Robotics and
Machine Intelligence

RMI Inductions ‘23

INTERMEDIATE TASKS

Weekly Tasks Track

Greetings from RMI!

Congratulations on successfully completing the basic tasks. We hope that you have got a good exposure to the various domains involved in robotics. As mentioned before, the inductions will be conducted in two stages:

1. The Practical Implementation Stage: Here, you will be given a set of tasks to work on, that will facilitate your learning and understanding.
2. The Personal Interview Stage

The Practical Implementation Stage is split into three sections: Basic, Intermediate, and Advanced. This document outlines all the Intermediate tasks to be completed. The intermediate tasks are designed in such a way that you can implement the concepts you might have learnt while attempting the basic tasks and introduce you to other new exciting concepts which is unique to the domain.

You might face issues or difficulties in completing the tasks, but do not give up. It is quite normal to hit roadblocks while progressing. In any case, feel free to contact any member of the club.

Rules:

1. Participants can attend any number of intermediate tasks from any domain. The number of questions attended and the overall points will be taken into consideration for evaluation.
2. Topics associated with each task are also listed. Please go through them to understand the concepts behind the tasks.
3. It is good to work in teams, but the tasks are not meant to test your teamwork. The problems must be solved individually. Learning how everything works is key. Make sure you understand everything well.
4. Create a drive folder named “<Roll Number> - Intermediate Tasks” and sub-folders named “<Roll Number> - <Domain Name>” for each

of the domains. Create folders for each task in a particular domain. Follow domain specific submission guidelines.

General Instructions:

1. Go through the problem statement and understand the task well before you start working on it. A task misunderstood and completed will not be considered. If you have trouble understanding it feel free to contact anyone of us.
2. We expect you to understand the concepts and write your own code. It is acceptable to refer to code snippets online but copying the code is not an option. You will be extensively questioned based on the code and will be asked to make modifications during PI's.
3. Divide a complex task into modules and work on the individual modules. Completing the modules will be appreciated even if you are unable to complete the entire task.
4. Partial submissions are accepted. What we look for is your approach and zeal, rather than the entire solution itself.
5. The Internet is your biggest resource. All your doubts are a Google search away.
6. You can always reach out to us regarding any doubts, queries or issues through the WhatsApp group or PM one of us through the same. We will be glad to help.

HAPPY ROBOTICS!!!

Electronics and Embedded Systems

Instructions:

1. You can implement the solution to these tasks in either Proteus VSM software or Tinker CAD.
2. Task 3 to Task 5 must be implemented in Embedded C only. One can use Arduino IDE, Atmel Studio, or any other coding environment that supports Embedded C.
3. You can use any AVR microcontroller for implementing the tasks. We would recommend using the Arduino UNO development board, which has the ATmega328p microcontroller to get started, in case you are a beginner
4. All the code must be commented well enough, and the circuit labelled for easier understanding.

Task 1:

Create a 4-bit BCD counter using logic ICs and display it in 7 segment display.
NOTE: No microcontroller should be used.
HINT: Use flip-flop
Topics to Learn: Binary counters, decade counter, flip-flop.

Task 2:

Build a door alarm using 555 timer IC and basic circuital elements. Green LED should blink when the door opens, and Red LED should blink when it closes. Start the countdown using the 555 timer IC when the door opens, and the buzzer should start beeping if the door is kept open for more than one minute.

NOTE: Consider slide-switch as the door.

Topics to Learn: Working and different operation modes of 555 timer.

Task 3:

Display the speed (in RPM) of a DC motor with an encoder in a LCD display.

Topics to Learn: Working of DC Motor with encoder.

Task 4:

Build a timer whose ON/OFF operation is controlled by a push button. When the timer starts, a green LED should light up and turn OFF when the timer is stopped. A Red LED should glow when the timer reaches 10secs.

NOTE: Use an external interrupt for getting input from the push button. Don't use any additional header files other than util/delay.

Topics to Learn: AVR GPIO, External interrupt.

Bonus: Utilize AVR timer registers to generate the required delay.

Task 5:

Sweep the servo motor by adjusting the potentiometer. Read the signal from the potentiometer on an analog pin by implementing ADC conversion and display it in the serial monitor.

NOTE: Don't use analogRead() function**.**

Bonus: Sweep the shaft of the servo motor by generating PWM signals.

Topics to Learn: UART, ADC, Servo working and control, PWM wave generation.

Learning resources:

1. 555 Timer Tutorial - The Monostable Multivibrator (electronics-tutorials.ws)
2. Timer in AVR ATmega16/ATmega32 | AVR ATmega Controllers (electronicwings.com)

Submission Guidelines:

1. Make a folder titled "_Tronix" .
2. Make sub-folders titled with <Q_Questionno>, eg. "Q1" for question 1.
3. Upload all files related to the task into the folders.
4. For TinkerCAD simulations, you can attach a text file with the TinkerCAD circuit link and the Arduino Code. Ensure your circuit is set to public access before sharing. Attach a screenshot of the

circuit. The screenshot must cover the whole window/screen and must not be cropped.

5. For Proteus simulations, you can upload the circuit file and the Arduino program files.
6. Upload this folder in your drive and make the drive link sharable and fill it in the Submission form.

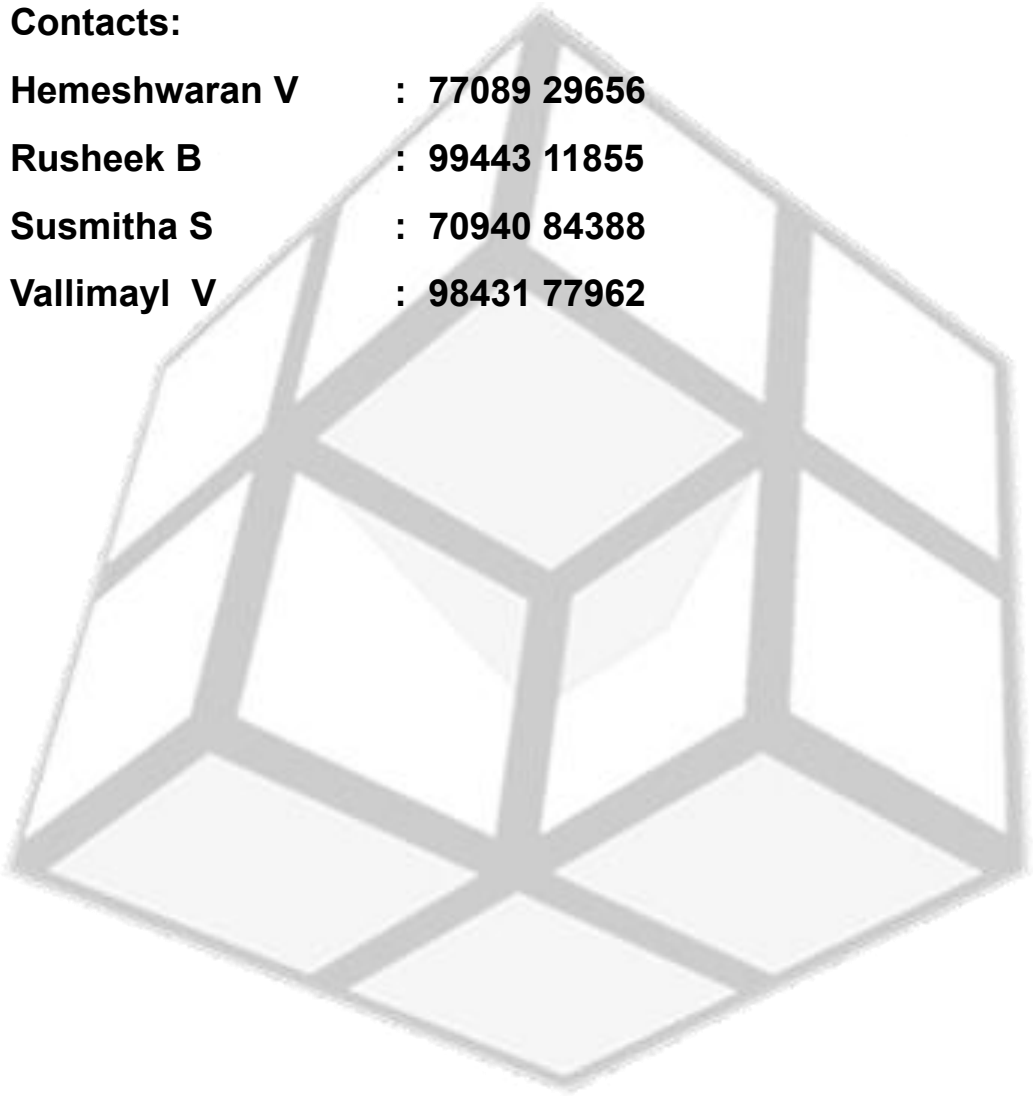
Contacts:

Hemeshwaran V : 77089 29656

Rusheek B : 99443 11855

Susmitha S : 70940 84388

Vallimayl V : 98431 77962



Controls

“A robot is an autonomous system which exists in the physical world, can sense its environment (perceive), and can act on it to achieve some goals”

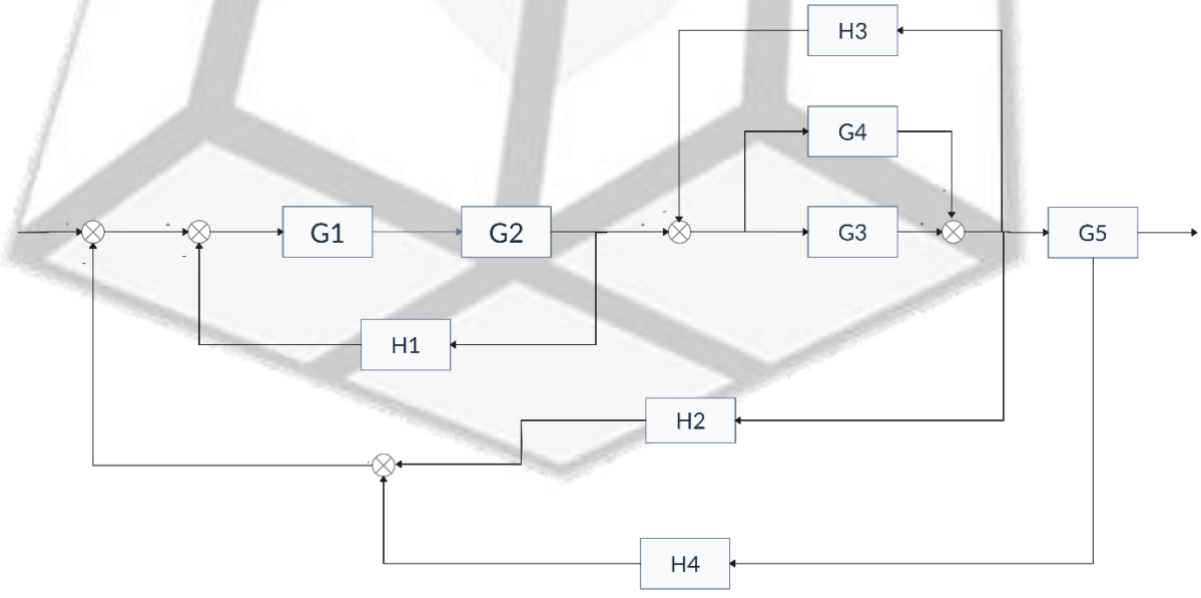
- Professor Matarić’s

With the help of various sensors, a robot can understand the world around it. We use this data and do some computation to achieve our desired goal. How does a robot know if it has achieved its goal or not? – It can only understand it through sensor feedbacks, this is how control systems come into play.

A control system manages commands, directs, or regulates the movement and functions of various parts of the robot, as well as executes a specific set of forces and motions

Task 1:

Simplify the given closed loop control system, and find it’s transfer functions. The labels G, H denote the impulse response of that system.



Task 2:

Use *MATLAB's Simulink* to design a unity gain closed loop control system, that has the following transfer function.

$$H(s) = \frac{Y(s)}{X(s)} = \frac{s(s-3)}{(s+1)(s+4)(s^2+12s+35)}$$

Also find the following,

- Zeros
- Poles
- Dominant Pole
- Time Constant

Task 3:

Torsional and resonance vibrations on a crankshaft of an engine is controlled by a harmonic damper. The differential equation of such a system is given by,

$$107\frac{d^2x}{dt^2} + 123\frac{dx}{dt} + 161x = 0$$

where,

$x \rightarrow$ displacement of crankshaft in cm

$t \rightarrow$ time in s

Model the system in the *Simulink* with the initial condition $x(0) = 0.3\ cm$ and $x'(0) = 0.0231\ cm/s$. Also, plot the displacement vs. time graph.

Task 4:

You need to design a temperature controller, which should increase or decrease the environment temperature based on its internal process, to maintain the temperature at $37℃$. The transfer function of the internal process if given below.

$$\frac{0.051s + 0.083}{s^2 - 4s + 5}$$

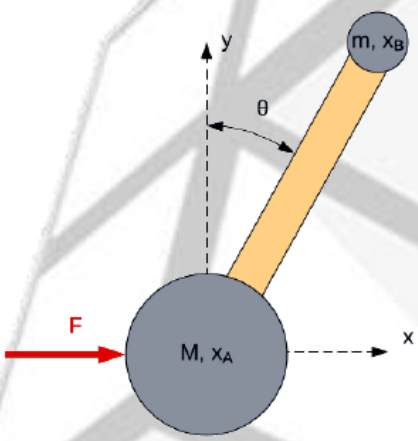
Design a PID controller to raise an ice cube to the desired temperature within 12 seconds and plot its temperature vs. time graph in *Simulink*.

Task 5:

Given below is an inverted pendulum model of *Two-Wheeled Self Balancing Bot* and its equations of motion.

Define state space model of a system.

A state space model for the above system is given below.



$$x: \quad (m+M)\ddot{x} + ml\ddot{\theta} = F$$

$$\theta: \quad \ddot{x} + l\ddot{\theta} - g\theta = 0$$

$$x'' = \frac{F}{M} - \frac{mg}{M}\theta$$

$$\theta'' = -\frac{F}{Ml} + \frac{(m+M)g}{Ml}\theta$$

Input: [F]

Output: [θ]

State space variables: [$x \;\; x' \;\; \theta \;\; \theta'$]^T

Write its *system, input, output* and *feedforward matrices* for the above model. Also check, whether the system is controllable, and its state space variables can be observed from its outputs.

Submissions Instructions:

1. Make a folder titled “_Controls”.
2. Make sub-folders titled with, eg. “Q1” for question 1.
3. Upload all files related to the task into the folders.
4. Upload this folder in your drive and make the drive link sharable and fill it in the Submission form

Contact:

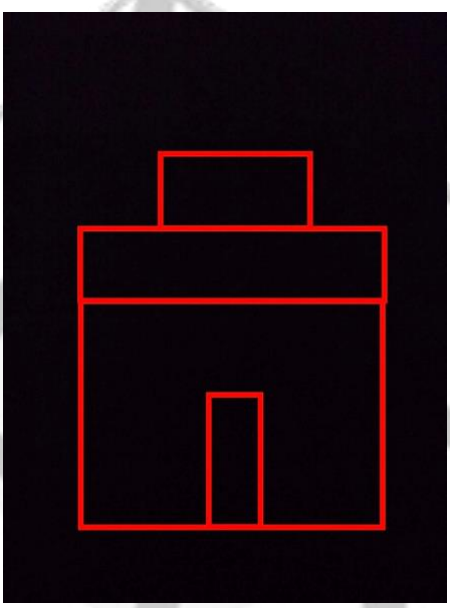
Ganishk D : +91 95668 53093

Computer Vision

Task 1:

Create any number of trackbars. Using the coordinates obtained from these trackbars, draw a picture of the following house on a blank canvas, and save the resultant image.

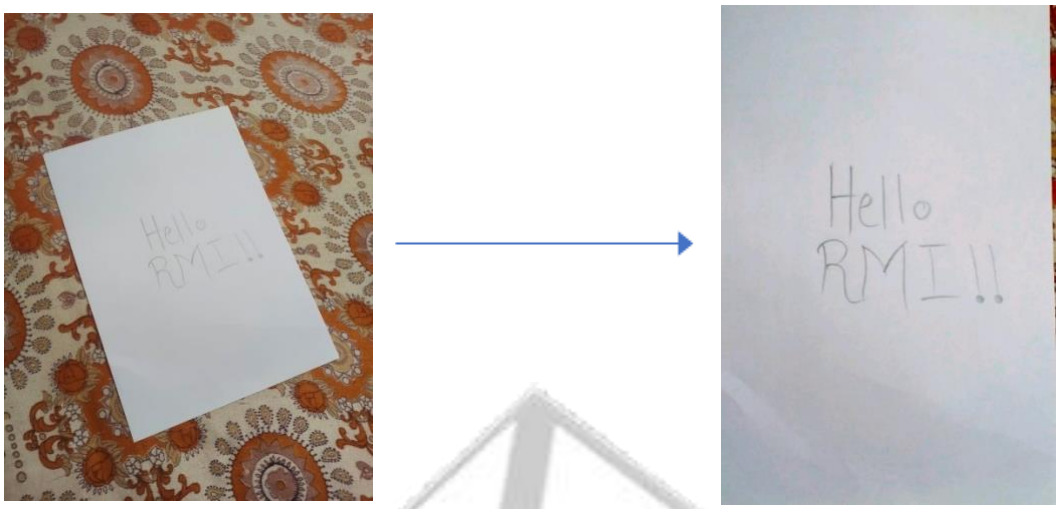
Concepts to learn : Creating trackbars, Drawing in OpenCV



Task 2:

Create a document scanner using openCV. Upload a picture of a document which is not aligned with the camera frame properly, and use computer vision techniques to warp the image in such a fashion that it is perfectly in frame. Also apply filters to improve the clarity of the text. Save the resultant image.

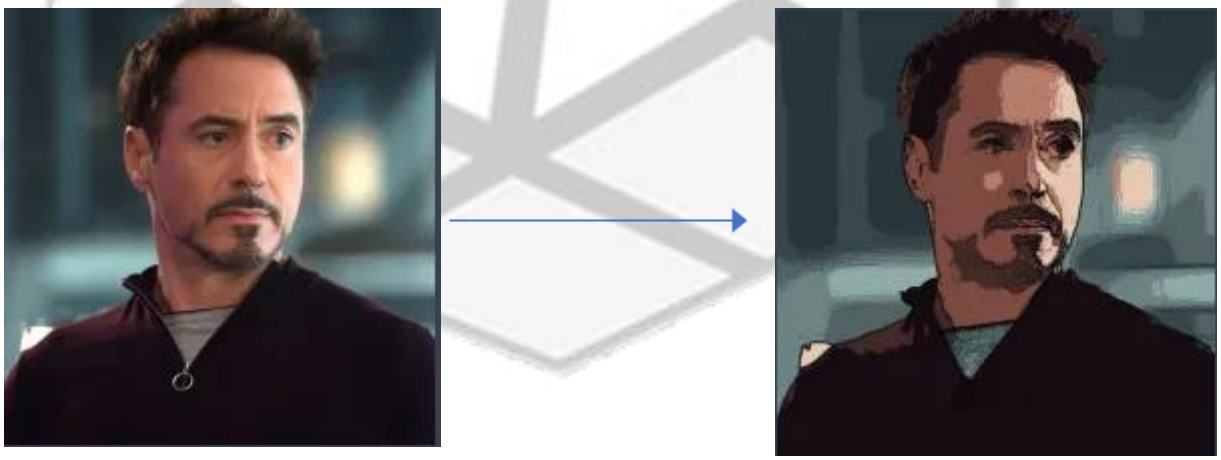
Concepts to learn: Image warping, Image filters in OpenCV



Task 3:

Cartoons play a very impactful role in today’s journalism. They draw the reader's attention to minute details and help them perceive things in a different way. They explain and explore stories in a way that articles cannot.
Given to you is the picture of an infamous global leader. Use OpenCV to convert his picture into a cartoon.

For example:



Link to picture

Task 4:

Create an object tracker or a motion detector to track the moving objects in the live feed by using edge detection techniques. You are free to use any colour coding or drawing technique to highlight the detected edges effectively.

Concepts to learn: Live feed processing, edge detection algorithms.

Task 5:

Isolate an object of any colour of your choice from the background. Use this object to simulate a very popular DVD screensaver. Said object should move across the screen, and deflect upon collision with the boundary of the background (which can be any image of your choice). Also change the colour of the object with each collision.

Screensaver to simulate: https://youtu.be/-pdVUsCqd2U

Concepts to learn: Drawing in OpenCV, Separating an image from it's background

Submission Guidelines:

1. Make a drive folder titled “Computer_Vision” in the main folder.
2. Make sub-folders titled with <Q_questionno>, eg. “Q1” for question 1.
3. Upload all files related to the task into the folders.
4. Files to be uploaded include the .py file, along with the screenshots of output terminal, input image and output image. The screenshot must cover the whole window/screen and must not be cropped.
5. For each question, upload a one-shot video of the code working and explanation of the same. You can either screen record using any of the available software or use a mobile camera for the same. But ensure the whole window is captured clearly and your voice is audible while explaining.

Contacts:

Aditya Singh : 75185 36909

Gayathri Rajesh : 81977 68603

Mechanics

Task 1:

Aim: To understand manipulator kinematics.
Task:

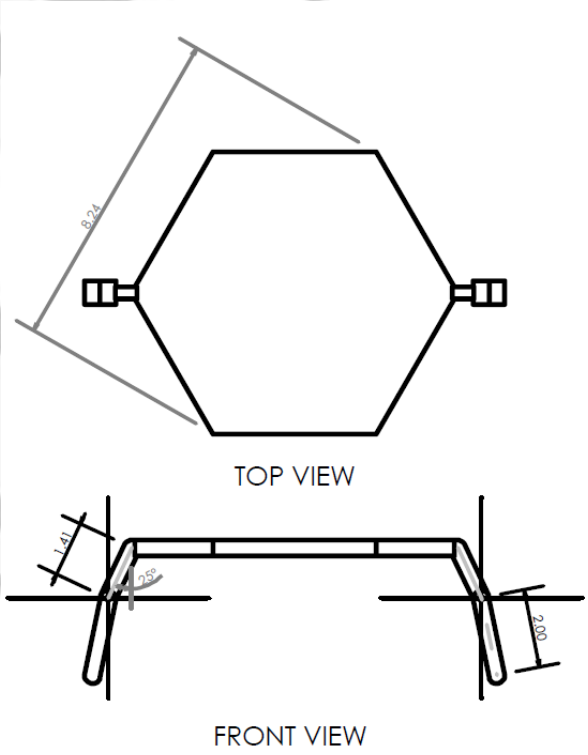
a. What is forward kinematics and inverse kinematics? What is the difference between them?
b. Out of these two, which one do you think is mostly used in robotics and why? What do you understand by the term homogeneous transformation matrix?

Topics to learn: Forward kinematics, and inverse kinematics.

Task 2:

Aim: To implement manipulator kinematics
Task:

Given below are the top view and front view of a robot with a regular hexagonal body and two legs $l1$ and $l2$ of equal length rigidly attached to the body separated by a distance of 8.24 cm on the x axis. The length of $l1$ and $l2$ is 1.41 cm. The axis of the $l1$ and $l2$ are at an angle of $65°$ to the hexagonal surface. Assume the thickness of the body to be zero. Two more legs $l3$ and $l4$ are attached to the legs $l1$ and $l2$ respectively with revolute joints. The length of the links $l3$ and $l4$ is 2.0 cm. θ_1 and θ_2 are the angles made by legs $l3$ and $l4$ with the axis of $l1$ and $l2$ respectively.



How many degrees of freedom does this robot have? The robot moves in such a way that it's centre of mass always lies above the plane z=-1. Determine the maximum separation that can be achieved between the robot's legs.

Find the range of θ_1 and θ_2. Find the maximum height and minimum height achieved by the robot.

Task 3:

Aim: Understanding quadrotors and their types of movements.
Task:

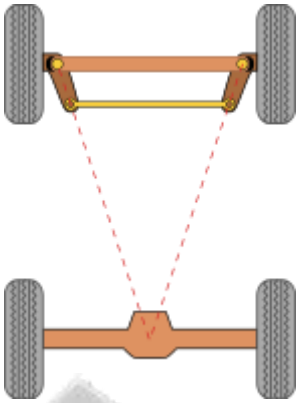
a. What are four fundamental quadcopter movements? Explain them in brief. Let's say a quadcopter drone is pitching forward by an angle. In this case express the lift vector produced by each motor in the inertial frame.
b. Let's assume that one of the motors of your quadcopter drone is damaged. Let's say you can control the speeds and directions of remaining motors. Can you perform roll, pitch and yaw with your quadrotor? Specify the speeds and directions of each motor for every type of movement that is possible.

Topics to learn: Quadrotor mechanics.

Task 4:

Aim: To get to know about steering systems and their importance.
Task:

a. What are the basic steering systems that are being employed in four wheeled vehicles? What's meant by the turning radius of a vehicle? Think of ways to reduce the turning radius for four wheelers.
b. In the Ackermann geometry shown below, let's say that you can turn each wheel individually. Then write a computer program which computes the angle with which each wheel must be turned for a given turn radius. The length of the wheelbase is "L" m. Assume the wheels to be cylindrical with a radius "r" m and a height "d" m. The track width is "a" m.



Topics to learn: Types of steering systems.

Task 5:

Aim: To design a vegetable cutter
Task: Design a vegetable chopper with 4 blades on its axle. The vegetable chopper should have a handle bar, which when rotated will cut the vegetables. The maximum volume of the vegetables it can cut should be *600ml*. The chopper should fit in a box of dimensions *9x9x9 cm.*
Topics to learn: motion analysis in CAD.

Submission Guidelines:

1. For the above tasks, you can either make a typed document or handwritten document in PDF format with sketches and steps mentioned clearly wherever needed.
2. For **Task 4,** you are free to use any programming software.
3. For **Task 5**, submit your solid models in .stl format, along with screenshot of the model taken from different angles. You can add a text file to explain about different features in your model.
4. Each file should have the following common prefix: ***Basic_M_<Roll_Number>***

Contacts:

Akhil Kasyap : 73308 17529
Ganishk D : 95668 53093
Snehkumar Sabhaya : 95105 50137

Machine Learning:

Task 1

Aim: To build a Convolutional Neural Network

Task:
Using the Fashion MNIST dataset, apply Convolutional Neural Networks (CNN) to classify the different fashion items. Build a model that learns from the training set and then predicts the labels for the testing set. Evaluate the model's performance by calculating accuracy and any other relevant evaluation metrics.
Describe the architecture of your CNN model, including the number and type of layers used, and any hyperparameters you tuned.

Note: You can build the model using any framework of your choice.

Concepts to Learn: NumPy, Pandas, ML Frameworks, Convolutional Neural network

Submission Instructions: Submit a well -documented code.

Task 2

Aim: To understand and implement classification using Support Vector Machines.

Task:
Support Vector Machine is an algorithm that is popularly used for classification problems. Explain the use of kernels in SVMs. Logistic regression is also commonly used for classification (refer Basic Tasks if you are not familiar). How do you think SVMs are different from logistic regression? Can the two algorithms be used interchangeably?
Given is a dataset that describes whether a given cell, when tested for cancer, is malignant or benign. Perform classification of data using Support Vector Machines. Implement the algorithm using Numpy only. Also visualize the hyperplane using Matplotlib.

Dataset: Link

Concepts to learn: Support Vector Machines, Numpy, Plotting using Matplotlib

Submission Instructions: Type/Write the answer and upload it as PDF. If the programming question is also attempted, submit a well-documented code.

Task 3

Aim: To understand the concepts of clustering in Unsupervised learning.

Task:

Cluster the following eight points (with (x, y) representing locations) into three clusters:
A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4,9)

Initial cluster centres are: A1(2, 10), A4(5, 8) and A7(1, 2).

The distance function between the two points a = (x1, y1) and b = (x2, y2) is defined as-

$$P(a, b) = |x2 – x1| + |y2 – y1|$$

Use K-Means Algorithm to find the three cluster centres after the second iteration.

Submission Instructions: Write down the solution with all the calculations up to 2 iterations in a piece of paper and upload it as PDF.

Task 4:

Aim: To realise the importance of dimensionality reduction in machine learning problems.

Task:

Real-world machine learning datasets often consist of high dimensional features. These features often correlate with one another and lead to heavy computational costs. To address these issues, dimensionality reduction is a popularly used machine learning technique. Principal Component Analysis and Linear Discriminant Analysis are two such techniques. What are the differences between the two techniques? Explain the situation in which you would use the two of them respectively. Given the dataset, reduce the dataset to 50% of its dimension using LDA. Link

Concepts to Learn: PCA, LDA, Dimensionality Reduction

Submission Instructions: Submit a well -documented code.

Task 5:

Aim: To understand how hyperparameters affect a model.

Hyperparameters are parameters that are not learned from the data but are set before the training process begins.

Task:

How do hyperparameters affect a model's performance in the following cases?

- Choice of learning rate in Linear regression
- Value of lambda in regularization
- Number of iterations during training
- Type of activation function in NN

Concepts to Learn: Hyperparameters.

Submission Instructions: Type down your answers in a doc and upload it as PDF.

Contacts:
Srivattsan S : 75199 81998
Anas Khalid : 73790 83468