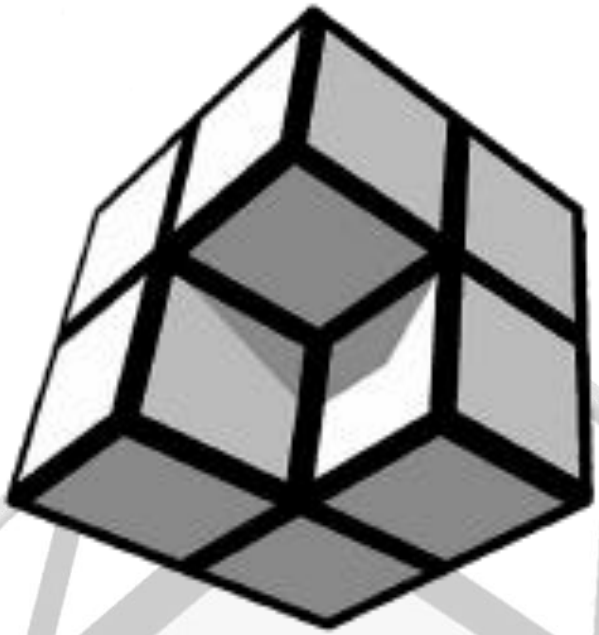


RMI

ROBOTICS AND MACHINE INTELLIGENCE

RMI
INDUCTIONS ’21

ADVANCED TASKS

Rules:

1. The tasks are split up into 5 Domains.
2. You can attempt any number of tasks from these Domains.
3. 3. The Domains are as follows:
   a. Mechanics
   b. Embedded Systems and Electronics
   c. Control
   d. Computer Vision
   e. Programming (Algorithms and AI)
4. All the tasks should be done individually.
5. Partial Completion of Tasks will also be accepted.
6. Maintain a small documentation of all the tasks you have done and take videos of the tasks you have completed. You may be asked to explain the working of the task using the video or told to explain using your setup.
7. You can use the microcontroller and programming language of your choice, except Embedded Systems and Electronics Domain, where you must use Embedded C as your
8. We highly recommend you read this entire document before starting the tasks.

Mechanics

Motion, forces, torques, kinematics, dynamics and stability. These are important considerations to be made when building a robot. Merely altering the mechanics of the various subsystems in our robot can open up a whole new world of possibilities and applications. What would we be if we had only our nervous system? The same goes for a robotics engineer without mechanics. The response of a robot to any command depends on the mechanical aspect of the robot, geometric constraints, force-torque relations etc. Mechanics gives the basic structure and provides a bridge between the electronics, intelligence, and the resultant physical effect of the robot. Below are the advanced problem statements for the mechanics domain, we hope to see a lot of creative and practical solutions.

1.
https://drive.google.com/file/d/1hcQVM6RvPPOoX6B8sPXdZHAsbohRAj2z/view?usp=sharing

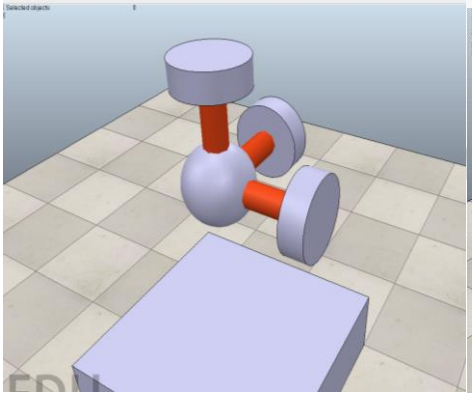


Fig-1

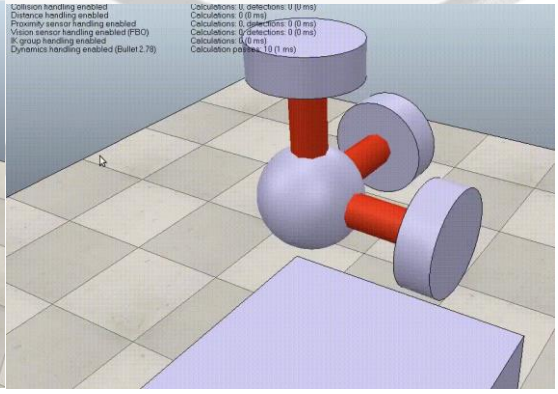
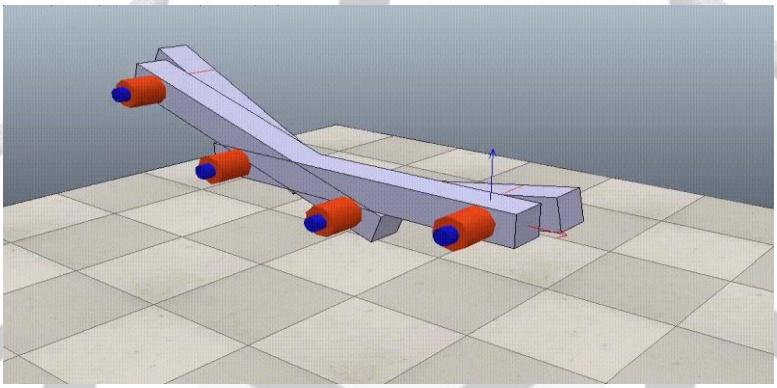


Fig-2

Consider the system given in the fig-1 with three links along three mutually perpendicular directions attached to a spherical joint at the centre. If the system is released, it oscillates about an equilibrium position. If a coordinate system {s} is attached to reference frame and a coordinate system {b} is attached to the equilibrium body frame, then find the angle(r,p,y) by which {s} has to be rotated to obtain {b}. [Follow right hand screw for assigning axis].

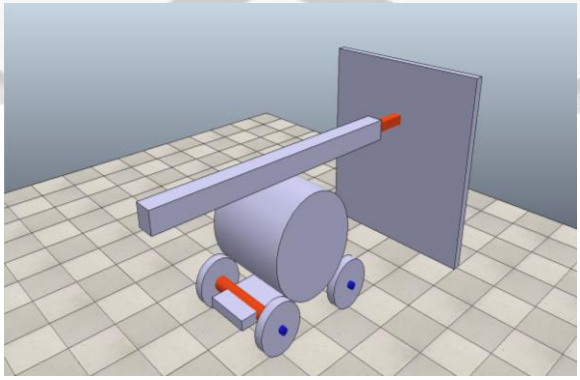
2. Inversion of a closed chain mechanism refers to fixing one link at a time to obtain different mechanisms. Try to simulate any two inversions of four bar mechanisms in the Coppelia Sim simulation environment. Try to simulate all inversions in a single scene. Record your screen for 10 seconds and submit it as a single mp4 file. (Example is shown below)
https://drive.google.com/file/d/1D4WIBgaG2V79uPvD_dWYBAtn_xTfXFiZ/view?usp=sharing



3. Consider the system given in the figure with four wheels of radius 10 cm and a larger wheel of radius 20 cm placed on top of the four wheels touching them tangentially. If a rod is used to impart force to the larger cylinder tangentially, then what will happen to the system if the rod moves forward with a velocity ‘v’.

a) Wheels move at velocity v.
b) Wheels move with velocity greater than v
c) Wheels move with velocity lesser than v

Distance between centres of any two adjacent wheels is 40 cm. Assume no slip condition between all surfaces in contact. Justify your answer with suitable expressions.



4. Matrices are very useful in Robotics. The first step in learning Robotic Manipulators is to understand linear algebra, particularly matrices and vectors. A matrix is basically a linear transformation that transforms a vector into another vector. Here we focus on two special types of matrices that are widely used in Robotics, namely Rotation matrices and Transformation matrices.
This task requires you to study and implement these matrices in a language such as **Python or MATLAB** (either is fine. You can also use any other language in which you can plot your results).

1. Let us take the 2-D case (xy plane). Your job is to formulate a rotation matrix that can rotate a vector by an angle of ""(input by user) about the z-axis. You are allowed to use packages such as Numpy (in Python). Plot your initial and final vectors accordingly. Once you are done with the 2-D rotation matrix, you are required to formulate a **single Transformation matrix** that can both rotate *and* translate a vector in a single operation. The angle of rotation and the displacements should be given as user input. Again, plot the results accordingly.
2. Your next job is implementing these matrices for the 3-D case. Here, there are three axes of rotation- x (roll),y (pitch) and z (yaw) axes. All these three angles should be given as user input. Formulate the matrix accordingly and plot the results as you did before. Once this is done, your next job is to again formulate a single Transformation matrix for both rotation and translation of a 3-D vector. The angles and the displacements are given as user input. Again, plot the initial and final vectors.

Concepts involved: Linear Algebra, Rotation and Transformation matrices, Numpy and Matplotlib (if you are using Python)

5. Make a solid model of the functional part of a hair trimmer with appropriate mates to constrain motion. The motor need not be modelled, but the rotor has to be. Make a small video displaying the mechanism in motion.

For any doubts, contact:
Aravindh – 9597690930; Jai Kesav - 8903509713

Electronics and embedded systems

All computers are only practical because of the constantly evolving field of electronics and embedded systems. The astounding and at the same time fickle world of resistors, capacitors, diodes, transistors etc always has opportunities for those who want to learn. From handling motors to all sorts of sensors, microcontrollers have served as revolutionary tools in prototyping and development of ideas. But have you ever wondered how this assembly of active and passive components actually works? Have you ever wondered what one means by a register in embedded

systems? Answers to these and a lot many questions lie in embedded C. A language designed specifically to code microcontrollers. It hangs somewhere between assembly language and C language bringing good of the both worlds together. A powerful tool when it comes to efficiency and understanding microcontrollers. With a little practice, grit and perseverance anyone can master this tool.
Given below is a set of problems specially designed to make electronics easy and fun to learn.
Note: All the coding involved in the upcoming tasks must be done in Embedded C. One can use Arduino IDE, Atmel Studio or any other coding environment that supports Embedded C.

Task 1:

Objective: Count and display the number of presses of a push button.
Details: Interface a push button and make an arrangement to count and display the number of times it has been pressed.
Concepts involved: Interrupts, UART, Button Debouncing.

Task 2:

Objective: Rotate two potentiometers to change frequency and duty cycle of pulse being generated on a PWM pin. Detect both parameters of the wave using External interrupts and display the values (frequency & duty cycle) on a screen.
Details: Generate a PWM pulse on any of the PWM pins but with a little twist. Make an arrangement so that you can set the frequency and duty cycle of the pulse using two potentiometers. One potentiometer must be used to set the frequency and the other must be used to set the duty cycle. Once done feed this pulse to any of the external interrupts pin and read and display the frequency and duty cycle of this pulse.
Concepts involved: Pulse Width Modulation (PWM), Timers, Interrupts, ADC, UART.

Task 3:

Objective: Establish I2C communication between two microcontrollers with the master being connected to an ultrasonic sensor and slave connected to a servo.
Details: Control the servo motor on I2C slave based on the reading of the ultrasonic sensor on the I2c master. Readings of the ultrasonic sensor must be mapped linearly to the servo. No external libraries should be used.
Concepts involved: I2C, Interrupts, Timers, PWM.

Task 4:

Objective: To build a Bluetooth controlled door system
Details:
You decide to build a Bluetooth controlled door system and take its simulation in Proteus as the first step for the same. The door is opened and closed using a motor where spinning the motor clockwise for 5 seconds opens the door and spinning anticlockwise for 5 seconds closes the door. Use appropriate motor drivers for driving the motor. For control, you decide to use the HC05 Bluetooth module. Let ‘a’ stand for open and ‘b’ stand for close. Hence, upon receiving a character

from the Bluetooth module, the microcontroller needs to actuate the motor accordingly. Use serial terminals inside the Proteus simulation software to mimic the serial communication through HC05 module. Make sure to set the frame parameters of the serial monitor equivalent to the HC05 module [Frame bits, parity, etc]
Use any Atmel AVR microcontroller for the same, programmed entirely in Embedded C.
Do not use any libraries for UART communication and timer delays.
Concepts involved: UART, HC05 Bluetooth module, Serial monitors in proteus, COM ports, UART implementation in Embedded C. Motor drivers, Timers.

References:

https://www.circuitbasics.com/basics-uart-communication/
https://www.electronicwings.com/avr-atmega/atmega1632-usart
https://en.wikipedia.org/wiki/AVR_microcontrollers
https://www.electronicwings.com/avr-atmega/inside
https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
https://www.theengineeringprojects.com/2016/03/bluetooth-library-for-proteus.html
https://www.theengineeringprojects.com/2013/05/how-to-use-virtual-terminal-in-proteus.html

For any doubts, contact:
Aswin – 9840535598; Satish – 9150146012; Girish - 8925720359

Control systems

What's control Theory, or "control" for that matter? When you are riding a bicycle, you are continually assessing what is going on through Kinaesthetic sense and keep on balancing it. What did you do there? You just controlled your bicycle by adjusting your posture for balance. Check out the following link containing video of a Stewart platform, to get an idea of what precisely a control system is.
Ball and Plate PID control with 6 DOF Stewart platform
The very objective of any control system in robotics is to help your robot stay in equilibrium against any sudden and unknown disturbance or sometimes make your robot stay in an unstable equilibrium for longer periods of time. Here's a video that introduces control theory to you…
Why Learn Control Theory
Before moving to the problem statement, you would need to brush up on the following:

1. Classic control theory: Open and Closed Loop Control Systems
2. Control Algorithms

Task 1:

Purpose:
Before going on to the next problem, let us discuss about control systems, controllers and their significance in robots. No matter what, all physical systems (like the robots) have an error in its output due to physical disturbances like friction, inertia, vibration, etc. Most of the time, we want the exact output without any errors for safety reasons, efficiency, etc. For example, if there is an error in the output of a tree cutting robot, it may cut something precious instead of cutting the tree. So, it is important to get the required output. To get the required output, we need to correct the error. To correct the error, we need to implement a system in the robot, called "control system". Controller is a part of the control system, which decides what kind of action should be taken in response to the error. There are many different types of controllers, for example, bang-bang controller, PID controller, fuzzy logic controllers, etc (Read about them if you have time). Most of the controllers we use in our club's robots are PID controllers. PID controllers can be designed mechanically, pneumatically, electronically or by programming in a processor. So, in this task, we want you to build a PID controller by programming a microcontroller.

Aim: Designing a PID controller.
Task: Assume there's a control system which has a sensor that gives values from 0 to 255. It has an actuator which when given an integer value of -1, it decreases the value of sensor, when given a value of +1, it increases the value of sensor and when given a value of 0, it does nothing. Your job is to write a C program with PID control algorithm to make the actuator maintain the sensor value at 128. The program should receive sensor value as input.
What to learn: PID Algorithm, C Programming.

Task 2:

Purpose: DC motors work well in robotics because they allow the robot to be battery powered, which offers great advantages for a variety of robotic applications, particularly mobile and collaborative robots.In addition to enabling the use of batteries to facilitate mobility, two key features that make DC motors great for robotics are speed variation and torque. Since it is also simpler to model it, it can also be used to teach control theory fundamentals. Microcontrollers are reliable instruments to control the speed of different size DC motors with very high precision.Most modern microcontrollers include built-in A/D and D/A converters(PWM). The use of standalone microcontrollers for speed control has gained ground. Hence through this task , we will learn how to accurately control the speed of a DC motor for high precision robotics application.

Aim: Speed control of DC motor by implementing a PI controller on arduino

Task: Choose a DC Motor model with inertia, loading and position encoder in proteus with the following parameters:
Voltage: 12v; Coil Resistance: 12 ohms; Coil inductance: 100mh; No load rpm: 360; Load/Max torque%: 1; Effective mass: 0.01; Pulses per Revolution: 32
Design a pid controller in arduino with appropriate kp,ki gains and sampling time so that the motor achieves a speed of 100 rpm with a maximum overshoot of 5%. Establish a virtual COM port in proteus, send the speed and time data through it. Write a python script to receive that data and plot the speed vs time graph. Observe the overshoot and damping characteristics and tune your pid gains appropriately. (Motor must reach its steady state speed with minimum overshoot and time)

What to learn: PI Algorithm, Arduino programming

For any doubts, contact:
Rishivanthiya – 8300073420; Satish – 9150146012

Computer vision

1) **Aim:** Tic tac toe has always been an evergreen game but quite boring. So we want you to design tic tac toe with an interesting twist.
   Task: The play screen should contain a basic layout of a tic tac toe board (white canvas, two black horizontal lines and two black vertical lines) using OpenCV and the current state of the game with X and O placed in their respective grids. You should read the position of colored objects from your webcam to determine position in which to place your move.
   Things to learn: Masking, Drawing functions

2) **Aim:** Virtual reality has been growing these days. The biggest success in virtual reality has been the game Pokémon Go. Design a virtual video display using OpenCV.
   Task: In this task you need to design a virtual video displayer which displays a given video on a target image present on your live feed. The video should overlay only on the image even if you move the image around.
   Things to learn: Feature detection, Homography

3) **Aim:** Hand cricket has been a wonderful part of childhood. Create a hand cricket game using OpenCV.
   Task: Make a two-player hand cricket game environment (use numbers from 1 to 5) using OpenCV that displays scores, player turn and should end the game once a player wins.

Things To learn: Contours, Gesture detection

4) **Aim:** Tracking and prediction of path has always been a great part of robotics and it can be easily achieved through image processing. Track the path of a ball using OpenCV.
Task: Find a way to track and **predict the trajectory** of the blue ball in the video: https://drive.google.com/file/d/1fRHcsv6K4iR09nkTTz9DHP0VjoE7jMGP/view?usp=sharing

Also generate a compositing image like the one given below

Things to learn: Kalman Filtering, Image Compositing

For any doubts, contact:
Divya – 9150006456; Sivvani - 9655192435
Lokesh – 7598002344; Shrushti - 83297 51332

Machine Learning:

Task - 1:

Image Classification

Aim: Build a Convolutional Neural Network for Classification of Images.
Task: The **CNN** model must be built only using **PyTorch**. No other deep learning libraries are allowed. You can use OpenCV or Matplotlib for Data Preprocessing and Data Visualization. Your aim is to classify the given dataset into 2 classes, i.e., dogs and cats.

The model must have **at least** 2 Conv Layers, each followed by a Pooling Layer (of your choice) and 1 Fully Connected Layer (Excluding Output Layer).

Dataset description: Given is a zip folder of train and test dataset. Both folders contain images of cats and dogs. Train your model using the train dataset and test your model for predicting the classes for the test dataset.
Label each image in the test dataset, based on the classes they belong to and submit it in a csv file.

Dataset: Link
Evaluation Metrics:

- Accuracy of prediction and other metrics which you feel is necessary to portray your model efficiency.
- Efficient implementation of network.
- Code quality – use proper classes and functions, as necessary.

Submission Guidelines:
****** We encourage you to use **Google Colab** for training and testing your model.
Submit a folder with the following files:

- **Normal** – A folder containing normal task code- ipynb file.
- A csv file containing the output data in a new column, corresponding to the index value of test data name of each file.

What to learn: Programming with PyTorch, Convolutional Neural networks, Object classification, Hyper parameter tuning.

Task - 2:

Single Object Detection

Aim: Build a Convolutional Neural Network for a Single Object Detection (Object Classification + Localization) in images.

Task: The **CNN** model must be built only using **PyTorch**. No other deep learning libraries are allowed. You can use OpenCV or Matplotlib for Data Preprocessing and Data Visualization. Your aim is to build a model capable of classifying the image into 2 classes "Person" or "Non-Person" and finding a single bounding box of the major object present in the image. The output layer must contain 5 nodes (one for classification and the rest for the bounding box prediction).

You could either build a neural architecture or utilize an existing architecture for solving this task.

Dataset Description:
Given is a zip folder containing a folder named dataset with images in it and 2 text files train.txt and test.txt with each line containing data in the following format:

"002500.jpg 62 0 278 471 0"

Here 002500.jpg is the image, 62 0 278 471 represents bounding box parameters of the object in the image (x, y, w, h), in this (x, y) – top left corner coordinates of bounding box and (w,h) – width and height of the bounding box, 0 -corresponds to class it belongs to (1 – person class, 0 – Non-person class).

002500.jpg 62 0 278 471 0" – image, x, y, w, h, class
"Each image contains only one bounding box".

Dataset: Link

Evaluation Metrics:

- Accuracy of prediction of bounding boxes
- Efficient implementation of network.
- Code quality – use proper classes and functions as necessary.

General instruction:

- We encourage you to use **Google Colab** for training and testing your model.

Submission Guidelines:

Submit a folder with the following files:

- Normal – A folder containing normal task code- ipynb file.
- Output folder of test images containing the predicted bounding boxes.

What to learn: Programming with PyTorch, Convolutional Neural networks, Object Detection, Hyper Parameter Tuning.

Bonus Task:

Aim: Same as task 1.

Task: In Task 1, use **NumPy to** build the model, instead of using **PyTorch** or any other deep learning library. You can use OpenCV or Matplotlib for Data Preprocessing and Data Visualisation. Rest of the instructions are the same.

Evaluation Metrics:

- Accuracy of prediction and other metrics which you feel is necessary to portray your model efficiency.
- Efficient Implementation of network.
- Code quality – use proper classes and functions, as necessary.

General instruction:

- We encourage you to use **Google Colab** for training and testing your model.

Submission Guidelines:

Submit a folder with the following files:

- Normal – A folder containing normal task code- ipynb file.
- A csv file containing the output data in a new column, corresponding to the index value of test data name of each file.

What to learn: Programming with NumPy, Convolutional Neural networks, Object Classification, Hyper Parameter Tuning.

For any doubts, contact:

Rigved – 8879672720; Greeshwar - 9344066340