Mid Sem Exam
ME 639 - Introduction to Robotics
IIT Gandhinagar

Assigned: 30 September, 2021 9pm
Due: 11:59pm on Friday, **1st October, 2021**

Submission on GitHub: Submit by raising a pull request to the branch having your name on the repository harishpmiitgn/iitgn-robotics. Your files will be considered submitted only after you initiate the pull request. In case of any technical difficulties, email all the files to the TA and the Instructor by **11:50pm**. No email or gitHub submissions after 12:00 midnight will be accepted.

Submission Format: You are encouraged to submit all word and hand answers in a single PDF file and all codes and their output in a single Jupyter notebook file, colab file or Readme.md file. However, if you are not comfortable with any of these formats, you can choose any other convenient format.

Collaboration Policy: Discussion with all classmates (including phone calls and WhatsApp messages) and TA are permitted, however no exchange of (or showing) equations, or lines or sections of code are permitted. Two people cannot work on the exam in the same room.

Open-book Exam: Textbook, Notes, Previous Assignments, GitHub Repository, Online Resources are all permitted and may be referred to. Cite all work used from repositories.

Important Grading Policy:

Level 0 - Grades: Only **Tasks 5-9** are graded, the rest of the tasks have no bearing on your MidSem Exam grade!

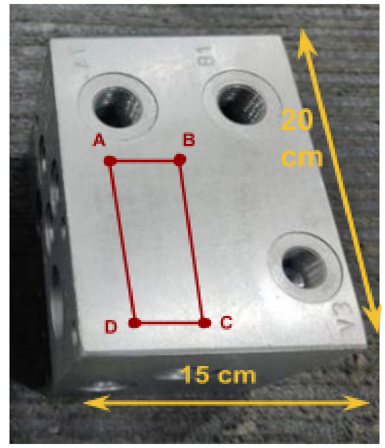
Level 1 - No EndSem: Reasonably good answers/solutions (even if they are not perfect) to **Tasks 1-a, 1-b, 1-c, 2-a, 2-b, 3-a, and 4-a** earns you a '**No EndSem Pass**'. In such a case, you need not give the EndSem exam nor will you lose any grade for that.

Level 2 - No Quiz 2: Reasonably decent answers/solutions (even if not perfect) to **Tasks 1-e, 1-f, and 4-b** earns a '**No Quiz 2 Pass**'. In such a case, you need not give the Quiz 2 nor will you lose any grade for that.

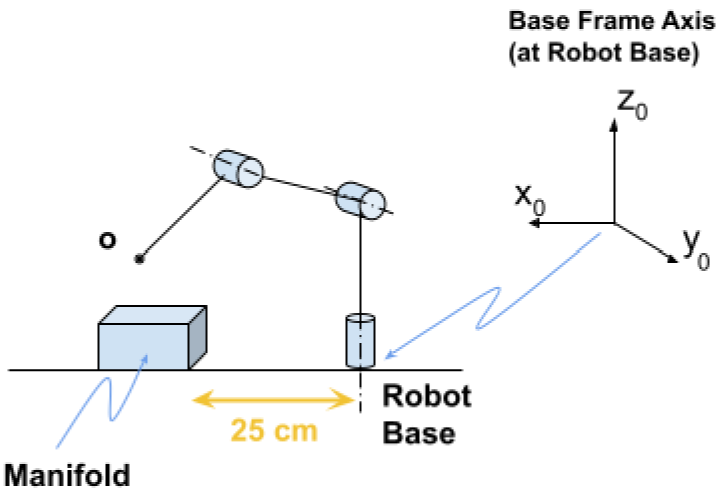
Level 3 - Freakin' Awesome & Superhero Mission: The rest of the tasks are to only prove that you are freakin' awesome and you can do them all. However, as a bonus, a reasonably 'solid'_*attempt* (even if answers are incorrect) of **Tasks 1-d, 3-b, and 4-c** in addition to earning the 'No EndSem Pass' and 'No Quiz 2 Pass' earns you an extra '**Trump Card**' that you can use to seek an automatic full score in any future or past assignment.
In addition to completing all of the above, if you are able to also submit a decent attempt for Task 3-c, then you are a **Superhero** (no impact on grade).

Tasks:

1. Consider the Milacron manifold block inspection project in which surface roughness on the top surface of the manifold shown in the image is to be checked by a robot. The following subtasks are to develop and execute a trajectory to be followed by an inspection robot such that the robot makes contact with the top surface of the manifold at point A and then traces out a path A-B-C-D-A while in contact with the surface.



   Let us assume that the manifold block is 20 cm by 15 cm with 10cm height (as shown in the image). As illustrated in the schematic below showing a robot (with an PUMA-style RRR configuration), the manifold block is situated 25 cm away from the robot base and is on the same horizontal surface as the robot. The base frame axis ($o_0x_0y_0z_0$) directions are indicated in the schematic (shown separately for simplicity). Point **o** shown in the figure is the wrist centre. The manifold block's longest side is aligned with the x_0 direction in the base frame and the shorter side is aligned with the y_0 direction in the base frame.



   Please note that the maximum possible range of the (x,y,z) coordinates (in metres) of points A, B, C and D are (0.45, 0.075, 0.1), (0.45, -0.075, 0.1), (0.25, -0.075, 0.1), and (0.25, 0.075, 0.1) respectively, as these are the coordinates of the corners of the top surface of the manifold block. Also note that the z-coordinate on the top surface of the manifold block is always 0.1.

   We consider three robot types, Robot 1 - Stanford-type (RRP), Robot - 2, PUMA-type (RRR), and Robot 2 - SCARA-type (RRP), for the following tasks. Assume all link lengths in all three robot types to be 0.25m and masses of all links to be 0.8kg and moment of inertia to be 0.005 kg-m^2. Feel free to either assume any other values that may be needed or change the values of the above parameters if necessary (stating any such assumptions or changes clearly in your solution file). For these first set of tasks, assume only three links (no wrist). All tasks below are to be performed <u>for all three robot types</u>.

a. Verify for all three robot types that the entire top surface of the manifold lies within the robot workspace by ensuring that all four corners lie within its workspace, by using inverse kinematics to compute joint parameters and then substituting these joint parameters in the forward kinematics to ensure that the same coordinates are obtained back. Repeat this for each robot and for each corner. If any of the points are outside the workspace, accordingly change the link lengths (and state the new link lengths) to ensure they come within the workspace. Submit both the code and the output from the code showing computed joint variables and the verification.

b. Assemble a single code that will take as a user input the values of desired coordinates of points A, B, C and D and work out the corresponding joint variables and then substitute back in the corresponding forward kinematics code to determine back the end-tip coordinates and output them. You may either have one code that incorporates all three types of robots (preferred), or have one code for each robot type (less preferred). Submit both the code and the output from the code showing computed joint variables and the verification (with your own choice of values for points A, B, C and D).

c. Choose A = (0.40, 0.06, 0.1), B = (0.40, 0.01, 0.1), C = (0.35, 0.01, 0.1), and D = (0.35, 0.06, 0.1) and repeat the above task. These are coordinates to trace out a 5cm X 5cm square. Submit the output of the code showing computed joint variables and the verification for all three robots.

d. Assume you have the control systems in place to achieve the desired joint variable values (joint angles or joint extensions). Work out how to compute desired joint velocities to go from one point to the next in the above sequence with a constant velocity of 0.01 m/s (submit hand-worked out solution showing the key steps). Code the solution and verify the solution by printing out both the joint variables and velocities and the end-tip coordinates and velocities at 1cm intervals. That is, when traveling from A to B, print the above-said values at points (0.40, 0.05, 0.1), (0.40, 0.04, 0.1), (0.40, 0.03, 0.1), and (0.40, 0.02, 0.1) and likewise for the other three paths. Also, even though cartesian velocity is constant along any one line of the path, are the joint velocities expected to be constant, please comment on why or why not. <u>Dynamic simulation is not expected in this task</u>, only kinematic computations. Submit the PDF of the worked out solution, the code, and the output of the code showing computed joint variables and the verification for all three robots.

