Cooldown of smaller ring sample on flag to check if sample Super conducts. And also to check if the temp stability improves with twisted pair wire all the way through to the sample contacts.
The outer cable was replaced with shielded twisted pair for better performance.

In [1]:

```
import matplotlib.pyplot as plt
%matplotlib inline
#%matplotlib qt
plt.rcParams.update({'font.size': 20})
plt.rcParams['figure.figsize'] = [15, 12]
```

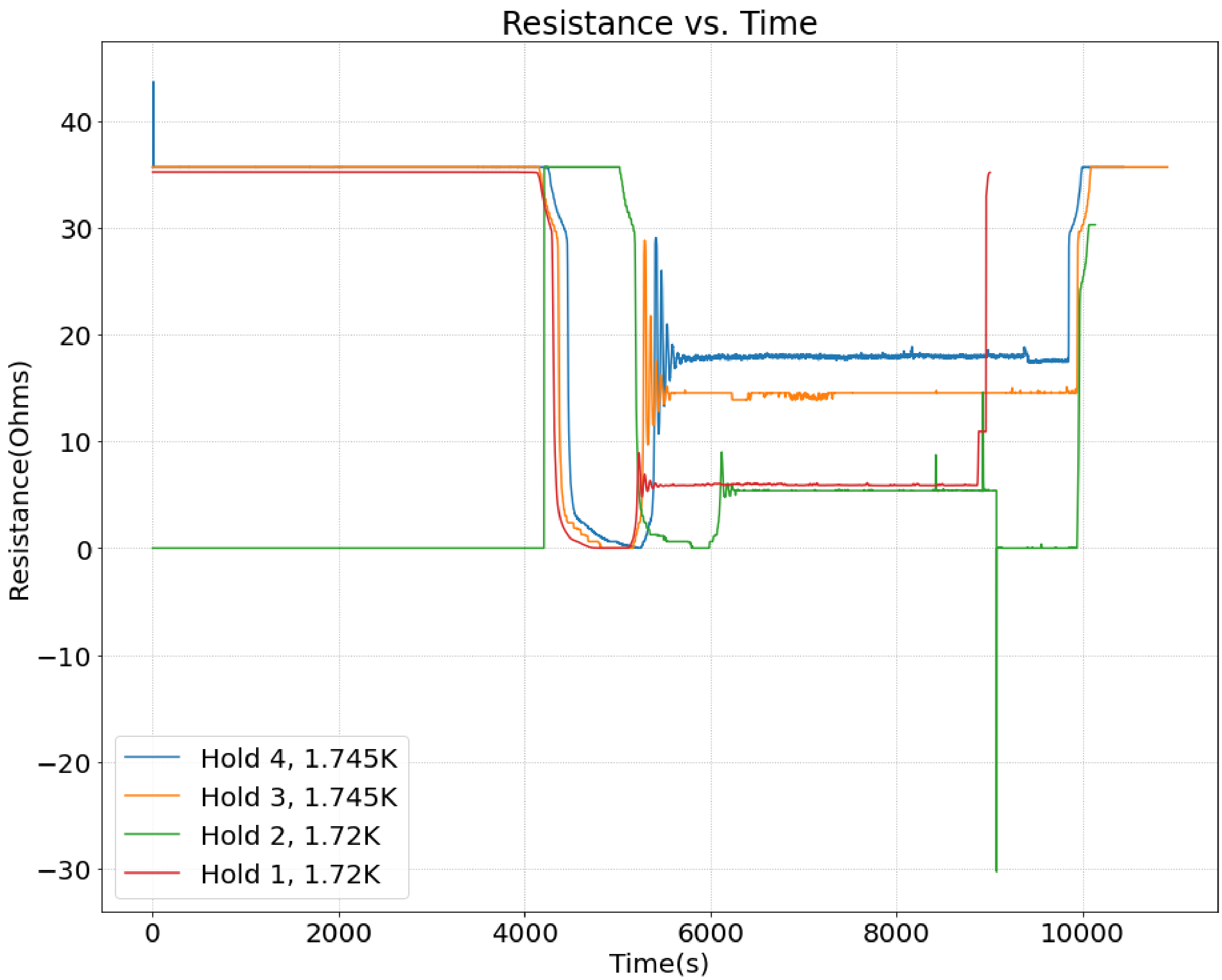
I did 4 LP temperature holds till 2_12_2023 for cooldwon 21, in all these case I had a 100K Ohm resistance in series.\n
1st Hold: Cooldown21_2_2_2023_LP_control TC of 3s, 24dB slope 1.72K, C Wing SRS830, 1V, SENS 500uV.
2nd Hold: Cooldown21_2_9_2023_LP_Control, TC of 1s, 24dB slope 1.72K, Basement SRS830, 0.25V, SENS 100uV.
3rd Hold: Cooldown21_2_10_2023_LP_Control, TC of 1s, 24dB slope 1.745K, Basement SRS830, 0.25V, SENS 100uV.
4th Hold: Cooldown21_2_11_2023_LP_Control, TC of 300ms, 24dB slope 1.745K, Basement SRS830, 0.25V, SENS 100uV.

We are now going to see the time traces of the resistance while the temperature of the pot is held at different values.

In [2]:

```
import numpy as np
#time is coloumn 0
tme = np.genfromtxt("Cooldown21_2_11_2023_LP_Control.dat",usecols = 0)

#pot temperature
temp = np.genfromtxt("Cooldown21_2_11_2023_LP_Control.dat",usecols = 5)

#resistance of the sample is coloumn 9
res = np.genfromtxt("Cooldown21_2_11_2023_LP_Control.dat",usecols = 9)

#flag temperature
ftemp = np.genfromtxt("Cooldown21_2_11_2023_LP_Control.dat",usecols = 11)

for i in range(len(tme)):
    res[i] = 10000000*res[i]/25

plt.grid(linestyle = ':')
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')

plt.title('Resistance vs. Time')

plt.plot(tme,res,label = 'Hold 4, 1.745K')


tme2 = np.genfromtxt("Cooldown21_2_10_2023_LP_control.dat",usecols = 0)

#pot temperature
temp2 = np.genfromtxt("Cooldown21_2_10_2023_LP_control.dat",usecols = 5)

#resistance of the sample is coloumn 9
res2 = np.genfromtxt("Cooldown21_2_10_2023_LP_control.dat",usecols = 9)

#flag temperature
ftemp2 = np.genfromtxt("Cooldown21_2_10_2023_LP_control.dat",usecols = 11)
```

```python
for i in range(len(tme2)):
    res2[i] = 10000000*res2[i]/25

plt.grid(linestyle = ':')
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')

plt.title('Resistance vs. Time')

plt.plot(tme2,res2, label = 'Hold 3, 1.745K')

tme19 = np.genfromtxt("Cooldown21_2_9_2023_LP_Control.dat",usecols = 0)

#pot temperature
temp19 = np.genfromtxt("Cooldown21_2_9_2023_LP_Control.dat",usecols = 5)

#resistance of the sample is coloumn 9
res19 = np.genfromtxt("Cooldown21_2_9_2023_LP_Control.dat",usecols = 9)

#flag temperature
ftemp19 = np.genfromtxt("Cooldown21_2_9_2023_LP_Control.dat",usecols = 11)
for i in range(len(tme19)):
    res19[i] = 10000000*res19[i]/25

plt.grid(linestyle = ':')
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')

plt.title('Resistance vs. Time')

plt.plot(tme19,res19, label = 'Hold 2, 1.72K')


tme3 = np.genfromtxt("Cooldown21_2_2_2023_LP_Control.dat",usecols = 0)

#pot temperature
temp3 = np.genfromtxt("Cooldown21_2_2_2023_LP_Control.dat",usecols = 5)

#resistance of the sample is coloumn 9
res3 = np.genfromtxt("Cooldown21_2_2_2023_LP_Control.dat",usecols = 9)

#flag temperature
ftemp3 = np.genfromtxt("Cooldown21_2_2_2023_LP_Control.dat",usecols = 11)
for i in range(len(tme3)):
    res3[i] = 10000000*res3[i]/100

plt.grid(linestyle = ':')
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')

plt.title('Resistance vs. Time')

plt.plot(tme3,res3, label = 'Hold 1, 1.72K')







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



For better viewing we are gonna plot the graphs from the same time.

In [3]:

```
res1 = []
tme1 = []
temp1 = []
ftemp1 = []
k = 0
for i in range(len(tme)):
    if temp[i] < 2:
        tme1.append(k)
        res1.append(res[i])
        temp1.append(temp[i])
        ftemp1.append(ftemp[i])
        k = k + 2.5
```

In [4]:

```
res191 = []
tme191 = []
temp191 = []
ftemp191 = []
k = 0
for i in range(len(tme19)):
    if temp19[i] < 2:
        tme191.append(k)
        res191.append(res19[i])
        temp191.append(temp19[i])
        ftemp191.append(ftemp19[i])
        k = k + 2.5
```

In [5]:

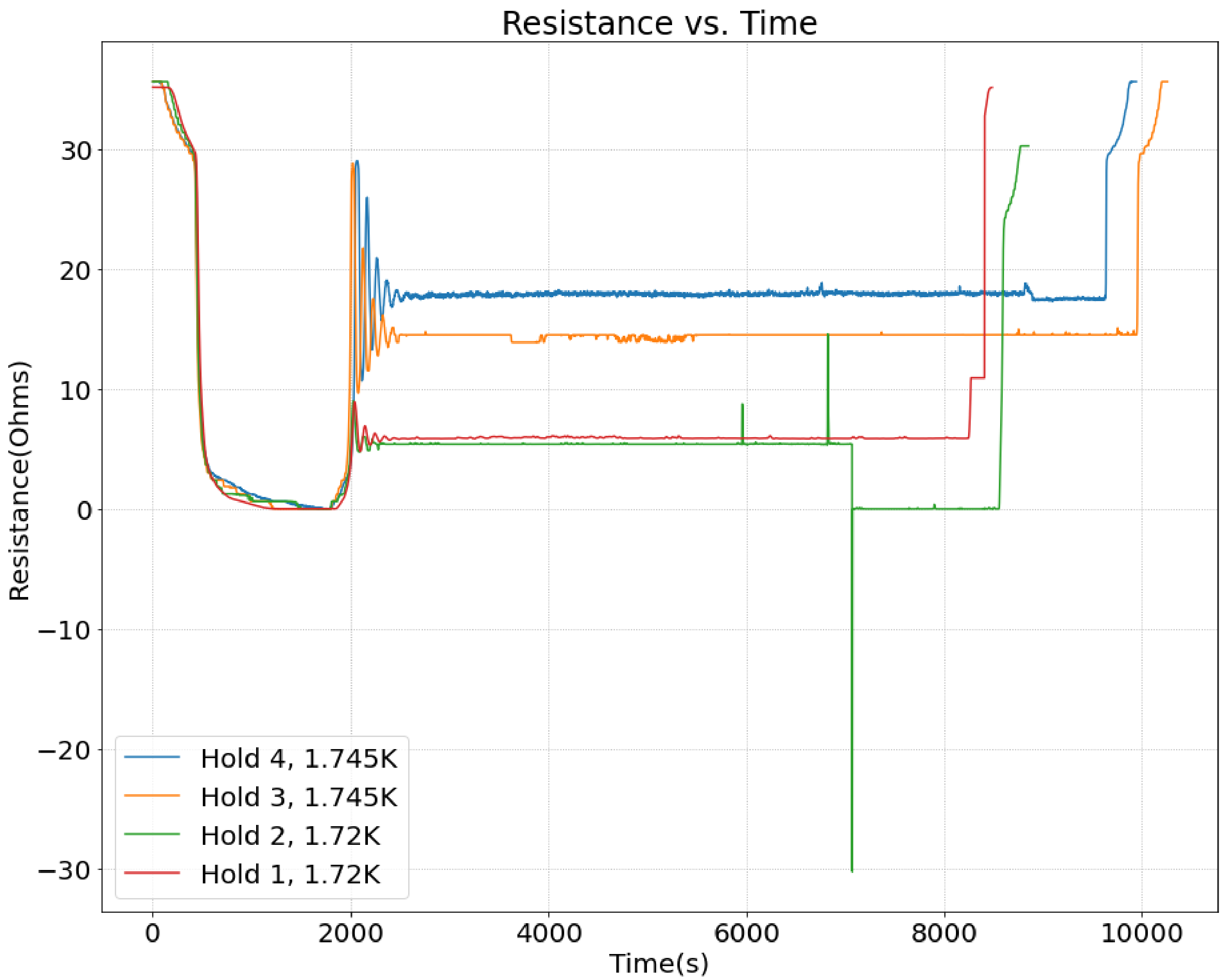
```
res12 = []
tme12 = []
temp12 = []
ftemp12 = []
k = 0
for i in range(len(tme2)):
    if temp2[i] < 2:
        tme12.append(k)
        res12.append(res2[i])
        temp12.append(temp2[i])
        ftemp12.append(ftemp2[i])
        k = k + 2.5
```

In [6]:

```
res13 = []
tme13 = []
temp13 = []
ftemp13 = []
k = 0
for i in range(len(tme3)):
    if temp3[i] < 2:
        tme13.append(k)
        res13.append(res3[i])
        temp13.append(temp3[i])
        ftemp13.append(ftemp3[i])
        k = k + 2.5
```

In [7]:

```
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')
plt.grid(linestyle = ':')
plt.title('Resistance vs. Time')
plt.plot(tme1,res1,label = 'Hold 4, 1.745K')
plt.plot(tme12,res12, label = 'Hold 3, 1.745K')
plt.plot(tme191,res191, label = 'Hold 2, 1.72K')
plt.plot(tme13,res13, label = 'Hold 1, 1.72K')
plt.legend()
plt.show()
```



So we can clearly see that for the same time constant we have hold 3 < hold 2 < hold 1 in terms of noise.

Now we want to see how the transition looks like for each of the cooldowns

In [8]:

```
res1 = []
tme1 = []
temp1 = []
ftemp1 = []
for i in range(len(tme)):
    if tme[i] > 4000 and tme[i] < 5000:
        tme1.append(tme[i])
        res1.append(res[i])
        temp1.append(temp[i])
        ftemp1.append(ftemp[i])
```

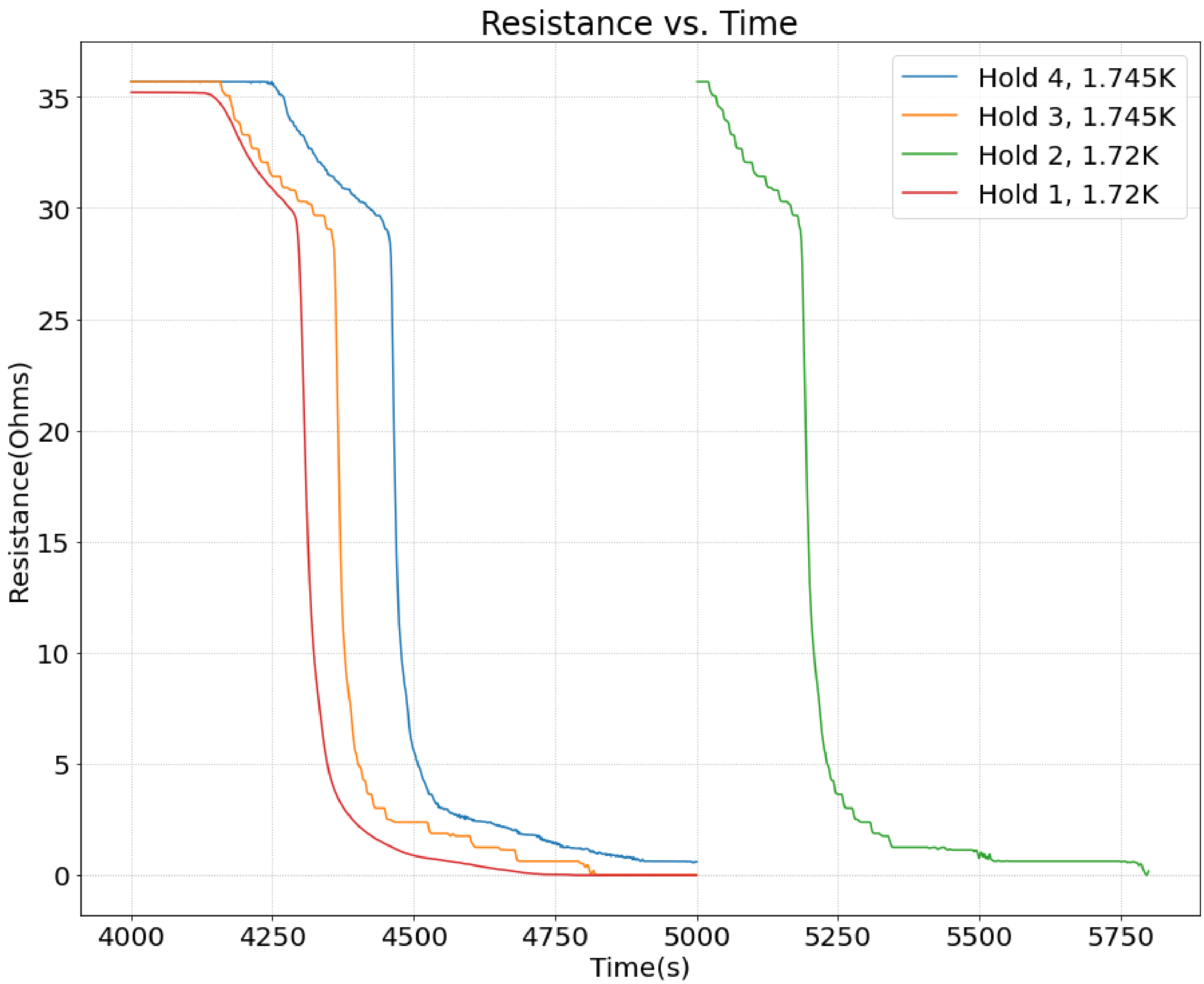
In [9]:

```
res191 = []
tme191 = []
temp191 = []
ftemp191 = []
for i in range(len(tme19)):
    if tme19[i] > 5000 and tme19[i] < 5800:
        tme191.append(tme19[i])
        res191.append(res19[i])
        temp191.append(temp19[i])
        ftemp191.append(ftemp19[i])
```

In [10]:
```
res12 = []
tme12 = []
temp12 = []
ftemp12 = []
for i in range(len(tme2)):
    if tme2[i] > 4000 and tme2[i] < 5000:
        tme12.append(tme2[i])
        res12.append(res2[i])
        temp12.append(temp2[i])
        ftemp12.append(ftemp2[i])
```

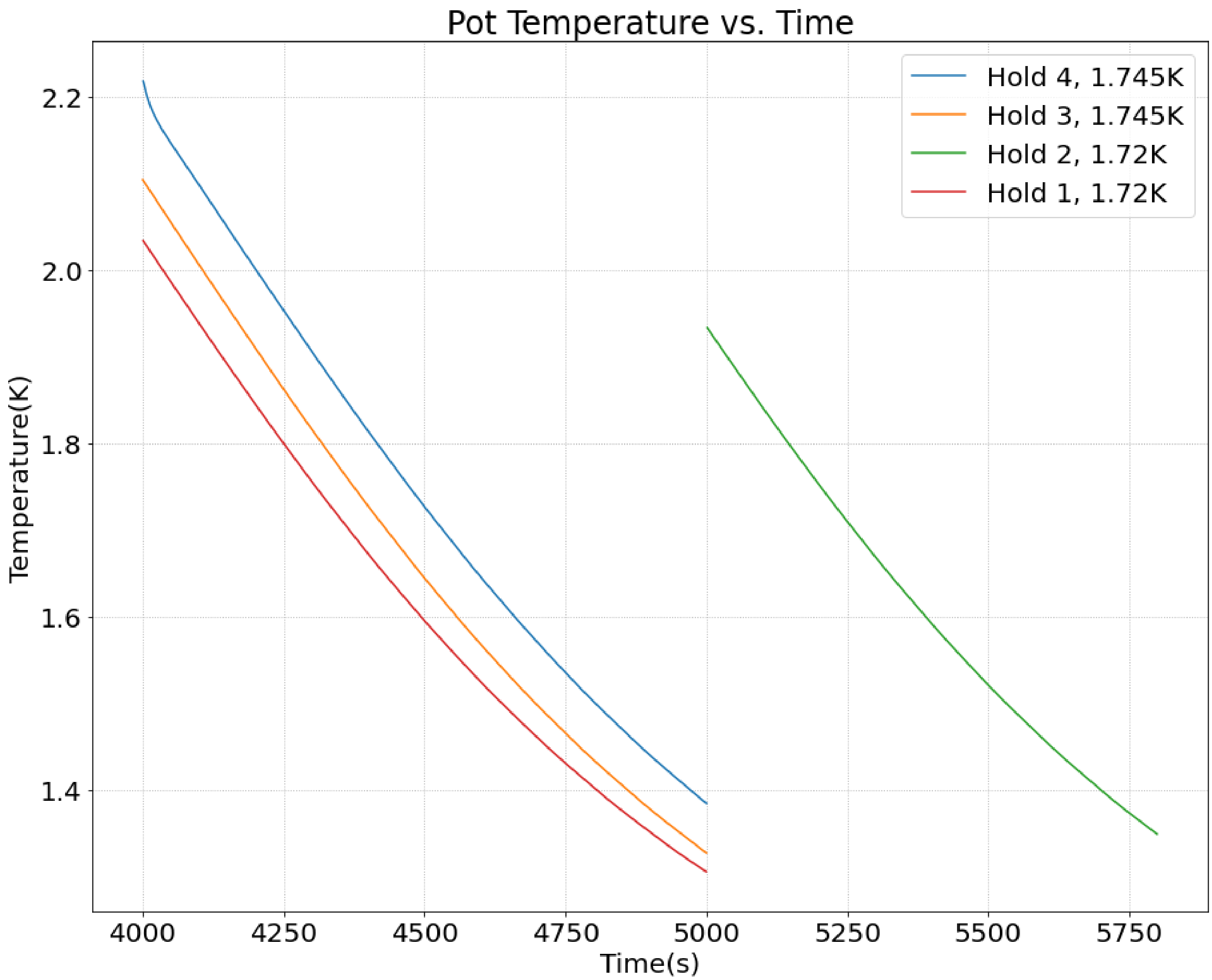
In [11]:
```
res13 = []
tme13 = []
temp13 = []
ftemp13 = []
for i in range(len(tme3)):
    if tme3[i] > 4000 and tme3[i] < 5000:
        tme13.append(tme3[i])
        res13.append(res3[i])
        temp13.append(temp3[i])
        ftemp13.append(ftemp3[i])
```

In [12]:
```
plt.ylabel('Resistance(Ohms)')
plt.xlabel('Time(s)')
plt.grid(linestyle = ':')
plt.title('Resistance vs. Time')
plt.plot(tme1,res1,label = 'Hold 4, 1.745K')
plt.plot(tme12,res12, label = 'Hold 3, 1.745K')
plt.plot(tme191,res191, label = 'Hold 2, 1.72K')
plt.plot(tme13,res13, label = 'Hold 1, 1.72K')
plt.legend()
plt.show()
```



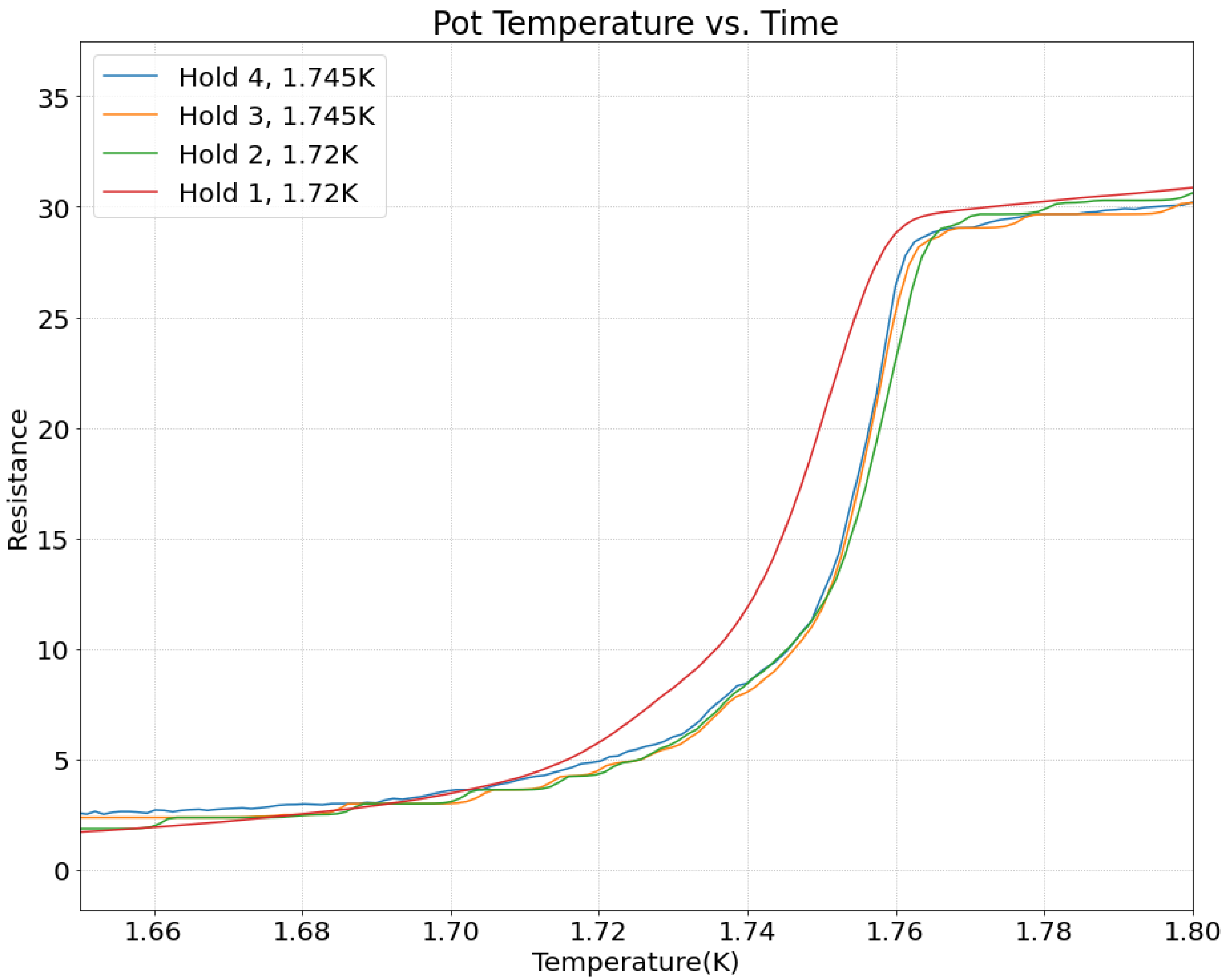
Below is the pote temperature evolution with time for the transitions.

In [13]:
```
plt.ylabel('Temperature(K)')
plt.xlabel('Time(s)')
plt.grid(linestyle = ':')
plt.title('Pot Temperature vs. Time')
plt.plot(tme1,temp1,label = 'Hold 4, 1.745K')
plt.plot(tme12,temp12, label = 'Hold 3, 1.745K')
plt.plot(tme191,temp191, label = 'Hold 2, 1.72K')
plt.plot(tme13,temp13, label = 'Hold 1, 1.72K')
plt.legend()
plt.show()
```



In [14]:

```
plt.grid(linestyle = ':')
plt.xlabel('Temperature(K)')
plt.ylabel('Resistance')

plt.title('Resistance vs. pot temperature')

plt.title('Pot Temperature vs. Time')
plt.plot(temp1,res1,label = 'Hold 4, 1.745K')
plt.plot(temp12,res12, label = 'Hold 3, 1.745K')
plt.plot(temp191,res191, label = 'Hold 2, 1.72K')
plt.plot(temp13,res13, label = 'Hold 1, 1.72K')
plt.xlim(1.65,1.8)
plt.legend()
plt.show()
```



There are some steps in the Tc measurements which shows up mainly in the transitions measured with the basement SR830

Next I am going to fit the transition edge to a line and extrapolate the temperature fluctuation of sample from the resistance fluctuations of the sample from the Tc fluctuations.

In [15]:

```
res1 = []
tme1 = []
temp1 = []
ftemp1 = []
for i in range(len(tme)):
    if tme[i] > 4463 and tme[i] < 4469:
        tme1.append(tme[i])
        res1.append(res[i])
        temp1.append(temp[i])
        ftemp1.append(ftemp[i])
```

In [16]:

```
res191 = []
tme191 = []
temp191 = []
ftemp191 = []
for i in range(len(tme19)):
    if tme19[i] > 5235 and tme19[i] < 5245:
        tme191.append(tme19[i])
        res191.append(res19[i])
        temp191.append(temp19[i])
        ftemp191.append(ftemp19[i])
```

In [17]:

```
res12 = []
tme12 = []
temp12 = []
ftemp12 = []
for i in range(len(tme2)):
    if tme2[i] > 4375 and tme2[i] < 4385:
        tme12.append(tme2[i])
        res12.append(res2[i])
        temp12.append(temp2[i])
        ftemp12.append(ftemp2[i])
```

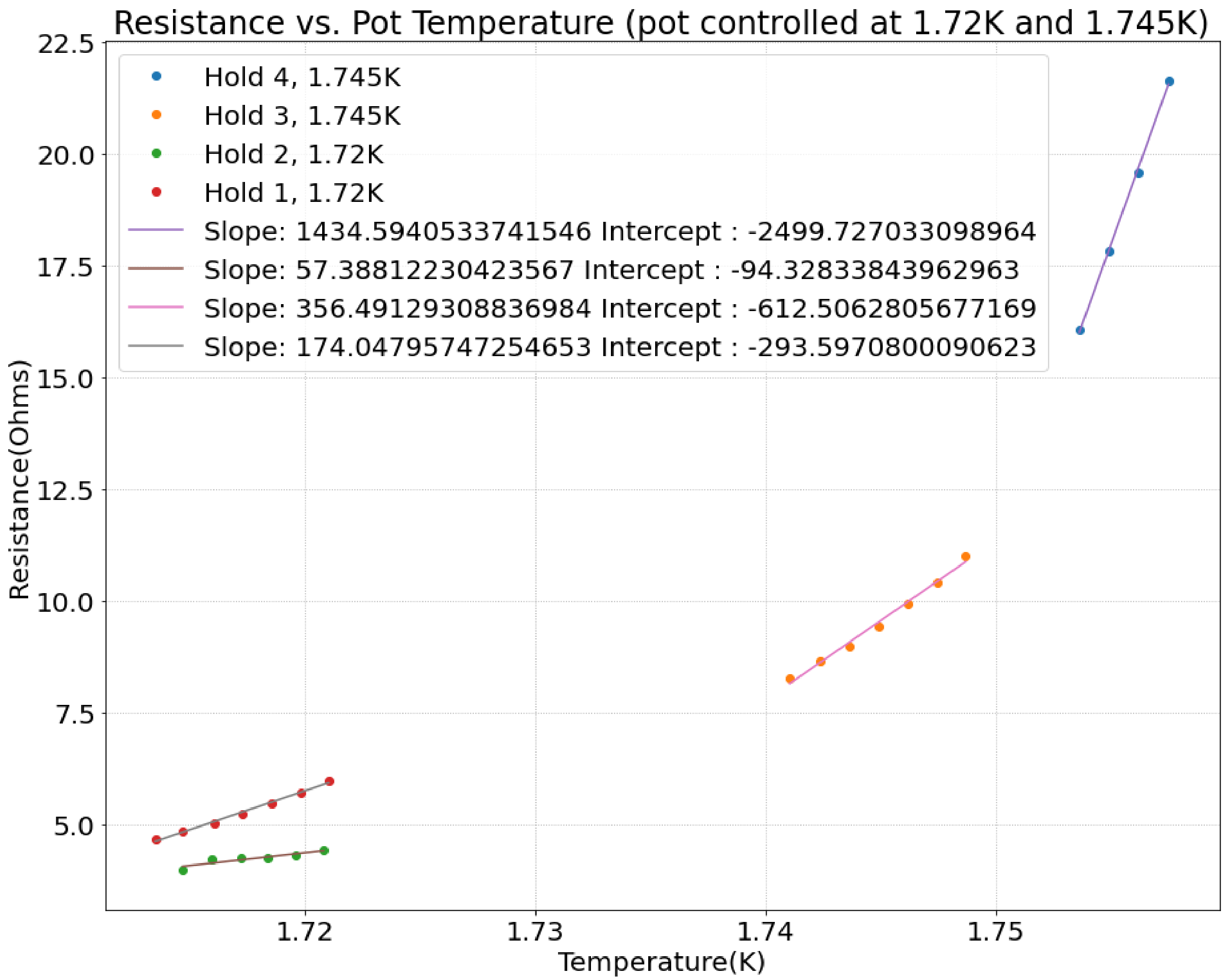
In [18]:

```
res13 = []
tme13 = []
temp13 = []
ftemp13 = []
for i in range(len(tme3)):
    if tme3[i] > 4340 and tme3[i] < 4350:
        tme13.append(tme3[i])
        res13.append(res3[i])
        temp13.append(temp3[i])
        ftemp13.append(ftemp3[i])
```

Going to fit the transition edge to a line and extrapolate the temperature from there.

In [19]:

```
plt.title('Resistance vs. Pot Temperature (pot controlled at 1.72K and 1.745K)')
plt.plot(temp1,res1,'o',label = 'Hold 4, 1.745K')
plt.plot(temp12,res12, 'o',label = 'Hold 3, 1.745K')
plt.plot(temp191,res191, 'o',label = 'Hold 2, 1.72K')
plt.plot(temp13,res13, 'o',label = 'Hold 1, 1.72K')

plt.ylabel('Resistance(Ohms)')
plt.xlabel('Temperature(K)')
plt.grid(linestyle = ':')
a, b = np.polyfit(temp1, res1, 1)
a19, b19 = np.polyfit(temp191, res191, 1)
a2, b2 = np.polyfit(temp12, res12, 1)
a3, b3 = np.polyfit(temp13, res13, 1)

tempf1 = []
tempf19 = []
tempf2 = []
tempf3 = []
for i in range(len(temp1)):
    tempf1.append(a*temp1[i] + b)
for i in range(len(temp191)):
    tempf19.append(a19*temp191[i] + b19)
for i in range(len(temp12)):
    tempf2.append(a2*temp12[i] + b2)
for i in range(len(temp13)):
    tempf3.append(a3*temp13[i] + b3)


plt.plot(temp1, tempf1,label = 'Slope: ' + str(a) + ' Intercept : ' + str(b))
plt.plot(temp191, tempf19, label = 'Slope: ' + str(a19) + ' Intercept : ' + str(b19))
plt.plot(temp12, tempf2,label = 'Slope: ' + str(a2) + ' Intercept : ' + str(b2))
plt.plot(temp13, tempf3,label = 'Slope: ' + str(a3) + ' Intercept : ' + str(b3))

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



Now that we have found the slope of the transition edge we can use that as a sensor. But first i am going to plot the pot temperature stability for all the holds.

In [20]:

```
res1 = []
tme1 = []
temp1 = []
ftemp1 = []
for i in range(len(tme)):
    if tme[i] > 6000 and tme[i] < 9250:
        tme1.append(tme[i])
        res1.append(res[i])
        temp1.append(temp[i])
        ftemp1.append(ftemp[i])
```

In [21]:

```
res191 = []
tme191 = []
temp191 = []
ftemp191 = []
for i in range(len(tme19)):
    if tme19[i] > 6500 and tme19[i] < 9250:
        tme191.append(tme19[i])
        res191.append(res19[i])
        temp191.append(temp19[i])
        ftemp191.append(ftemp19[i])
```

In [22]:

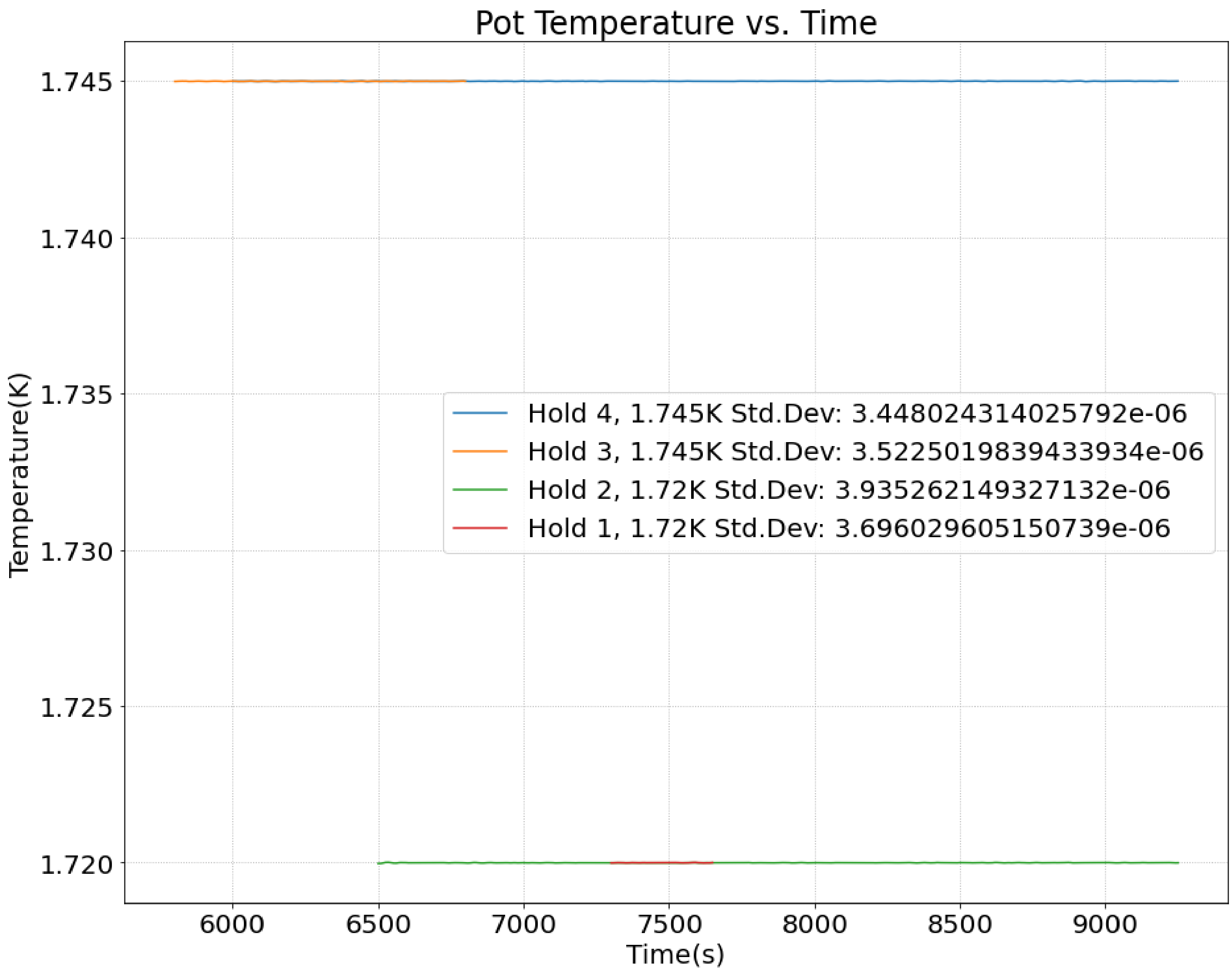
```
res12 = []
tme12 = []
temp12 = []
ftemp12 = []
for i in range(len(tme2)):
    if tme2[i] > 5800 and tme2[i] < 6800:
        tme12.append(tme2[i])
        res12.append(res2[i])
        temp12.append(temp2[i])
        ftemp12.append(ftemp2[i])
```

In [28]:

```
res13 = []
tme13 = []
temp13 = []
ftemp13 = []
for i in range(len(tme3)):
    if tme3[i] > 7300 and tme3[i] < 7650:
        tme13.append(tme3[i])
        res13.append(res3[i])
        temp13.append(temp3[i])
        ftemp13.append(ftemp3[i])
```

In [24]:

```
plt.ylabel('Temperature(K)')
plt.xlabel('Time(s)')
plt.grid(linestyle = ':')
plt.title('Pot Temperature vs. Time')
plt.plot(tme1,temp1,label = 'Hold 4, 1.745K Std.Dev: ' + str(np.std(temp1)))
plt.plot(tme12,temp12, label = 'Hold 3, 1.745K Std.Dev: ' + str(np.std(temp12)))
plt.plot(tme191,temp191, label = 'Hold 2, 1.72K Std.Dev: ' + str(np.std(temp191)))
plt.plot(tme13,temp13, label = 'Hold 1, 1.72K Std.Dev: ' + str(np.std(temp13)))
plt.legend()
plt.show()
```



So we have temerature stability is the range of 3-4 uK.

Next i am going to seek out the stable zones of the resistance fluctuations, usually over all the region stability varries a lot but usually around 100 - 300uK.

