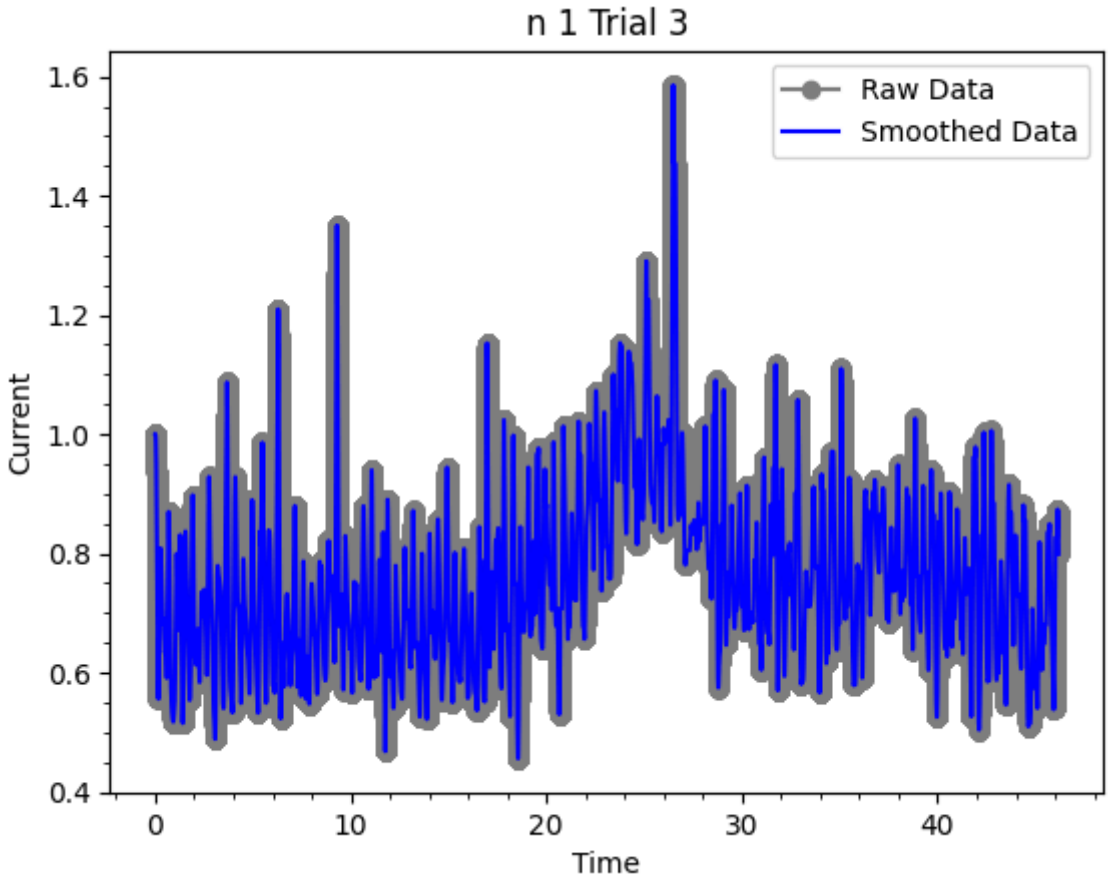
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [38]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3889D2T3BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 1 Trial 3')
```

```
plt.minorticks_on()
plt.legend()
plt.show()
```

```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```

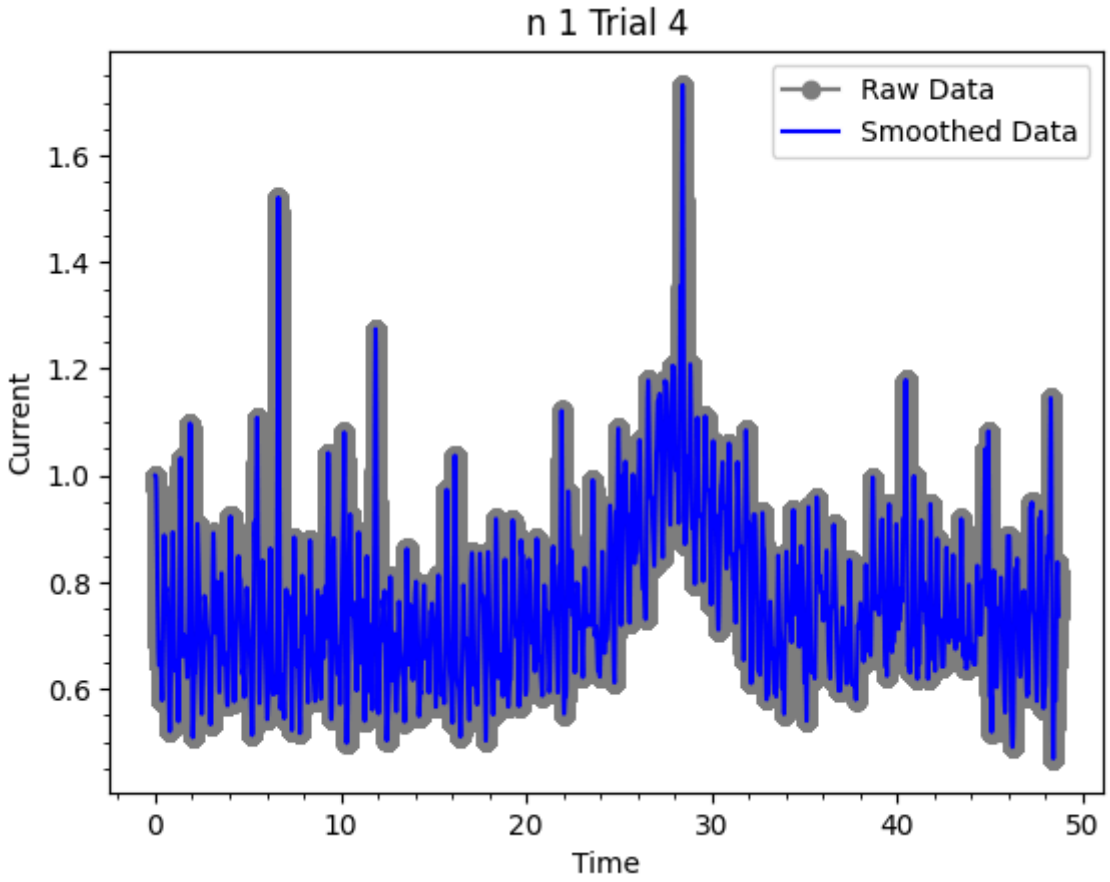


In [39]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3889D2T4BM.csv', header=None)


print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')
```

```
# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 1 Trial 4')

plt.minorticks_on()
plt.legend()
plt.show()
```

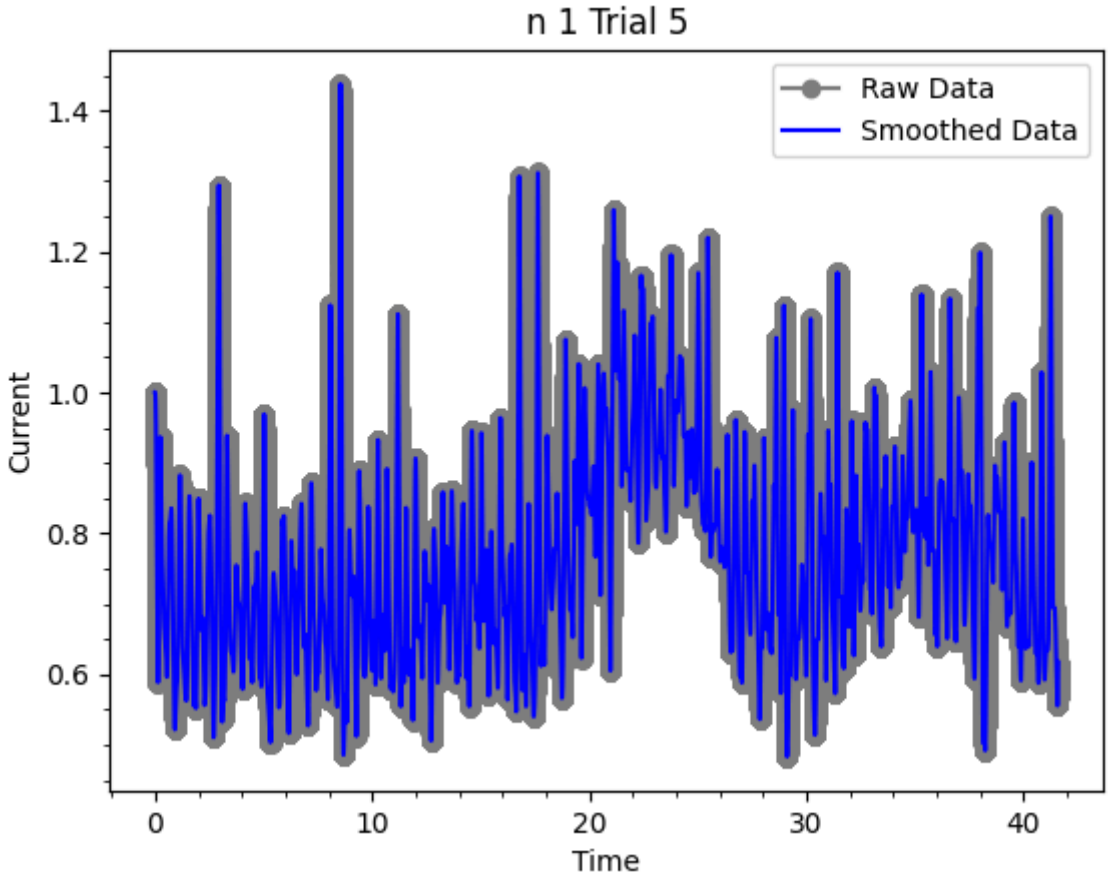
```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```



In [40]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the CSV file
data = pd.read_csv('3889D2T5BM.csv', header=None)
```

```
print(data.head())


x = data[0]
y = data[1]

# Plot the raw data
plt.plot(x, y, label='Raw Data', color='gray', marker='o')

# Apply a smoothing algorithm (moving average)
window_size = 5  # Adjust window size for smoothing
y_smooth = y.rolling(window=window_size, center=True).mean()


plt.plot(x, y_smooth, label='Smoothed Data', color='blue')


plt.xlabel('Time')
plt.ylabel('Current')
plt.title('n 1 Trial 5')

plt.minorticks_on()
plt.legend()
plt.show()
```

```
         0         1
0  0.00010  0.999987
1  0.00018  0.999987
2  0.00026  0.999987
3  0.00034  0.999987
4  0.00042  0.999987
```


n 1 Trial 5
Raw Data
Smoothed Data
Current
Time
1.4
1.2
1.0
0.8
0.6
0
10
20
30
40

In [ ]: