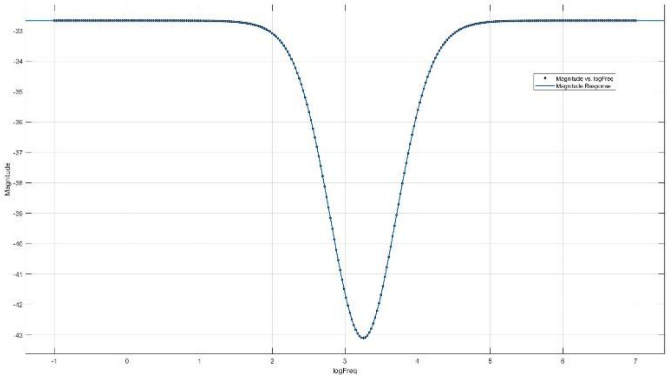
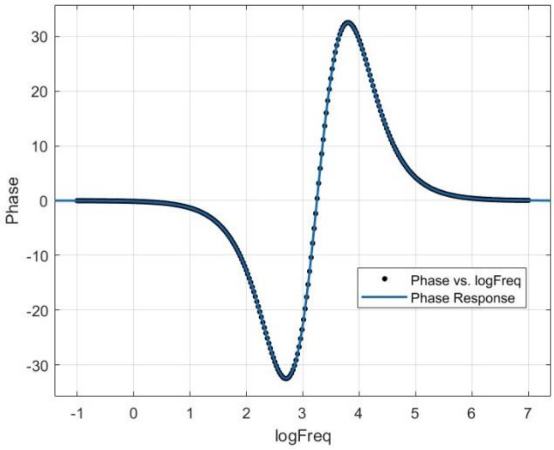
LIGO Questionnaire Answers

Question 1:

The given Magnitude and Phase is plotted against the log of Frequency. The following plots are obtained,





We can observe that this a notch filter (band stop filter) and we require a 10db/decade attenuation and amplification for this response. A notch filter is made using the combination of a low pass and a high pass filter. But since transfer functions of a simple RLC low/high pass filter gives attenuation/amplification of multiples of ~20db/decade, achieving a slope of 10db/decade would be a challenge. We utilise a Twin-T notch filter for addressing this. Being able to use an amplifying unit (opamp) would give us better control over our transfer function but due to the constrains of the question(R,L,C), we limit ourselves to a passive filter.

On analysis the magnitude response we get the following characteristics,

Frequency at which attenuation from baseline is 3db on lower frequency side (f_{3db_L}) = 319.04 Hz

Frequency at which attenuation from baseline is 3db on higher frequency side (f_{3db_R}) = 9931.28 Hz

Bandwidth = f_{3db_H} - f_{3db_L} = 9612.24 Hz

$F_{Notch} = \sqrt{f_{3db\_H} * f_{3db\_L}}$ = 1780.02 Hz

Baseline magnitude response = -32.6694 db

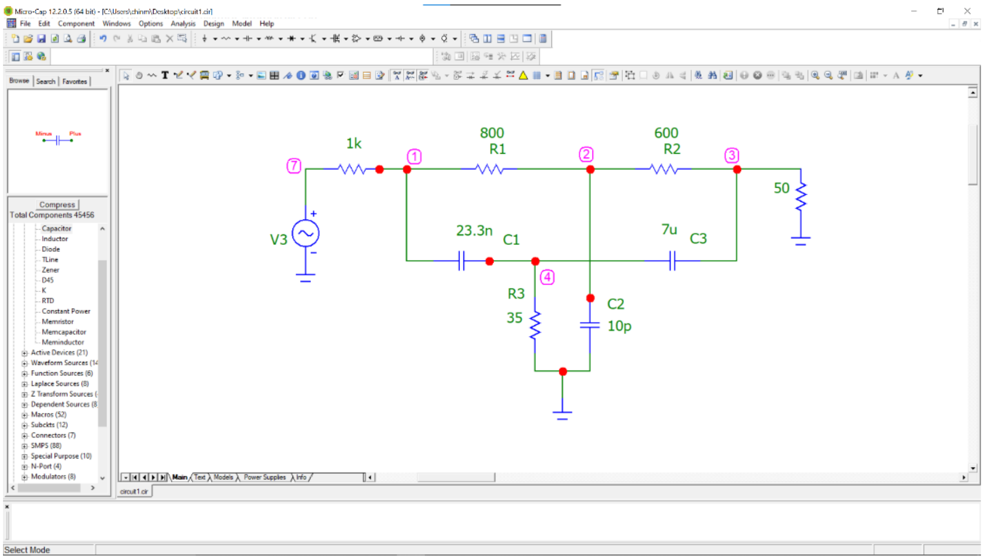
Minimum magnitude response = -43.1063 db

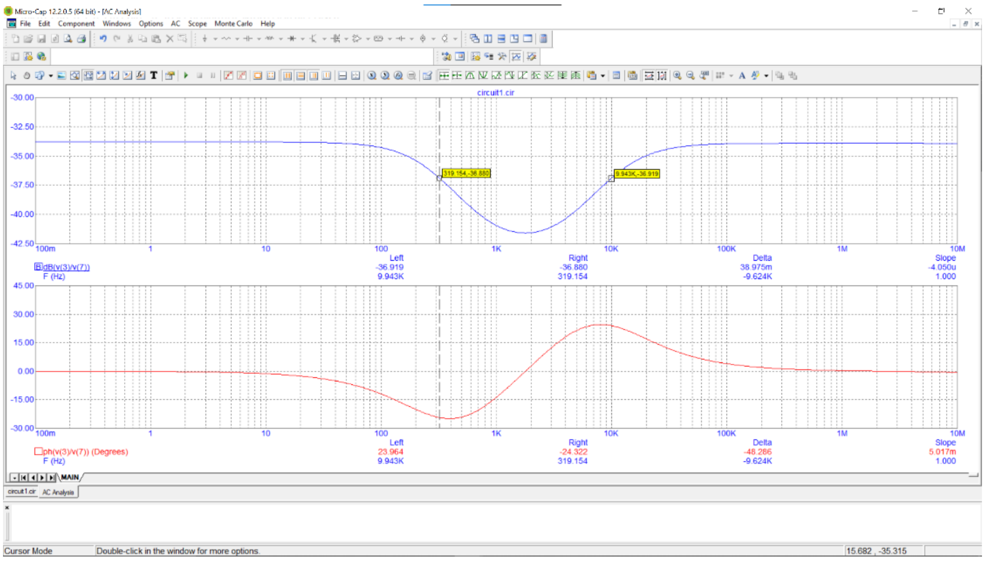
Also, upon analysis of this Frequency Response in MATLAB’s System Identification toolbox and using the tfest() (transfer function estimation) function we curve fit the frequency response data to obtain a transfer function of,

$$\frac{-32 * s^4 + 1.3e5 * s^3 + 1e9 * s^2 + 4.115e11 * s - 3.276e14}{s^4 + 3056 * s^3 - 2.327e7 * s^2 + 9.67e9 * s + 1e13}$$