In [25]:

```
res1 = []
tme1 = []
temp1 = []
ftemp1 = []
for i in range(len(tme)):
    if tme[i] > 6000 and tme[i] < 8000:
        tme1.append(tme[i])
        res1.append(res[i])
        temp1.append(temp[i])
        ftemp1.append(ftemp[i])
```

In [26]:

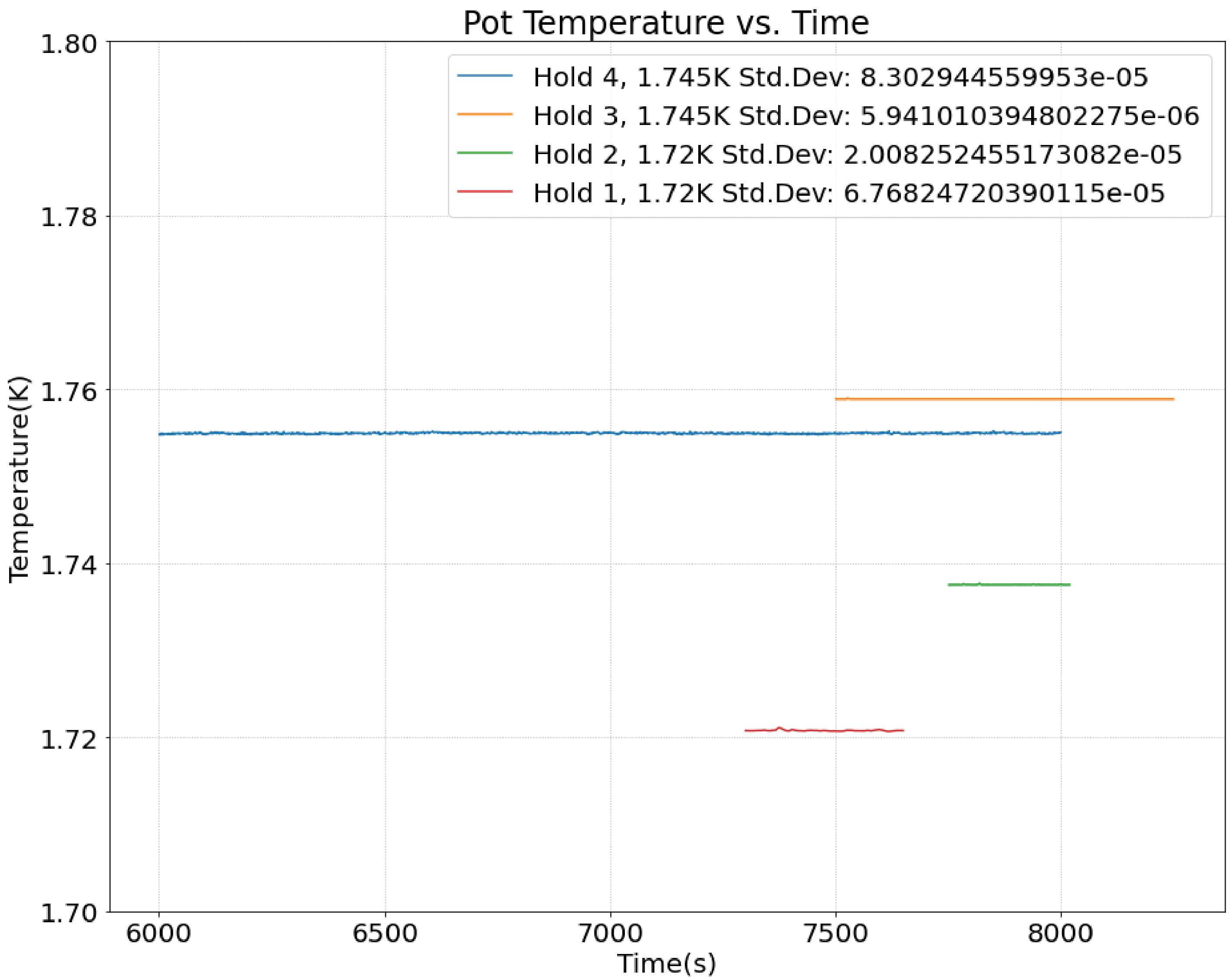
```
res191 = []
tme191 = []
temp191 = []
ftemp191 = []
for i in range(len(tme19)):
    if tme19[i] > 7750 and tme19[i] < 8020:
        tme191.append(tme19[i])
        res191.append(res19[i])
        temp191.append(temp19[i])
        ftemp191.append(ftemp19[i])
```

In [27]:
```
res12 = []
tme12 = []
temp12 = []
ftemp12 = []
for i in range(len(tme2)):
    if tme2[i] > 7500 and tme2[i] < 8250:
        tme12.append(tme2[i])
        res12.append(res2[i])
        temp12.append(temp2[i])
        ftemp12.append(ftemp2[i])
```

In [28]:
```
res13 = []
tme13 = []
temp13 = []
ftemp13 = []
for i in range(len(tme3)):
    if tme3[i] > 7300 and tme3[i] < 7650:
        tme13.append(tme3[i])
        res13.append(res3[i])
        temp13.append(temp3[i])
        ftemp13.append(ftemp3[i])
```

In [29]:
```
tempf1 = []
tempf19 = []
tempf2 = []
tempf3 = []
for i in range(len(temp1)):
    tempf1.append((res1[i]-b)/a)
for i in range(len(temp191)):
    tempf19.append((res191[i]-b19)/a19)
for i in range(len(temp12)):
    tempf2.append((res12[i]-b2)/a2)
for i in range(len(temp13)):
    tempf3.append((res13[i]-b3)/a3)
```

In [30]:
```
plt.ylabel('Temperature(K)')
plt.xlabel('Time(s)')
plt.grid(linestyle = ':')
plt.title('Pot Temperature vs. Time')
plt.plot(tme1,tempf1,label = 'Hold 4, 1.745K Std.Dev: ' + str(np.std(tempf1)))
plt.plot(tme12,tempf2, label = 'Hold 3, 1.745K Std.Dev: ' + str(np.std(tempf2)))
plt.plot(tme191,tempf19, label = 'Hold 2, 1.72K Std.Dev: ' + str(np.std(tempf19)))
plt.plot(tme13,tempf3, label = 'Hold 1, 1.72K Std.Dev: ' + str(np.std(tempf3)))
plt.ylim(1.7,1.8)
plt.legend()
plt.show()
```



So with more or less good SRS830 and selecting proper TC we can see around 6uK of sample temperature.
i will do another Lp hold with TC of 1s again to recreate similar stability of hold 3.

In [ ]: