First Order and Second Order System

Group Number	**17**
Team Members	**S.S.S.Vardhan (19CH30018)** **Saksham Arya (19CH100** **Aayush Airan (19CH**

Objective

- To study the step response of a thermometer.

Theory

A thermometer bulb is a first-order system, whose response can be described by a first-order linear differential equation. The dynamic control configuration of a system is described by means of differential equations. The order of differential equations representing the dynamics of the system is the same as that of the order of the control system.
The dynamic response of first-order type instruments to a step change can be represented by

$$T\frac{d\theta}{dt} + \theta = \theta_F$$

Where,

θ = temperature indicated by thermometer

θ_F = Final steady-state temperature

t = time

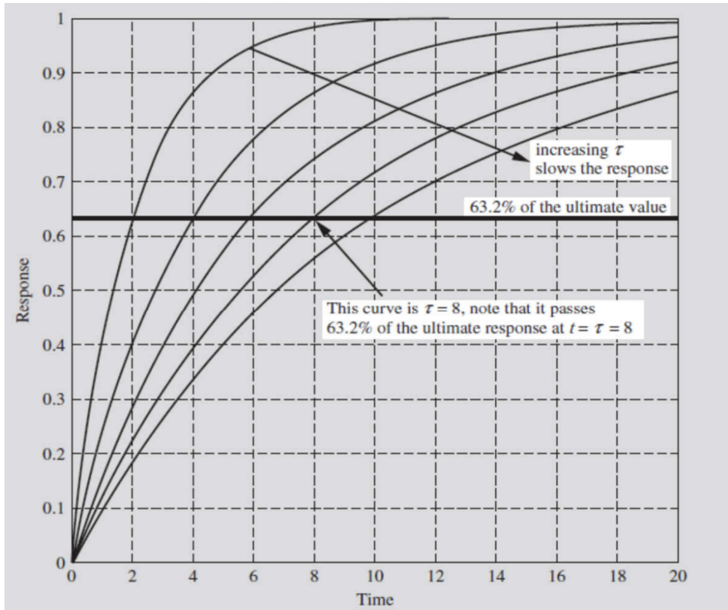
T = time constant

The linear first-order differential has the particular solution for given initial conditions,

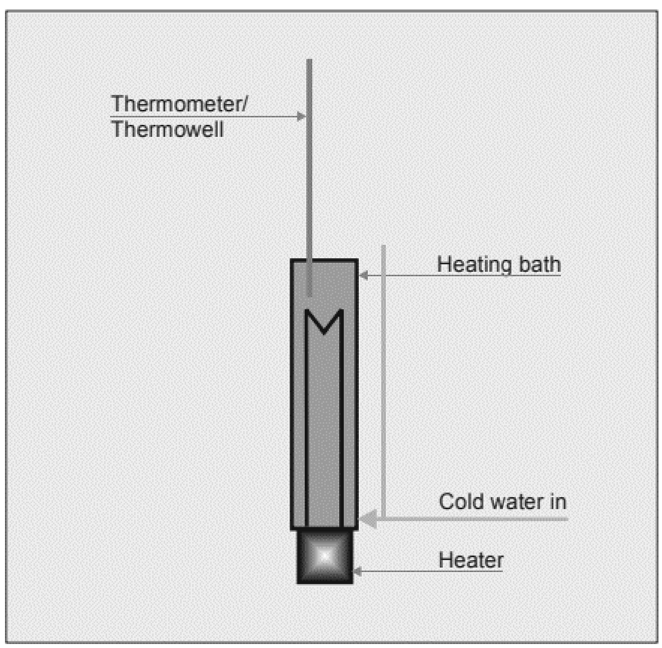
$$\frac{\theta}{\theta_F} = 1 - e^{-t/T}$$

The time constant T is the time required to indicate 63.2% of the complete change. The time constant T is numerically equal to the product of resistance and capacitance.