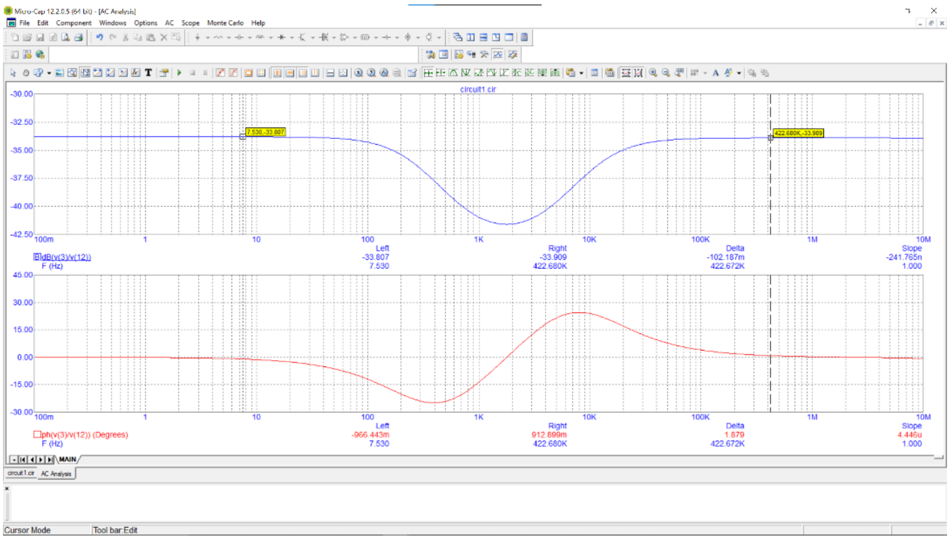
with a MSE of 0.04047 and 97.76% Fit.

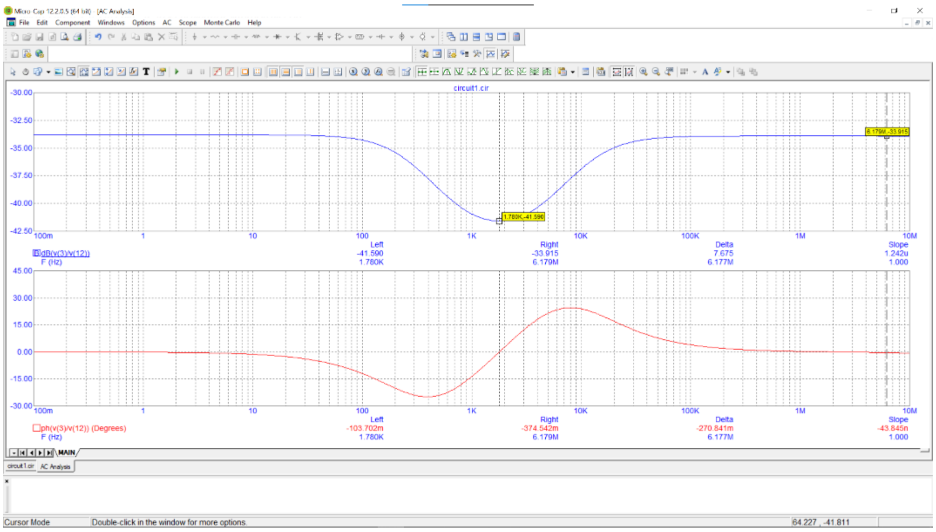
1) The following circuit design gave practically identical results to the given black box frequency response:





```
                          Left      Right     Delta     Slope
dB(v(3)/v(7))            -36.919   -36.880   38.975m   -4.050u
F (Hz)                    9.943K   319.154   -9.624K    1.000

                          Left      Right     Delta     Slope
ph(v(3)/v(7)) (Degrees)  23.964    -24.322   -48.286    5.017m
F (Hz)                    9.943K   319.154   -9.624K    1.000

Cursor Mode    Double-click in the window for more options.    15.682 , -35.315
```





f_{3db_L} = 319 Hz

f_{3db_R} = 9943 Hz

Bandwidth = 9622 Hz

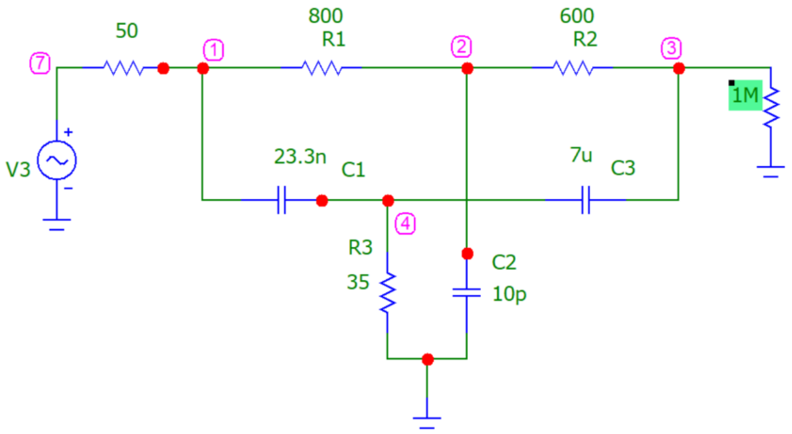
f_{Notch} = 1780 Hz

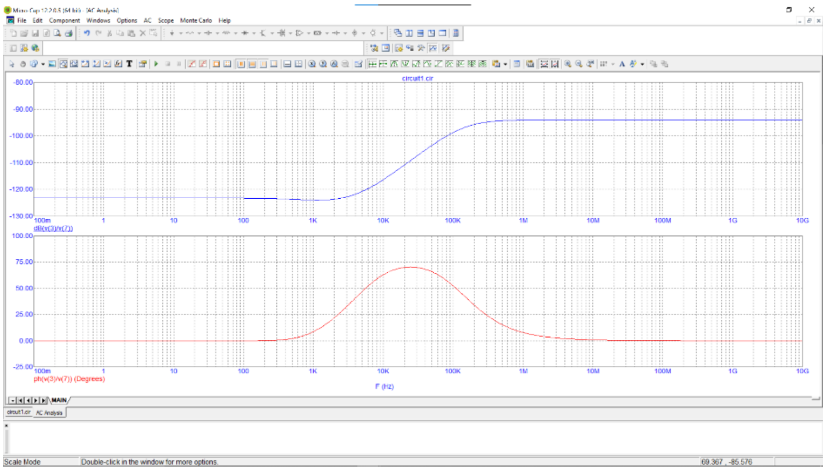
Baseline magnitude response (at lower frequency) = -33.8 db

Baseline magnitude response (at higher frequency) = -33.9 db

Minimum magnitude response = -41.590 db

2) Upon using the new conditions on the designed black box, we get,

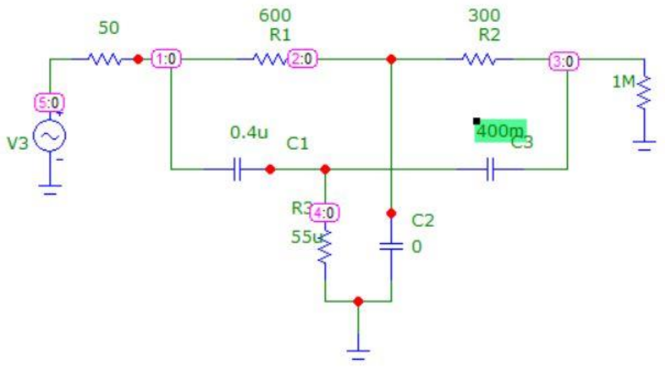


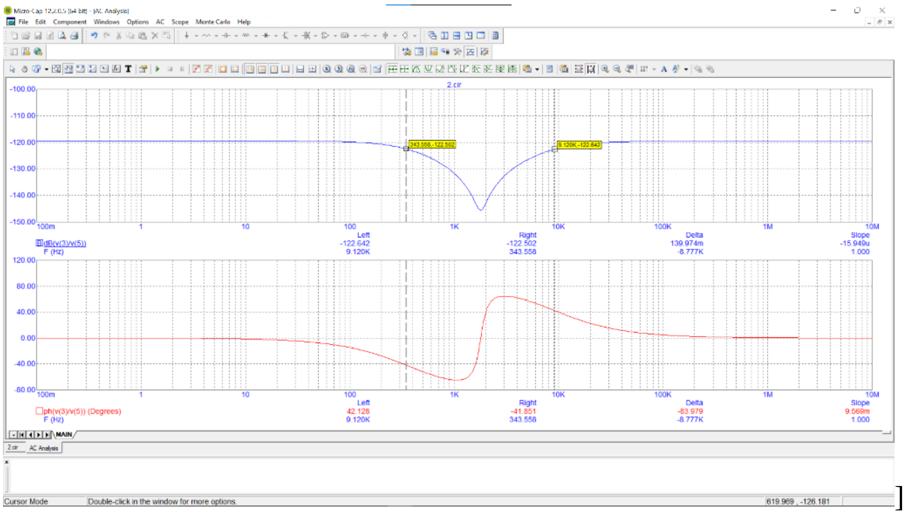


Double-click in the window for more options.

As we can observe from the frequency response, these conditions result in the circuit acting as a high pass filter.

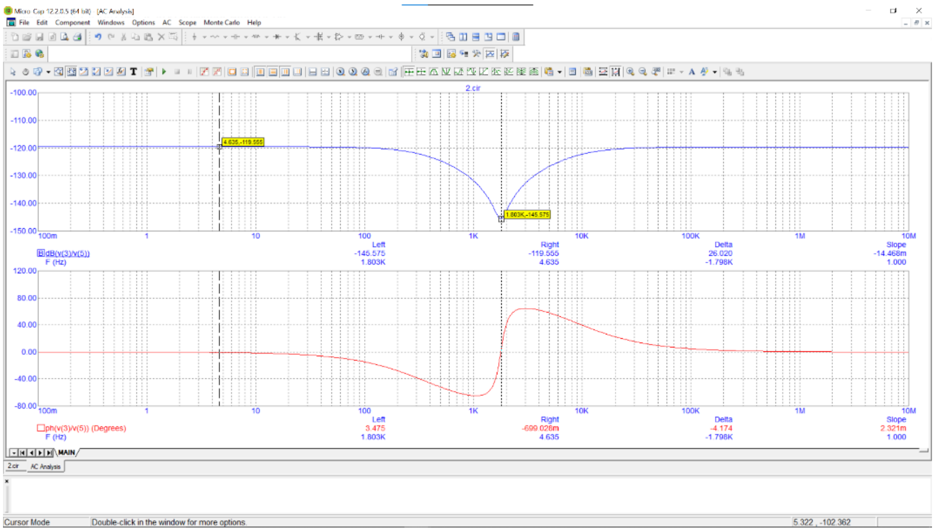
3) Upon closer inspection, we observe a sight dip in the end of maximum attenuating region. We can infer from this that the conditions caused the low pass filter to “flatten” out, resulting in just a high pass filter. The sight dip is a residual from the low pass filter due to imperfect “flattening”. This implies that we need to bring back the low pass filter in order to get back the original frequency response. Upon tweaking the RC values, we get the following response,





```
                    Left      Right      Delta      Slope
dB(v(3)/v(5))    -122.642   -122.502   139.974m   -15.949u
F (Hz)             9.120K    343.558    -8.777K      1.000

                    Left      Right      Delta      Slope
ph(v(3)/v(5)) (Degrees)
                   42.128    -41.851    -83.979     9.569m
F (Hz)             9.120K    343.558    -8.777K      1.000
```

Cursor Mode   Double-click in the window for more options.   619.969 , -126.181



```
                    Left      Right      Delta      Slope
dB(v(3)/v(5))    -145.575   -119.555    26.020    -14.468m
F (Hz)             1.803K      4.635    -1.798K      1.000

                    Left      Right      Delta      Slope
ph(v(3)/v(5)) (Degrees)
                    3.475   -699.026m    -4.174     2.321m
F (Hz)             1.803K      4.635    -1.798K      1.000
```

Cursor Mode   Double-click in the window for more options.   5.322 , -102.362

f_{3db_L} = 343.5 Hz

f_{3db_R} = 9120 Hz

Bandwidth = 8776.5 Hz

f_{Notch} = 1803 Hz

Baseline magnitude response= -119.5 db

Minimum magnitude response = -145.6 db

During all the analysis, a higher significance was given to getting the 3db frequency to be as close as possible over the magnitude response value so as to filter the same set of frequencies.

4) A notch circuit can be used in experimental situation for signal processing in the following ways:

i. Removing noise from the signal using a notch filter with notch frequency centred at the mean noise frequency and bandwidth equal to noise bandwidth.
ii. Removing unnecessary resonant frequencies from the signal.

5) Quality factor is measure of how good a filter is. Higher the quality factor is, lower the losses are and hence better the filter.

Quality Factor for notch filter = $\frac{Bandwidth}{F_{Notch}}$

Quality factor of original circuit = 9622/1780 = 21.62

Quality factor of new circuit = 8776.5/1803 = 4.87

To improve the quality factor, we can increase the bandgap keeping the notch frequency the same. In our circuit, C1 affects the higher 3db frequency whereas C3 affects the lower 3db frequency. Since the formula for f_{notch} is,

$$f_{Notch} = \sqrt{f_{\mathrm{3db\_H}} * f_{\mathrm{3db\_L}}}$$

If we increase $f_{\mathrm{3db\_H}}$ and decrease $f_{\mathrm{3db\_L}}$ by the same factor, f_{Notch} doesn't change but the bandwidth increase and hence the quality factor improves.

Question 2:

General Characterisation of the System and part i)

The variables of the system are:

Input (U): (T_{env}) Temperature of the environment

State Variable (X): (T_{cop}) Temperature of Copper block

Output (Y): (T_{cop}) Temperature of Copper block (Same as state variable)

The state space representation of a linear time-invariant (LTI) system can be modelled by,

$$\dot{X} = A * X + B * U$$

$$Y = C * X + D * U$$

This being a single input-single output system, all the matrices A, B, C, D are all scalars. Since our output is the same as the state variable, i.e. Y = X, comparing with the sate space equations we get,

C = 1 and D = 0

Our state space equations boil down to,

$$\dot{X} = A * X + B * U$$

Y = X

The block diagram for this is given by:

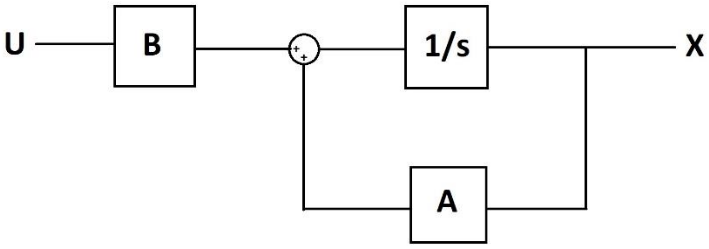


Fig1. Block Diagram for the Cu system

Due to the kryptonite coating, the block is perfectly reflective and non-conductive at the surface. Therefore all energy is conducted to and from the cube via the conducting strip alone (heat cannot escape or reach the cube via its surface)

The dimensions of the copper strap are 1m x 10^-2 m^2. Since the cross-sectional are of the strap is relatively smaller, we can make the assumption that the temperature of copper strap across its cross-section is constant. This implies that the heat flow across the cross-section of the strap is 0 and hence a temperature gradient can only exist along the length of the copper strap.
In the problem, the temperature of the block has also been taken to be constant.

Our system will consist of copper in a temperature range ϵ [300K, 400K]. The density of copper at room temperature (300K) is 8960 kg/m^3 and at 400K is 8910 kg/m^3. Since these differ by (8960-8910)/8960 = 50/8960 = 0.558 %, we'll assume that density is at a constant of (8960+8910)/2 = 8935 kg/m^3.

Specific Heat Capacity = (380 + 390)/2 = 385 J/kg K. [1] [2]

Considering the aforementioned model of heat transfer, the following equations are used –

1. Rate of Heat Transfer through strip:

$$H = \frac{T_{env} - T_{Cu}}{\frac{1}{\sigma}\frac{l}{A}}$$

$$H = \frac{T_{env} - T_{Cu}}{\frac{1}{\sigma}\frac{l}{A}}$$

Where: H = Rate of Heat Transfer (W)
T_{env} = Temperature of Environment (K)
T_{Cu} = Temperature of Cu Block (K)
σ = Conductivity of Cu (W/m K)
l, A = Dimensions of Cu Strip (m, m^2)

2. Change in Temperature of block due to incoming Heat energy

$$\Delta E = mC\Delta T_{Cu}$$

Where: ΔE = Heat Energy (incoming +ive, outgoing -ive)
m = Mass of Cu Block
C = Specific Heat Capacity
ΔT_{Cu} = Change in Temperature of Cu block

Considering m and C to be constant over time (and independent of change in Temperature)

$$\frac{dE}{dt} = mC\frac{dT_{Cu}}{dt} = \sigma\frac{A(T_{env} - T_{Cu})}{L} \quad \ldots\ldots..*$$

$$\frac{dE}{dt} = mC * \frac{dT_{Cu}}{dt} = \sigma * \frac{A(T_{env} - T_{Cu})}{l}$$

In State Space Representation –

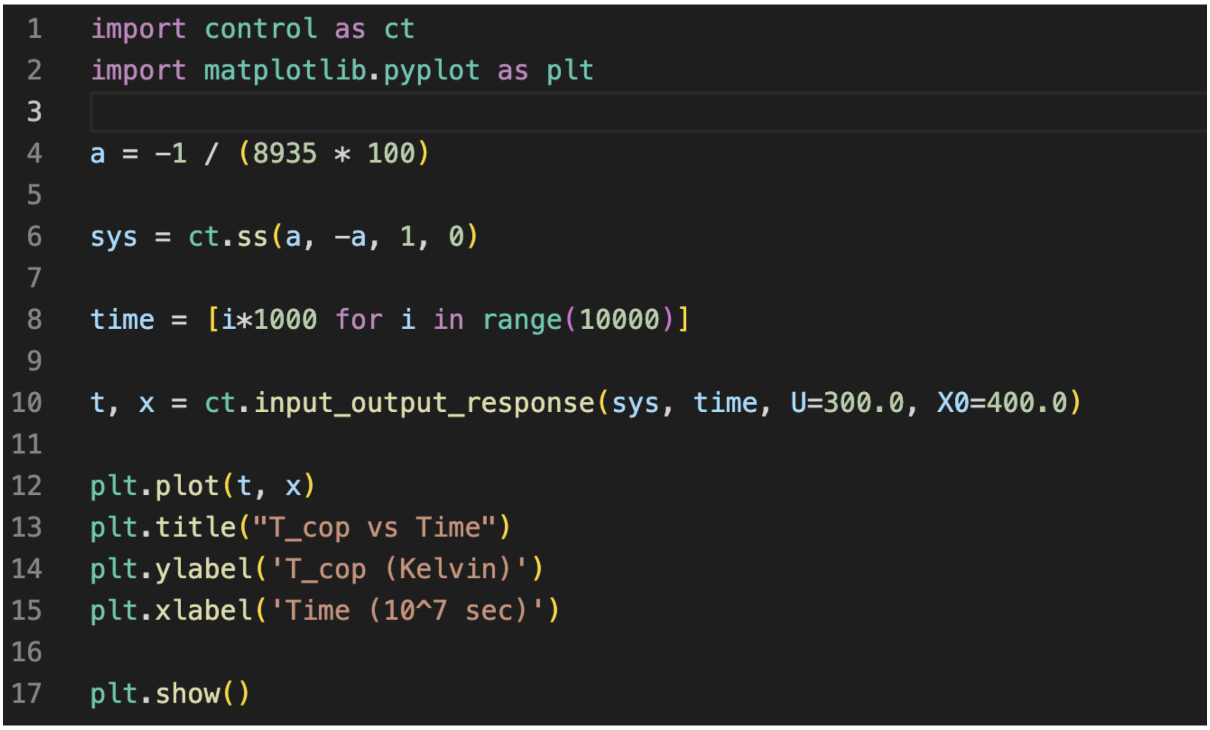
$$T_{cu}dot = -AT_{Cu} + AT_{env}$$

where – $A = \frac{\sigma A}{mCl}$ ……**

Part ii) and iii) – Control Package and Simulation

Following is a snippet of the code –

1. Representing the above equation in the State-Space using the python Control library
2. Plotting the output for the given condition: T_{env} = 300 K, T_{Cu} = 400 K



```python
import control as ct
import matplotlib.pyplot as plt

a = -1 / (8935 * 100)

sys = ct.ss(a, -a, 1, 0)

time = [i*1000 for i in range(10000)]

t, x = ct.input_output_response(sys, time, U=300.0, X0=400.0)

plt.plot(t, x)
plt.title("T_cop vs Time")
plt.ylabel('T_cop (Kelvin)')
plt.xlabel('Time (10^7 sec)')

plt.show()
```

Fig2. Code Snippet for part ii) and iii)

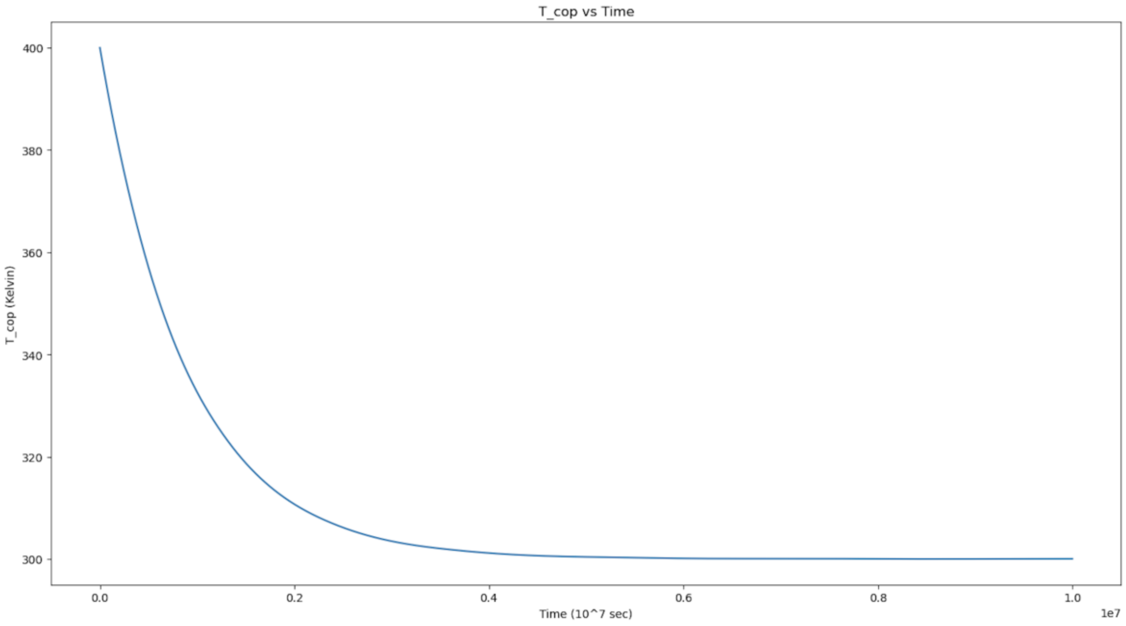


Fig3. Plot of T_{Cu} vs Time

Intuitively, we expect this since the rate of Temperature change is proportional to the difference in Temperature between the block and environment. Therefore, the smaller the temperature difference becomes, slower the rate of Temperature change
Also note, as expected from the equation (*), the plot of Temperature of the block vs Time shows exponential behaviour.

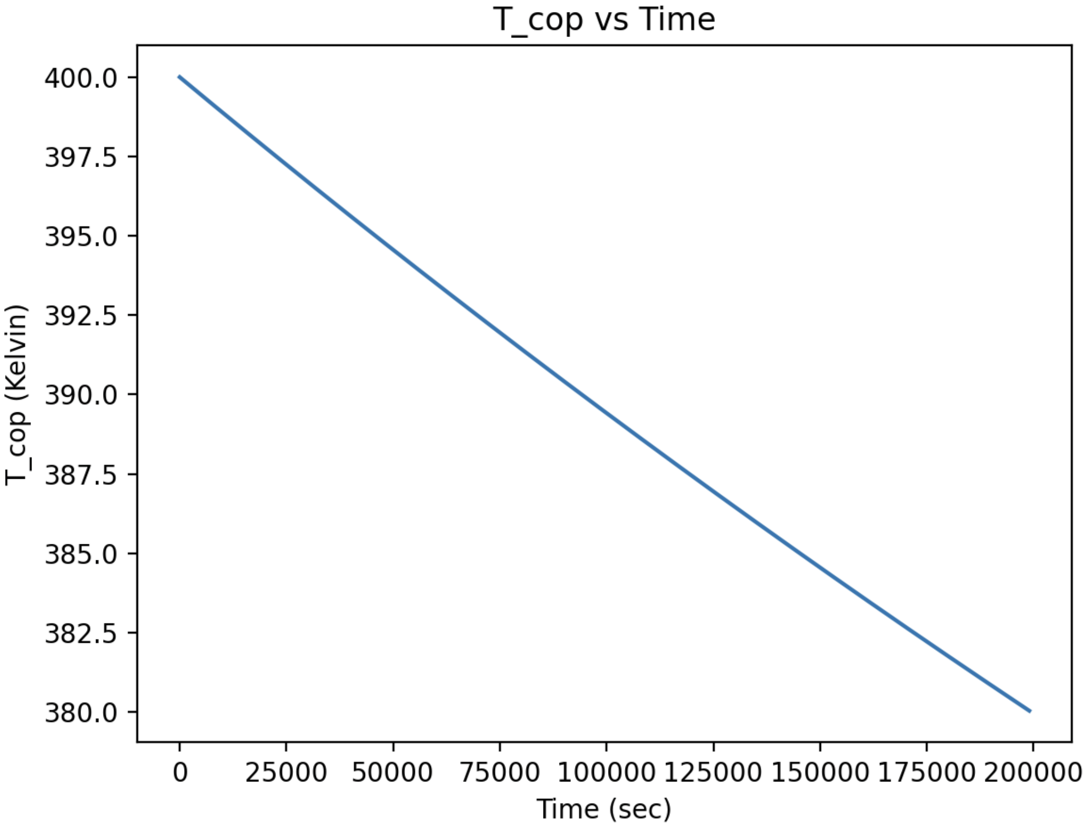


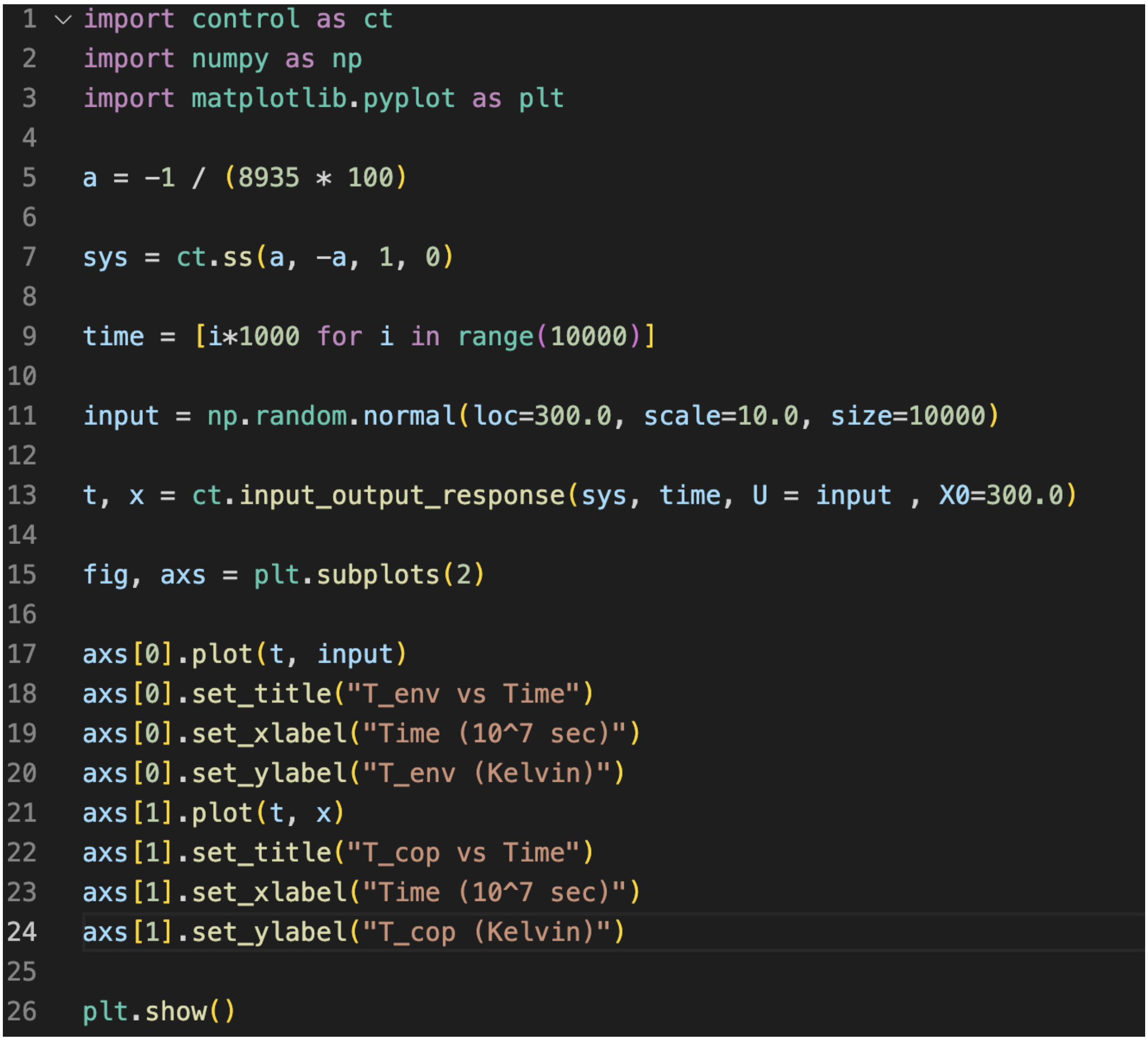
Fig4. A section of the above plot for a smaller ΔT_{Cu}

In the above plot, we can observe that for small A ΔT (A as per equation **) (or consequently a small change in Temperature) the plot appears linear.
This can also be expected by approximating the differentials to Δ ‘s in the original state equation (as expected this would be valid only for small changes in Temperature of Block i.e small time durations)

iv) Introducing Gaussian Noise

Following is a snippet of the code for the following conditions –

1. T_{env} = 300 K + Gaussian Noise
2. T_{Cu} = 300 K



```python
import control as ct
import numpy as np
import matplotlib.pyplot as plt

a = -1 / (8935 * 100)

sys = ct.ss(a, -a, 1, 0)

time = [i*1000 for i in range(10000)]

input = np.random.normal(loc=300.0, scale=10.0, size=10000)

t, x = ct.input_output_response(sys, time, U = input , X0=300.0)

fig, axs = plt.subplots(2)

axs[0].plot(t, input)
axs[0].set_title("T_env vs Time")
axs[0].set_xlabel("Time (10^7 sec)")
axs[0].set_ylabel("T_env (Kelvin)")
axs[1].plot(t, x)
axs[1].set_title("T_cop vs Time")
axs[1].set_xlabel("Time (10^7 sec)")
axs[1].set_ylabel("T_cop (Kelvin)")

plt.show()
```

Fig5. Code for part iv)

Sidenote:
The input_output_response function takes the following arguments:

1. sys = Defined system of block and strip in state space
2. T = Time array,
3. U = input (an array where each element represents the input at a given time step or a float (constant input))
4. X0 = initial condition

The function by default uses RK45 to numerically solve the given differential equation and plot the output

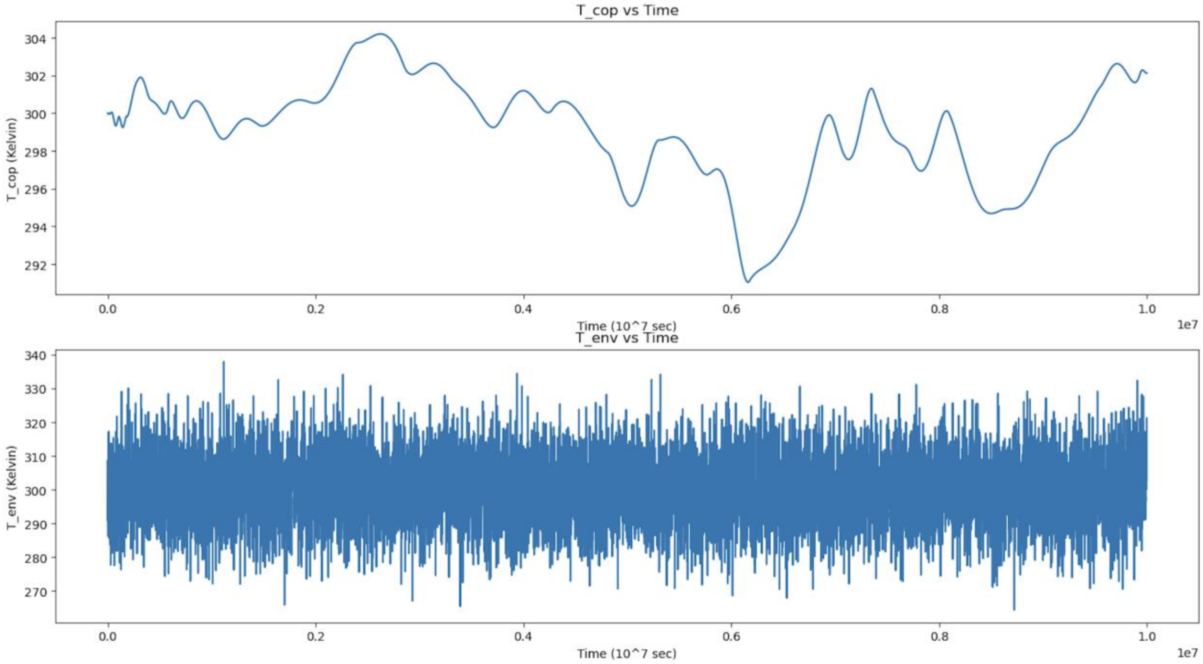
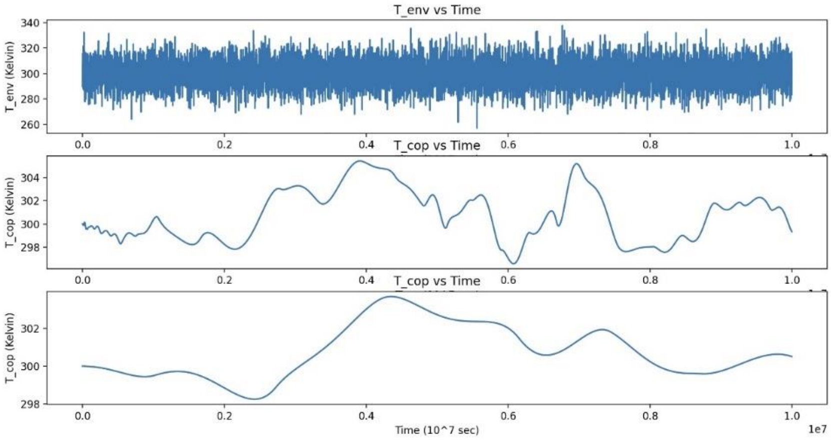


Fig6. Subplots
a) T_{Cu} vs time
b) Input Gaussian Noise vs Time

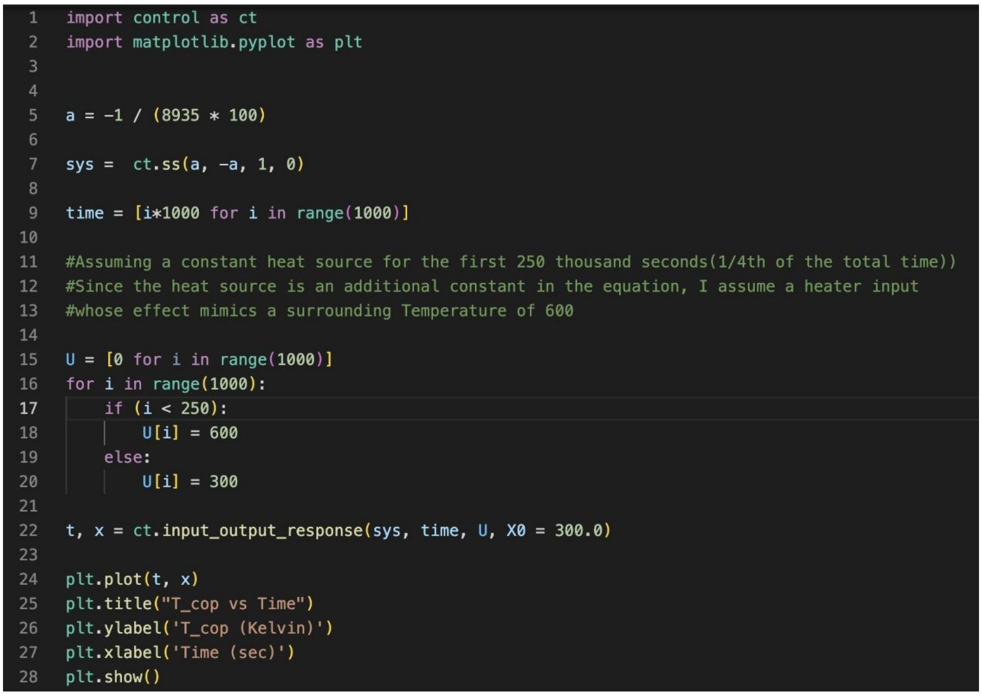
From the above plot, we see that the Copper Block shows significantly reduced response to the rapidly fluctuating Gaussian Noise. This can be intuited from the fact that the rate of change of Temp is proportional to the difference in Temp between the block and the environment (which is relatively small). Thus, for systems that are sensitive to temperature fluctuations, the above setup can be used as the Temperature source instead of the environment.

If possible, we can attempt to cascade the above setup, further reducing the fluctuations in Temperature.

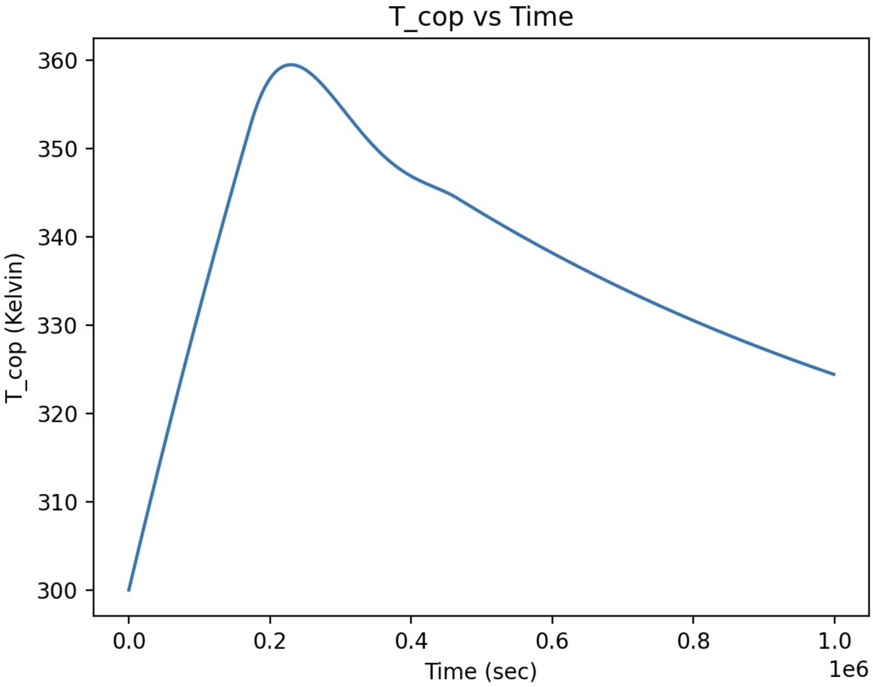


v) We can model this system by adding a dQ/dt term as input to our state space equation.

