A3_Question2

August 29, 2021

0.1 Question2

[ ]:
```
#Assumptions:
#Let initial inner temperature be 77K
#Temperature curve without intervention: Linear increase by 10K every sec
#Let duration between refill be randomized (as the operator replenishes at␣
 ↪random intervals)
#Total duration for plotting and simulation be 100 units
#Assume, After refill let temperature drop to 77K to 9/10th of current␣
 ↪temperature every second (As realistically temperature
#won't instantly drop after refill)
#Reason for assumption:
#Without refill-> Becomes unfit after 13 seconds
#With say 8 refills of random intervals (equal probability of choosing an␣
 ↪interval between say 5-10).
#Choosing 10 time units as intervals in between all refills is worst case
#By this the worst case timing we get at which it becomes unfit is close to 98␣
 ↪units of time
#Hence we can choose 100 units as a total duration to simulate the whole␣
 ↪phenomena
```

0.1.1 Required Packages:

[24]:
```
import random as rnd
import matplotlib.pyplot as plt
```

[156]:
```
#Random intervals at which operator refills
intervals=[0]
for i in range(8):
    if(intervals[-1]+rnd.randint(5,10)>100):
        break
    intervals.append(intervals[-1]+rnd.randint(5,10))
```

[157]:
```
intervals=intervals[1:]
```

[158]:
```
intervals
```

[158]: [9, 19, 28, 34, 40, 48, 57, 63]

[159]:
```
#Flagging:
def check(cur_T,threshold):
    return cur_T>threshold
```

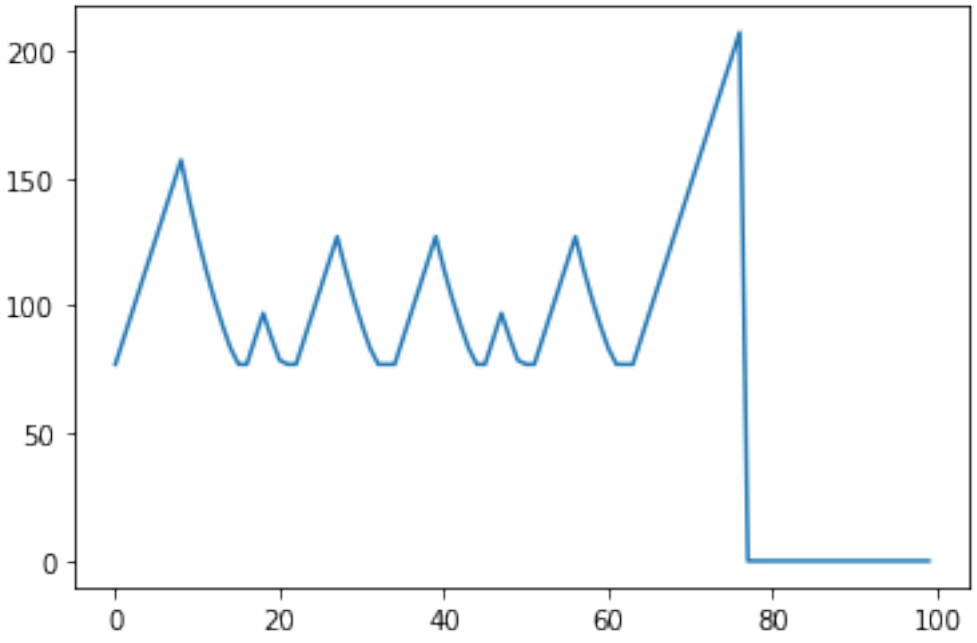
[160]:
```
#Thermocouple data
l=[i for i in range(0,101)]
t=[77]
f,c=0,1
for i in range(1,len(l)-1):
    if(check(t[-1],200) or f==1):
        nex=0
        if(f==0):print("Unfit for further use after:",i,"units of time")
        f=1
    elif(f==0 and i in intervals) or c==0:
        c=0
        nex=t[-1]*9/10
        if(nex<77):nex=77
        if(t[-1]==77):
            c=1
    elif(f==0 and c==1):
        nex=t[-1]+10
    t.append(nex)
l=l[:-1]
```

```
Unfit for further use after: 77 units of time
```

[161]:
```
print(l)
print(t)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
[77, 87, 97, 107, 117, 127, 137, 147, 157, 141.3, 127.17, 114.453, 103.0077,
92.70693, 83.436237, 77, 77, 87, 97, 87.3, 78.57, 77, 77, 87, 97, 107, 117, 127,
114.3, 102.87, 92.583, 83.32469999999999, 77, 77, 77, 87, 97, 107, 117, 127,
114.3, 102.87, 92.583, 83.32469999999999, 77, 77, 87, 97, 87.3, 78.57, 77, 77,
87, 97, 107, 117, 127, 114.3, 102.87, 92.583, 83.32469999999999, 77, 77, 77, 87,
97, 107, 117, 127, 137, 147, 157, 167, 177, 187, 197, 207, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

[162]:
```
plt.plot(l,t)
plt.show()
```


200
150
100
50
0
0
20
40
60
80
100