A sample response has been shown below



Experimental Set-up



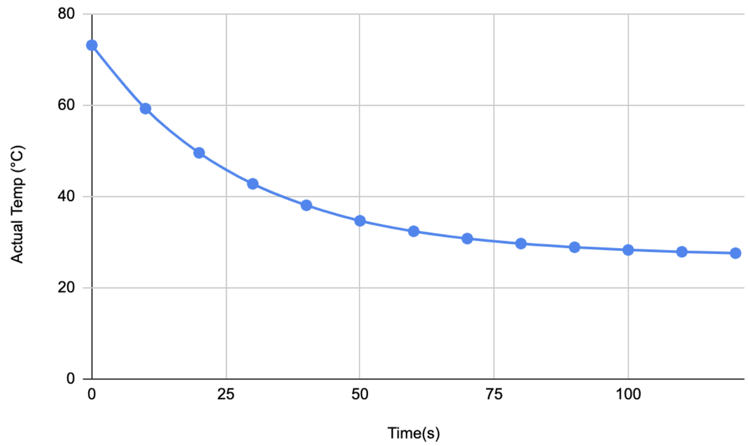
The setup consists of a ‘U’ tube manometer, heating bath, thermometer, thermowell, beeper for recording observations and timer for heater on-off operation. The components are mounted on the base plate. The set-up is tabletop mountable

Procedure

Fill the heating bath with clean water by opening the inlet valve of the heating bath. Then switch on the beeper and set the beep interval to 3 seconds. Ensure that the cyclic timer is set to 30 seconds on time and 30 seconds off the time. Switch on Mains to heat the water in the heating bath to its boiling point. Switch off the mains. The water in the heating bath is now near its boiling point. Insert the thermometer in the heating bath suddenly after noting its initial temperature.
Note the thermometer reading at each beep interval till the temperature reaches a steady state. Switch off the beeper and fill up the readings observed in “Observations” below.

Observation

Time(s)	Temp(deg C)
0	73.2
10	59.3
20	49.6
30	42.8
40	38.1
50	34.7
60	32.4
70	30.8
80	29.7
90	28.9
100	28.3
110	27.9
120	27.6



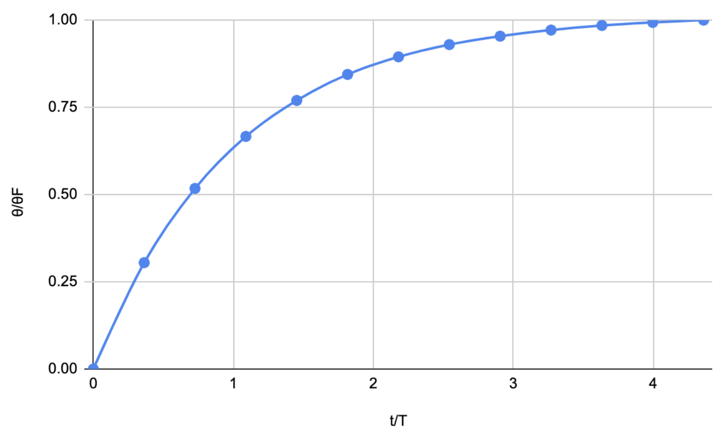
Result

1. Step change = Final temp. – Initial temp. = -45.6 ^{0}C
2. Value of 63.2% of step = 0.632 x (Final temp. – Initial temp.) + Initial temp. = 44.38 ^{0}C
3. From the graph of Actual temperature Vs time, we note the value of time at 63.2% of step change. This value is the observed time constant of the thermometer. Therefore, time constant (T) = 27.52 s
4. θ_F = Final temp. – Initial temp. = -45.6 ^{0}C

Time(s)	**Temp(deg C)**	**θ**	**t/T**	**θ/θ_F**
0	73.2	0	0	0
10	59.3	13.9	0.363372093	0.3048245614
20	49.6	23.6	0.726744186	0.5175438596
30	42.8	30.4	1.090116279	0.6666666667
40	38.1	35.1	1.453488372	0.7697368421
50	34.7	38.5	1.816860465	0.8442982456
60	32.4	40.8	2.180232558	0.8947368421
70	30.8	42.4	2.543604651	0.9298245614
80	29.7	43.5	2.906976744	0.9539473684
90	28.9	44.3	3.270348837	0.9714912281
100	28.3	44.9	3.63372093	0.9846491228

110	27.9	45.3	3.997093023	0.9934210526
120	27.6	45.6	4.360465116	1

Plot of θ/θ_F vs t/T:



Sample Calculations (Python Code):

We use the relation

Theoretical temp. =

Initial temp. +(Step change x (1-EXP ($\frac{-1\ x\ Time}{Time\ constant\ (From\ graph)}$)))

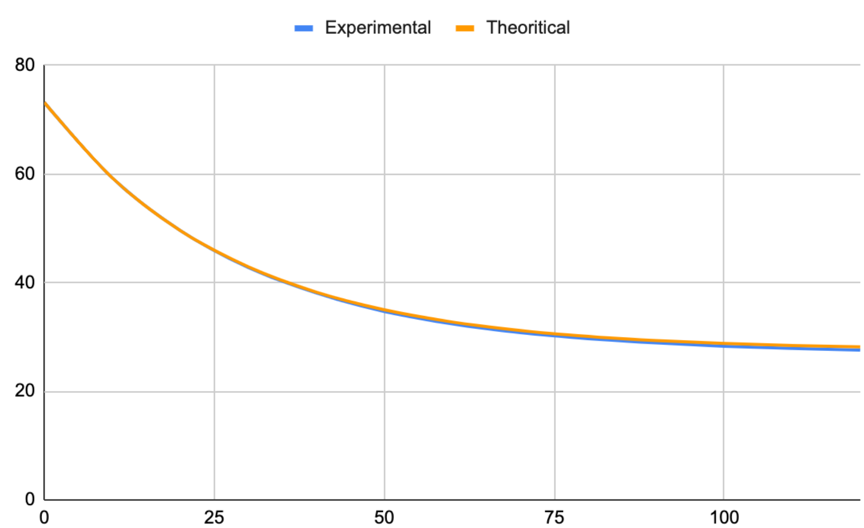
Automating using python code

```
x = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]
InitialTemp = 73.2
StepChange = 45.6
TimeConstant = 27.52
for time in x:
  theoretical_temp = InitialTemp;
  theoretical_temp -= (StepChange)*(1 - 2.718**(-1*(time)/TimeConstant))
  print(theoretical_temp)
```

We get the following table,

Time	Theoretical Temperature	Experimental Temperature
0	73.2	73.2
10	59.3	59.3
20	49.6	49.6
30	42.93	42.8
40	38.26	38.1
50	35.01	34.7
60	32.75	32.4
70	31.18	30.8
80	30.09	29.7
90	29.33	28.9
100	28.8	28.3
110	28.43	27.9
120	28.18	27.6

<u>Plot of Actual vs Theoretical</u>



Discussion

- The first-order control systems are not stable with ramp and parabolic inputs as theri responses go increase even at an infinite amount of time.
- As the impulse and step inputs have bounded output, the first-order control systems are stable. But, the impulse response does not have a steady-state term.
- The step signal is used widely in the time domain for analyzing the control system from their responses.

Results

- The observed step response tallies fairly with the theoretical step response.

First Order and Second Order System

Group Number	**17**
Team Members	**S.S.S.Vardhan (19CH30018)** **Saksham Arya (19CH100** **Aayush Airan (19CH**

Objective

- To study the step response of a mercury manometer.

Theory

The dynamic response of a second-order system to a step change can be described by a second-order differential equation. A second-order linear system is a common description of many dynamic processes. The response depends on whether it is an overdamped, critically damped, or underdamped second-order system. The solutions to the above equation involve three cases: an under damped condition [$\zeta<1$], critically damped condition [$\zeta=1$] and over damped condition [$\zeta>1$]. The response for under damped system [i.e. $\zeta<1$] can be written as:

$$y(t) = KM \times \left\{ 1 - e^{-\zeta t/\tau} \left[\cos\left(\frac{\sqrt{1-\zeta^2}}{\tau} t \right) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin\left(\frac{\sqrt{1-\zeta^2}}{\tau} t \right) \right] \right\}$$

In case of manometer:

y(t) = response at any time t after step change (deviation value).

K= Gain factor =1

M= magnitude of step change,

Damping coefficient (ζ)= $\dfrac{8L\mu}{\rho g D^2}\sqrt{\dfrac{2g}{L}}$

L = Column length in meter

μ = Dynamic viscosity in Kg/m.s.

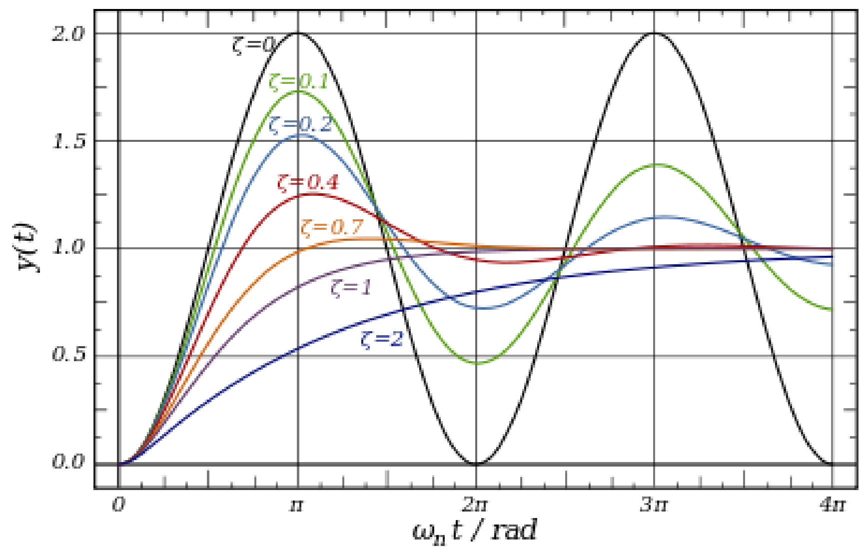
ρ = Mass density of the manometer fluid in kg/m^3

D = tube diameter in m,

g = Gravitational acceleration in m/sec^2

Characteristics time (τ) = $\dfrac{2\pi}{\omega_n}$ in sec.

Following figure shows the response of second-order system for different damping coefficient.



<u>Experimental Set-up</u>

Procedure

1. Ensure that mercury level in manometer is set at '0' on the scale.
2. Close vent connection by putting finger on it.
3. Adjust the needle valve and vent to raise the mercury level to @200mm from '0' level.
4. Note the mercury level reading and quickly open the vent to apply step change.
5. Note the top peak and bottom peak readings. Also simultaneously note the period of oscillation. (This can be noted by measuring time required for 4-5 oscillations and then calculating for each oscillation)
6. Repeat process 2-3 times for different step changes.

Experimental Data

Constants:

Manometer fluid = Mercury

Dynamic viscosity (μ) = 0.0016 Kg/m.s.

Mass density (ρ) = 13550 Kg/m^3.

Column length (L) = 0.760 m

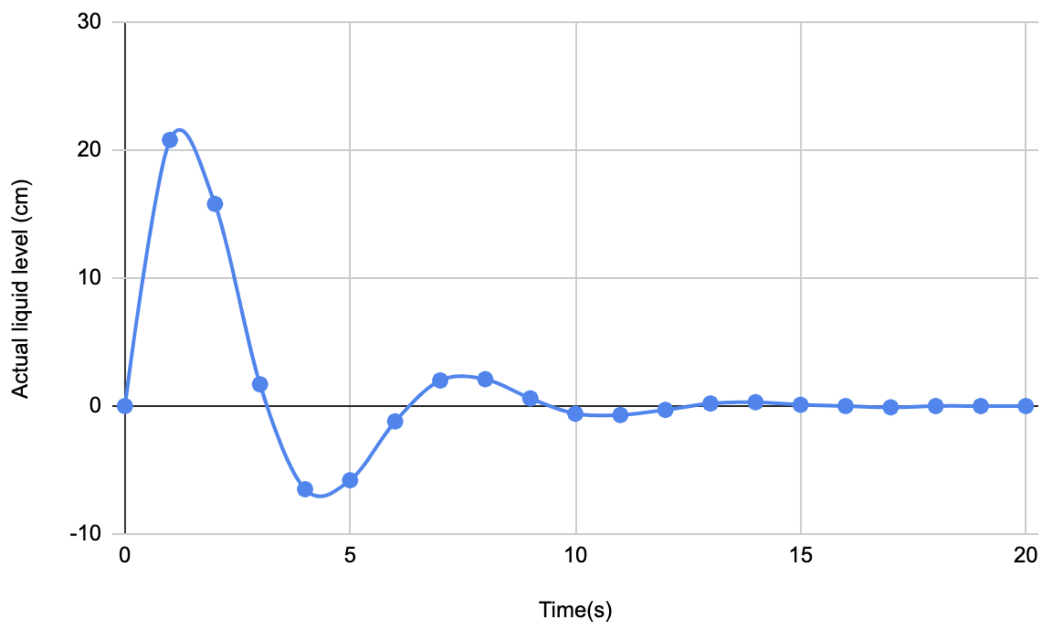
Tube diameter (d) = 0.005 m

Step change (cm) = 31.2 cm

Time (s)	Actual Liquid level(cm)
0	0
1	20.8
2	15.8
3	1.7
4	-6.5
5	-5.8
6	-1.2
7	2
8	2.1
9	0.6
10	-0.6
11	-0.7

12	-0.3
13	0.2
14	0.3
15	0.1
16	0
17	-0.1
18	0
19	0
20	0

Plot of Actual Liquid level(cm) vs Time(s):



Results (with sample calculations)

Natural frequency (ω_n) = $2\pi\sqrt{\frac{2g}{L}}$ in rad/sec

Damping coefficient (ζ)= $\frac{8L\mu}{\rho g D^2}\sqrt{\frac{2g}{L}}$

Period of oscillation = **P=** $\frac{2\pi}{\omega_n \times \sqrt{1-\zeta^2}}$

DR = Exp($\frac{-2\times \pi \times \zeta}{\sqrt{1-\zeta^2}}$)

Overshoot (OS) = a/b = $\sqrt{Decay\ ratio}$

Frequency of damped oscillation (f)= $\frac{\omega_n \times \sqrt{1-\zeta^2}}{2\pi}$ in cps.

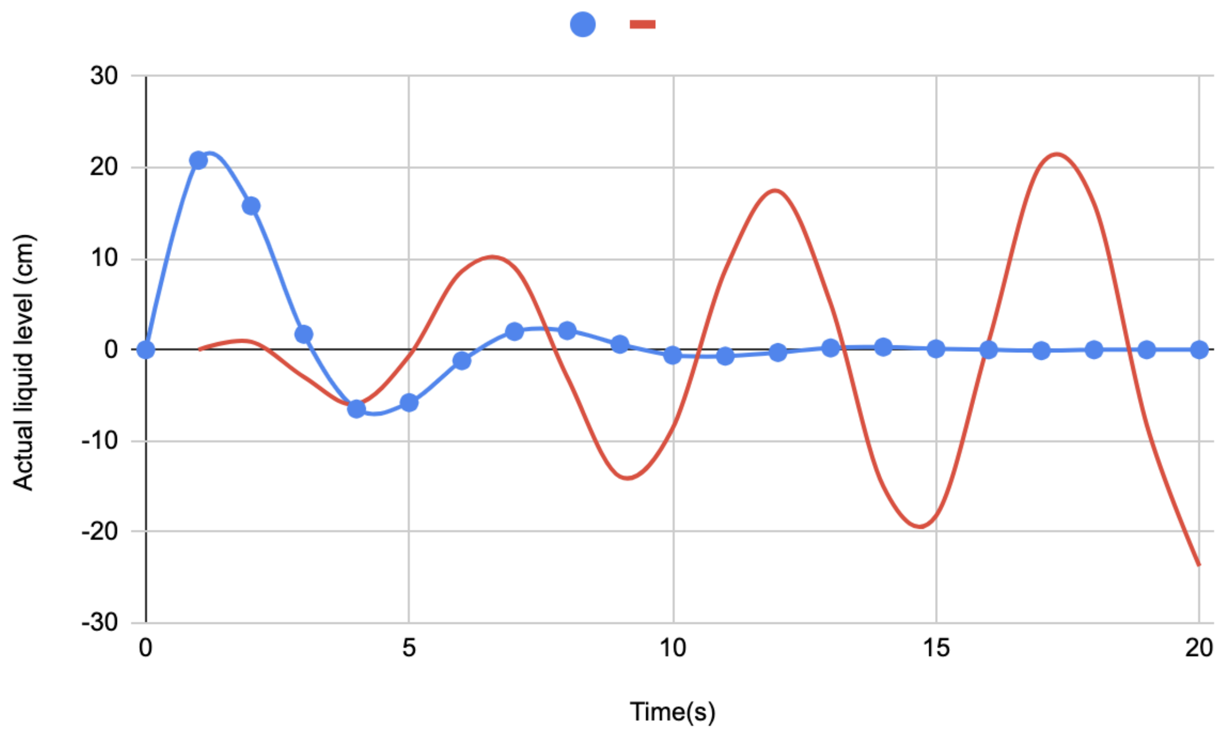
Characteristics time (τ) = $\frac{2\pi}{\omega_n}$ in sec.

Code

```
DynamicViscosity = 0.0016 # Kg/m.s.
MassDensity = 13550 # Kg/m3.
ColumnLength = 0.760 # m
TubeDiameter = 0.005 # m
StepChange = 31.2 # cm

NaturalFrequency = 2*3.14*((2*10/ColumnLength)**(0.5))
print("Natural Frequency:", NaturalFrequency)
DampingCoefficient = (8*ColumnLength*DynamicViscosity*((2*10/ColumnLength)**(0.5)))/(MassDensity*10*(TubeDiameter**2))
print("Damping Coefficient:", DampingCoefficient)
PeriodOfOscillation = 2*3.14/(NaturalFrequency*((1 - DampingCoefficient**2)**(0.5)))
print("Period Of Oscillation:",PeriodOfOscillation)
DecayRatio = (2.718)**((-2*3.14*DampingCoefficient)/((1 - DampingCoefficient**2)**0.5))
print("Decay Ratio:", DecayRatio)
OverShoot = (DecayRatio)**(0.5)
print("OverShoot:", OverShoot)
Frequency = 1/PeriodOfOscillation
print("Frequency:",Frequency)
CharacteristicsTime = 2*3.14/(NaturalFrequency)
print("CharacteristicsTime:", CharacteristicsTime)
```

Plot of Theoretical and actual liquid level v/s Time:



Red represents theoretical

Blue represents Actual

Time (s)	Actual Liquid level(cm)	Theoretical(cm)
0	0	0
1	20.8	0.88
2	15.8	-2.98
3	1.7	-5.97
4	-6.5	-0.668
5	-5.8	8.619
6	-1.2	9.023
7	2	-3.01
8	2.1	-13.92

9	0.6	-8.64
10	-0.6	8.75
11	-0.7	17.43
12	-0.3	5.07
13	0.2	-15.01
14	0.3	-18.24
15	0.1	0.925
16	0	20.382
17	-0.1	16.033
18	0	-8.251
19	0	-23.745
20	0	-11.0182

<u>Discussion</u>

- It is very difficult to design a system in real life which is critically damped. It will either end up as overdamped or underdamped.
- As we will never know whether an overdamped system has reached steady-state or not, most practical systems are made to be underdamped.

<u>Conclusion</u>

Natural Frequency: 32.21572025547384

Period Of Oscillation: 0.19495704308419406

Decay Ratio: 0.9116351383267647

OverShoot: 0.9547958621227706

Frequency: 5.129335079051955

CharacteristicsTime: 0.19493588689617927

The plot shows that this is an underdamped system with a damping coefficient of 0.01473