Modelling a constant dQ/dt which can now be clubbed into out input to create a new effective input.



```python
import control as ct
import matplotlib.pyplot as plt


a = -1 / (8935 * 100)

sys =  ct.ss(a, -a, 1, 0)

time = [i*1000 for i in range(1000)]

#Assuming a constant heat source for the first 250 thousand seconds(1/4th of the total time))
#Since the heat source is an additional constant in the equation, I assume a heater input
#whose effect mimics a surrounding Temperature of 600

U = [0 for i in range(1000)]
for i in range(1000):
    if (i < 250):
        U[i] = 600
    else:
        U[i] = 300

t, x = ct.input_output_response(sys, time, U, X0 = 300.0)

plt.plot(t, x)
plt.title("T_cop vs Time")
plt.ylabel('T_cop (Kelvin)')
plt.xlabel('Time (sec)')
plt.show()
```



Having observed the decaying rate of change of Temperature, I tried to model a situation where there is a High energy source providing energy for a short duration. As can be seen in the graph, the heat energy can be stored by the cube of Cu for a longer duration than the applied input. Therefore, this model can be used like a thermal capacitance, storing thermal energy

References:

[1] Investigations of temperature dependences of electrical resistivity and specific heat capacity of metal - https://www.sciencedirect.com/science/article/pii/S0921452616301090

[2] Heat Capacity of Reference Materials: Cu and W - https://srd.nist.gov/JPCRD/jpcrd263.pdf

From the results of the papers, we can observe that heat capacity of copper starts saturating near 300K and doesn’t change much from 300-400K. Hence, we take the specific heat of copper to be a constant of 385 J/Kg K

Question 3:

An attempt to measure the viscosity of silly putty using experiments which rely on low viscosity such as Stokes law (F = 6πηrv) seem to fail due the high viscosity of silly putty even at zero external stress and strain. [1]

Trying to measure certain solid properties of silly putty will lead to problems due to the it flowing under its own weight. This can be solved by filling the silly putty in a very flimsy balloon so as to conserve its shape to an extent. Care must be taken while filling the silly putty ensuring that it fits snug but not so tight that the balloon exerts a compressive force on the putty.

Shear Stress vs Strain Rate:

This relationship can be computed in two ways:

i) Using 2 horizontal plates with silly putty between them.

Consider a setup with two horizontal plates separated by a distance ‘y’ with silly putty between them. One of the plates is fixed to the ground while the other just rests on top of the material. A mass ‘m’ is connected to the top plate via a pulley so there is a constant external force (F=mg) acting on it. We measure the instantaneous velocity ‘v’ of the top plate at each instant using a motion sensor.

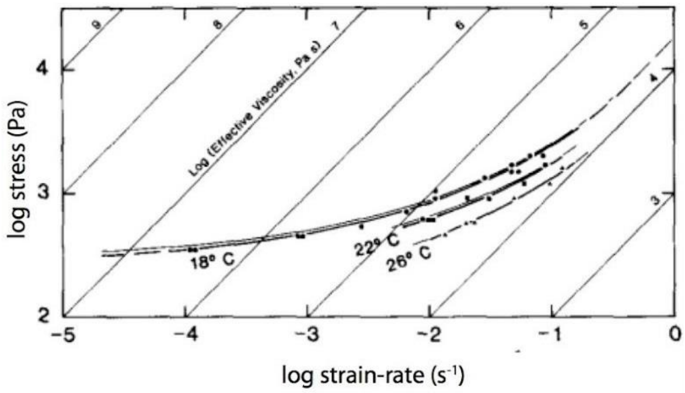
Assuming a linear change in velocity of silly putty, this will give us strain rate using,

$$\dot{e}(t) = \frac{v(t)}{y}$$

By plugging these values in the viscosity equation for shear-thickening fluids we can calculate various mechanical properties of our silly putty material, like, the time dependence of viscosity, Modulus, etc.

ii) Viscometer

Power-law for viscosity states that $\eta = K\boldsymbol{\gamma}^{n}$
Where K is a material-based constant. Silly putty, which is a dilatant fluid, has a characteristic n~ 7 at low strain rates, between 10^{-4} and 10^{-2} s^{-1}, but at high strain rates, over 1 s^{-1}, the theological behaviour is approximately linearly viscous. [2][3]



To measure this behaviour, we can construct the following setup:
Consider a viscometer-like setup with two coaxial cylinders and the silly putty in the middle. We connect a precise motor to the inner cylinder whose torque can be manually controlled. But instead of the traditional way of maintaining the rotation speed of the motor by current, we maintain the torque and calculate the dependence on viscosity and angular velocity.

$$\tau = \frac{\eta\, A\, r^{2} \omega}{y}$$

Thus, by some simple calculations, we obtain the plots of

1. Viscosity vs time for different values of applied torque
2. Angular velocity vs time
3. Relaxation time vs Torque\

Most of the mechanical properties of the material can be deduced from these plots. The relaxation time of silly putty can be deduced from the angular velocity vs time graph. The time at which there is a sharp change in the graph corresponds to relaxation time.

Spring Constant:

We would need to fill the putty inside the balloon as mentioned above in order measure any effect on oscillations. Based on my intuition, measuring changes from a known oscillation will be easier and more sensitive than measuring the small oscillations due to the silly putty itself.

Hence we will utilise a spring of known spring constant and connect the silly putty balloon in parallel with it. We will now attach a bob of known mass and give a small displacement to it. The time period of oscillation of the bob can be measured and used to derive the effective spring constant of the system.

Using the formula for springs in parallel,

$$k + k^{1} = k^{eq}$$

we can compute the spring constant of the spring.

Impulse required to Break the silly putty:

Upon providing a high enough impulse, the silly putty will break apart. The impulse required to break the silly putty would be inversely related to its size. A piston which can deliver a controllable impulse is used for this. The impulse on a silly putty of specific size is slowly increased till it breaks. This is repeated for different sizes of silly putty. Plotting this data gives us an estimate of how the impulse required to break the putty changes with size. The fracture point on the Stress vs Strain graph of the silly putty can be derived from this.

References:

[1] Cross, Rod. (2012). Elastic and viscous properties of Silly Putty. American Journal of Physics. 80. 870-875. 10.1119/1.4732086.

[2] Dixon JM and Summers JM (1985) Recent developments in centrifuge modelling of tectonic processes: equipment, model construction techniques and rheology of model materials. Journal of Structural Geology 7: 83 – 102

[3] Dixon JM and Summers JM (1986) Another word on the rheology of silicone putty: Bingham. Journal of Structural Geology 8: 593 – 595