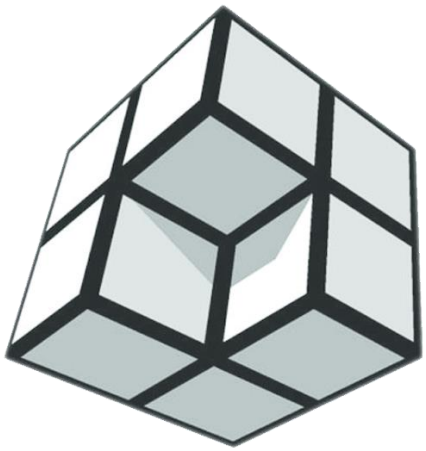


RMI

ROBOTICS AND MACHINE INTELLIGENCE

INDUCTIONS – 2019

ADVANCED TASKS

Rules:

1. The tasks are split up into 5 Domains.
2. You can attempt any number of tasks from these Domains.
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 also 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 programming language.
8. Bonus Tasks are only to be attempted if all the other tasks in that domain are completed.
9. We highly recommend you to read this entire document before starting the tasks.

Failure is the symptom of learning something new, feel free to switch domains and explore.

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 have 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.

Here's a video of Atlas which will give you an idea of the paramount importance of mechanics to a robot.

https://youtu.be/hSjKoEva5bg

Below are the advanced problem statements for the mechanics domain, we hope to see a lot of creative and practical solutions.

Task 1:

Objective: Fabricate Or design a 2-axis planar (2R) manipulator.

Details: Fabricate Or design a 2-axis planar (2R) manipulator using necessary actuators and appropriate materials which can be used for further tasks. (Dimension is left with your choice!)

What to study? Different types of manipulators and types of actuators -servo selection.

Task 2:

Objective: Implement Inverse kinematics to your mechanisms.

Details: With the help of microcontroller, program your mechanism to move the end effector to a particular position in workspace of manipulator.

What to study? Inverse Kinematics and interfacingg servos with microcontroller.

Task 3:

Objective: Make an appropriate gripper as an end effector.

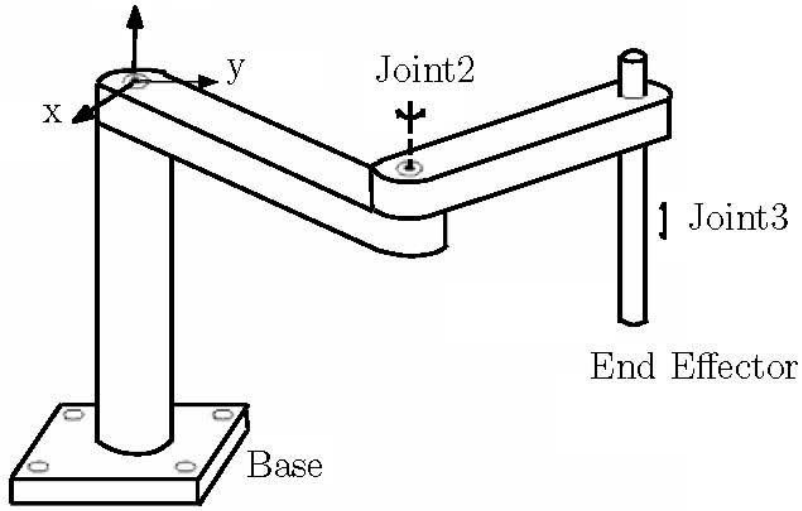
Details: The end effector can be made of appropriate material and dimensions of any suitable mechanisms to pick up and move any object **without deforming** it.

What to study? Types of grippers used for various scenarios.

Task 4:

Objective: Improvise the above manipulator(2R) into SCARA (2R1P) by adding an actuation to the end effector.

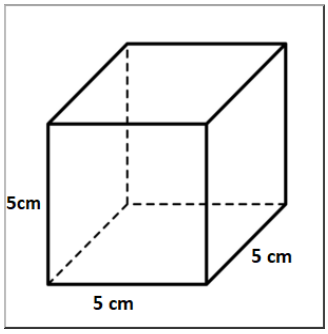
Details: 2R is fixed at some suitable height and the end effector is given a linear actuation by any suitable mechanism. Along with the gripper from previous task, use this manipulator to move from one position to another within the workspace.

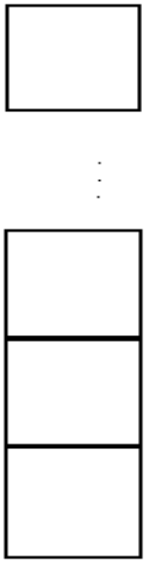


Bonus Task:

Objective: Using the above SCARA manipulator create a pick and place robot to arrange smaller cubic boxes in a stack arrangement.

Details: The objective is to pick some small cube boxes of dimensions **5x5x5 cm**, made of paper or cardboard (or any other suitable material) from known location within the workspace and arrange it in a linear stack arrangement within the workspace as shown. (The number of smaller cubic boxes depends upon the size of the manipulator)





Embedded Systems and Electronics

All computers are only practical because of 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 microcontroller has served as revolutionary tool 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? Answer to these and a lot many questions lies 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 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.

Estimated time: 3 days

Task 2:

Objective: Rotate two Potentiometers and print corresponding voltages on Serial monitor.

Details: Take input from two potentiometers and do the necessary to process and print the voltage across their individual wipers and ground on the serial monitor.

Concepts involved: Analog to Digital Convertor (ADC), UART.

Estimated time: 2 days

Task 3:

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.

Estimated time: 4 days

Task 4:

Objective: By using a MOSFET as switch, and applying a PWM pulse from a microcontroller control the speed of a motor (encoder feedback not needed). The required output duty cycle must be received from the Serial monitor. The output to the motor must resemble a "Pure Analog Wave" and not a PWM wave.

Details: The above arrangement is to be made to rotate the motor in one direction (Both directions are not needed). Use of a motor driver is prohibited.

Concepts involved: PWM, UART, Low Pass Filter.

Estimated time: 3 days

Tutorials: www.electronics-tutorials.ws/filter/filter_2.html

Task 5:

Objective: Fabricate a boost converter (Perf board soldering must be done), which functions by receiving PWM pulse from your microcontroller. Take 2 series Li-ion /Li-Po cells (rated at 7.4V), amplify its voltage to 12V & run a motor with the output.

Details: A boost converter is DC-to-DC power converter that steps up voltage on the cost of current on the output. Fabrication of one such circuit is to be done and that circuit has to be given input voltage using the sources given above and the amplified output is to be used to run a motor.

Concepts involved: Electronics, PWM, ADC.

Estimated time: 4 days

Bonus Task:

Objective: Fabricate a buck & a boost converter (soldering the connections is not compulsory), both receiving PWM from microcontroller, both taking input from 2 Li-ion/Li-Po (rated at 7.4V). Make the buck converter step down the voltage to 5V and self-power the microcontroller through the 5V power pin (a temporary power supply can be given to the microcontroller at the start for the program to start running). Connect a potentiometer to the Arduino & depending on its position, vary the output voltage of the Boost converter proportionally from 9V to 14V, and run a motor also print the output voltage of the Boost converter on the Serial monitor.

Concepts involved: Electronics, PWM, ADC, UART.

Estimated time: 5 days.

Control

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) Interfacing IR sensors and Ultrasonic sensors with a Microcontroller of your choice
3) Control Algorithms

Task 1:

Objective: Make a planar ultrasound sensor radar.

Details: Make a planar yaw mechanism with an ultrasound mounted. The mechanism must locate an object and find the x and y coordinates of the object's position with respect to the location of the radar.

Note: Your radar must be mounted on a differential drive robot later. So, choose the size of the radar accordingly.

What to study? Interfacing servo and ultrasound sensor.

Task 2:

Objective: Fabricate a planar yaw or pan mechanism that faces a fixed direction irrespective of the base direction.

Details: Fabricate a platform over a base capable of yaw motion. The platform must be focussing one direction and when the base is rotated, the platform must counter-rotate to focus to the original direction. You can use an IMU to sense the angle rotated and a servo or a DC motor with encoder.

Note: Your radar must be mounted on a differential drive robot later. So, choose a proper size for the yaw mechanism. You can transform the radar into this yaw mechanism as well by attaching an IMU. You can modify the code to make it a radar or a directional focussing yaw mechanism.

What to study? Interfacing servo using IMU feedback.

Task 3:

Objective: Fabricate a differential drive robot with the yaw mechanism.

Details: Fabricate a differential drive robot that is controlled from a laptop or mobile wirelessly. Mount the yaw mechanism to the robot and as the robot is moved by manual control, it must remain focussed to a fixed object. You can use any transmission method like Bluetooth or Wi-Fi.

What to study? Differential drive robot, communication and data transmission.

Task 4:

Objective: Wall following robot.

Details: Attach an ultrasound sensor to the differential drive robot to convert it to a wall follower so that it maintains a fixed distance from the wall while moving using feedback from the ultrasound sensor. Begin the implementation with a proportional controller (P), then extend it to PI, PD or PID controller.

What to study? PID control algorithm.

Task 5:

Objective: Find and revolve around an object.

Details: Mount the radar on the robot. Find an object using the radar fabricated and then revolve around the object maintaining a constant distance with the ultrasound facing the object always.

Note: You can use a cylindrical object with a suitable size for this task to make the revolving smoother and easier.

Bonus Task:

Objective: Map the wall follower and the object with reference to the initial coordinates.

Details: Find the x coordinate, y coordinate and angle with respect to the initial orientation of the robot at every instant and print it on a monitor screen. Simultaneously the yaw mechanism must face the object and print its position on the screen. You can use an IR tachometer (wheel encoder) for this purpose.

What to study? Odometry, interfacing wheel encoders.

Some commonly used sensors:

HCSR04 (Ultrasound), MPU6050 (IMU), HC05 (Bluetooth)

Computer Vision

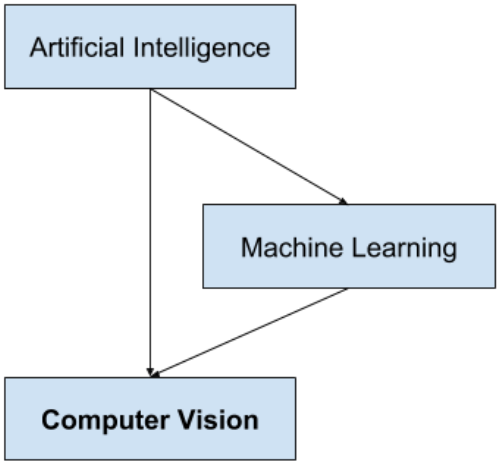
"If We Want Machines to Think, We Need to Teach Them to See"

- Fei-Fei Li

Computer Vision, often abbreviated as CV, is defined as a field of study that seeks to develop techniques to help computers "see" and understand the content of digital images such as photographs and videos.

The problem of computer vision appears simple because it is trivially solved by people, even very young children. Nevertheless, it largely remains an unsolved problem based both on the limited understanding of biological vision and because of the complexity of visual perception in a dynamic and nearly infinitely varying physical world.

It is a multidisciplinary field that could broadly be called a subfield of artificial intelligence and machine learning, which may involve the use of specialized methods and make use of general learning algorithms.



Task 1:

Objective: To learn interfacing of trackbar

Details: With top left corner as origin vary the x and y coordinates of a point. Use three track bars for this task, two for varying the x and y coordinates and one for varying the thickness of the line joining the origin and x and y coordinates.

Task 2:

Objective: To learn mouse call back functions and interfacing of keyboard inputs

Details: Make a version of mini paint with the following features

i) On pressing R, one should be able to draw a rectangle using mouse buttons.

ii) On pressing C, one should be able to draw a circle using mouse buttons.

iii) On pressing L, one should be able to draw a line using mouse buttons.

iv) S button should save the image at a location.

v) E button should clear the screen

It depends upon the user how he wants to use the mouse buttons and how many mouse buttons he wants to use for drawing a single figure. Lesser the number better it is.

Task 3:

Objective: To learn object detection

Details: -

i)Detect objects of three colours Red, blue and green. It can be of any shape as decided by you. Print on the display window the colour of the object detected.

ii)Determine the distance of the object from the screen and the velocity at which it is moving. Print the distance and velocity on the display window.

iii) Whenever you an object is detected it should write on notepad

Example output:

GREEN 1 (when green is detected)

BLUE 1

GREEN 2

RED 1

...

...

RED 10

RED is the winner

The colours can be shown in any random order. When the count for any colour reaches 10

It should write on notepad that colour is the winner.

Bonus Task:

Objective: To learn path prediction

Details: Download the video given in the link. You can see that it's possible to track its path when its visible but when that is not the case when it goes under the card board box. This phenomenon is known as occlusion and is one of the most researched topics by computer vision engineers. Now your job is to devise a method to predict the path of the ball under the cardboard box.

Programming

This Domain consists of 2 Sub-Domains. They are:

(i) Algorithms
(ii) Artificial Intelligence

Algorithms:

An algorithm is a sequence of steps to solve a given problem. Multiple algorithms can exist to solve the same problem, but may have different running times and memory usage.

Why does a robot enthusiast need knowledge of implementing and analysing algorithms?

This is needed to be able to choose the algorithm best suited for your purpose. This purpose could be minimum memory space of your raspberry pi or high decision-making speed requirements of your mobile robot. So here are a few problems that deal with the very basics of algorithms.

Task 1:

There are n cities and the cost of travelling between any pair of cities is known (Note: The cost of travelling from city A to city B, need not be the same as that from city B to city A). You are given a starting point and need to visit every city once and return to your starting point. Find the minimum cost it would take you to achieve this.

Assume: The inputs are number of cities and distance between each of them and starting point. The output should be the minimum cost/distance to visit each city from starting point and return there and the path that allows this cost.

Example:

N=3

A-B = 10

B-C = 12

A-C = 5

C-A = 8

C-B = 13

B-A = 10

Starting point: A

Output: Minimum cost is 28. Path A-C-B-A

Note: Like the given example you are expected to give a suitable input with higher value of n and get the corresponding result.

Things to Study: Graph Theory.

Task 2:

Given a grid of order RxC and an initial position, find the minimum distance from the source to any corner of the grid. A move can be made to a cell grid[i][j] only if grid[i][j] =0 and only left, right, up and down movements are permitted. If no valid path exists print -1.

The end of a grid is any cell present in first or last row and first or last column.

Example:

a) Input: i = 1, j = 1, grid[][] = { {1, 0, 1}, {0, 0, 0}, {1, 1, 0} }
Output: 2

b) Input: i = 0, j = 0, grid[][] = { {0, 1}, {1, 1} }
Output: 0

Note: Like the given examples you are expected to give a suitable input with higher value of r and c, and get the corresponding result.

Artificial Intelligence:

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision. Let's get started with AI, which is going to be the brain behind our robots.

Task 1:

Assume n people are playing a game with the following rules:

1) Each person proposes a strategy to split 100 coins among themselves.
2) Once the strategy is proposed, everyone vote on it.
3) If the strategy has majority or equal upvotes then it is adopted
4) Otherwise the person is kicked out of the game.

The order of proposal is in increasing order of n. Assume n is less than 100. Assume the computer is the first to propose a strategy. The following are the priorities of each player:

1) Stay in the game
2) Having ensured staying alive, maximise the reward for themselves
3) Kick other players out of the game given the first two priorities are met. This means given that in a particular situation a player can upvote or downvote and both result in the same outcome, the player will downvote.

What strategy should the computer propose to achieve its priorities? For a given n what is the number of coins the computer wins?

Drive link for datasets:
https://drive.google.com/open?id=1JH7G9vrBcQMuPM2CeqhDLMw5WCt_RAle

Task 2:

The folder TASK 1 in the drive link provided contains a training set, test set and data-description for predicting housing prices.

Go through the data description and train on the data provided in the training set and find the performance on the test set. The task is to predict housing prices given features of the house.

Try to accomplish as much from 1 to 6:

1) Data cleaning: Clean the data, remove outliers and first of all determine which features you think are necessary.
2) Use simple linear regression.
3) Decide on the cost function you are going to use. (Remember you are trying to predict continuous values)
4) Decide on an appropriate performance metric for the test set. This can be simple accuracy, percentage error or F1 scores, recall, etc
5) Some ideas may be obtained from plotting the data.
6) Use regularisation in training. Compare the performance on test set with that produced without regularisation.

Task 3:

(Feel free to use any packages such as TensorFlow, Keras, etc)

Using the same data provided, build an artificial neural network for achieving the above task of house price prediction.

Bonus in task 2: Try using dropout regularisation.

Bonus Task:

- ➔ Implement a simple convolutional neural network. Feel free to use any packages/wrappers. Try to change padding, filters, strides, architecture pipeline (flow of vector shapes from input to output).
- ➔ A lot of image datasets are available online. Simplest being MNIST Dataset. Try your network architecture on any such dataset.
- ➔ Change the dimensions to LeNet 5 Architecture. Compare performance.

Tutorial Links

1) Mechanics:

Basic Mechanics:

https://www.societyofrobots.com/mechanics_gears.shtml

https://www.societyofrobots.com/actuators.shtml

Forward and Inverse Kinematics:

Saeed B. Niku – Introduction to robotics: Analysis, Systems, Applications
John J. Craig – Introduction to robotics: Mechanics and Control
https://nptel.ac.in/courses/112101098/15

2) Embedded Systems and Electronics:

For Embedded C- https://www.youtube.com/playlist?list=PLA6BB228B08B03EDD
Also go through other videos of this channel for specific concepts.
ATmega328p Datasheet- Click here to Download
For Buck and Boost converter- https://www.elprocus.com/buck-boost-converter-circuit-theoryworking-applications/
You can also go through other sources if you don't find this sufficient.

3) Control:

Control Theory:
https://www.youtube.com/watch?v=oBc_BHxw78s&list=PLUMWjy5jgHK1NC52DXXrriwihVrYZKqjk

PID Algorithm: https://www.instructables.com/id/Control-Speed-Motor-PID/

Video Lectures by professors from foreign universities on sites like edX and Coursera.

4) Computer Vision:

Regarding reading material, you can refer this book in the link. It should be more than enough for solving the first three tasks.

Video Tutorials:

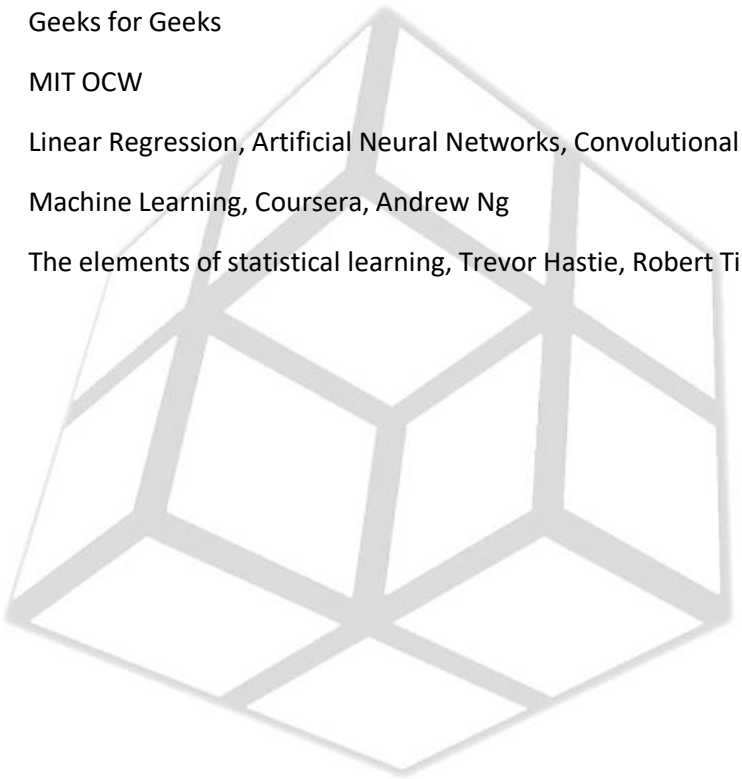
1) Sentdex
2) Pysource

Blogs:

1) PyImageSearch
2) LearnOpenCV

5) Programming:

1) For algorithms:
    - Graph Theory: Path planning, Traveling salesman problem
    - Geeks for geeks
    - Code Monk
    - Khan academy
    - Introduction to Algorithms -Charles.E. Leiserson, Clifford Stein, Ronald Rivest, Thomas.H. Cormen
2) For Machine Learning and AI
    - Geeks for Geeks
    - MIT OCW
    - Linear Regression, Artificial Neural Networks, Convolutional Neural Networks.
    - Machine Learning, Coursera, Andrew Ng
    - The elements of statistical learning, Trevor Hastie, Robert Tibshirani



People to Contact for Doubts on Tasks:

Mechanics:

Athithya (9442069879), Jinesh (9486461891)

Embedded Systems and Electronics:

Deepak (8870711551), Koushik (9003478945)

Control:

Nitish (7200790709), Mukesh (9445607612), Aquif (9500708264)

Computer Vision:

Sarthak (9439906947), Harshith (9445576198), Avinash (9790320049)

Programming:

Pranav (9962407645), Dharun (8555055334), Abdur (9123449175)

Apart from this you can post your doubts in RMI Freshers Forum, on Facebook.

(or)

In Genesis'18 WhatsApp Group - https://chat.whatsapp.com/22XE6qwjtdEAt22xXWhHCD