Finally, assume an additional wrist with the joint angles as Euler angles as discussed in class/Assignment 4. Assume the tool point is 5cm away from the wrist centre o (this distance is d_6 in the textbook notation).

e. Let us assume that the surface roughness tool held at the end of the robot has to always be perpendicular to the surface (pointing in the negative z_0 direction) for you to record accurate roughness readings. Hence for the four coordinates used in Task 1-c. above, compute the fresh joint variables for all three robots taking into account the new 5cm distance and orientation requirements. Submit the code and the output of the code showing the joint variables and final verification.

f. Compute the wrist angles for each point (used in Task 1-c.) and each robot to ensure that the tool is always held perpendicular to the manifold block top surface (pointing in the negative z_0 direction). Submit the code and the output of the code showing the joint variables and final verification.

2. Consider the pill picking robot desired by Timetooth Technologies. Their requirement is to have a preferably one-DOF gripper in a compact form factor (since it has to fit within a small cup) and be able to pick up a single pill lying in the cup in any orientation. A vacuum-based gripper is not desired.

   a. Comment on whether you think a hard gripper or a compliant/soft gripper will be more suitable for this task. Please include 4-6 sentence rationale for your preference. Even if you do not have a very precise reason, you could still include hypotheses in your arguments. Submit your answer in a PDF file format .

   b. Briefly review 'flexible mechanisms', 'soft robotic grippers', 'universal grippers', 'paper grippers' and 'origami robots' from information available on the web and comment on whether any of the ideas you explored seem suitable for this application. Include weblinks (either videos or papers) of any specific grippers that you felt might be worth considering and may serve as a good reference. Submit your answer in Word/ text (less preferred) format or in PDF file format (with preferably clickable links).

3. Consider the exoskeleton scruffing problem from TimeTooth Technologies. Consider only one side of the exoskeleton and consider it to be a planar 2R elbow manipulator.

   a. First, pick reasonable link lengths by searching for (or measuring your own) hip-to-knee distance as the first link and knee-to-ankle distance as the second link and state these values. Also read about gait trajectory, step height, and step length and explain what they are in 1-2 sentences each. Submit this in PDF format.

   b. Given the attached CSV file containing (x,y) values of the ankle coordinates for a typical gait cycle, figure out using online resources how to import/read this file (Hint: pip install xlrd). Then import the file and plot all the points on the x-y plane showing the ankle coordinates over the entire gait cycle. Submit code and the results from the code including the plot.

   c. Finally, work out the joint variables using inverse kinematics for all the provided (x,y) points and verify that the answer is correct by substituting these values in forward kinematics of 2R elbow manipulator and verifying the result. Adjust link lengths if needed to expand the workspace. Submit a code and the outputs of the code showing verification of the results.

4. Let us consider a simplified version of the Series-Elastic-Actuator project from ISRO and the Joint Impedance Control from Timetooth Technologies. Both these projects have somewhat similar requirements, let us consider a single link robot without any compliance for simplicity (final project objectives will require us to include compliance).

   a. For a robot with a single revolute joint and a single link of length l, work out the DH Parameters.

   b. For the joint to behave like a virtual torsional stiffness with linear characteristics (refer to MiniProject Task 3 for a general idea), what is the desired torque (neglecting gravity) to be provided by the motor connected at the joint (Hint: The torque will depend on the joint angle).

The dynamics of a single-link robot (with gravity) can be worked out to be $ml^2\frac{d^2q_1}{dt^2} + mgl\,sin(q_1) = \tau$ with mass m, length l and joint torque τ.

   c. Set up a dynamic simulation in python and substitute $\tau = mgl\,sin(q_1) + \tau_i$ where τ_i is the torque worked out in Task 4-b. Start the simulation with a non-zero initial condition for joint angle and see if the link behaves in a manner that makes the joint look like a virtual torsional spring (you should see a steady oscillating motion similar to a spring mass motion). Also verify if the total energy of the system ($mgl\,cos(q_1) + \frac{1}{6}ml^2\,(\frac{dq_1}{dt})^2$) is constant. Submit your codes and resulting plots and verifications from the code.

Submit answers to the following questions in PDF file format (short answers only, and in some cases only yes/no answers).

5. In the DH convention are all joint axes always aligned with respective z axis?

6. In the DH convention, are the origins of all the coordinate frames always at the centres of the joints?

7. Is it true that a homogeneous transformation consists of both a rotation and a translation?

8. For a sequence of rotations performed one after the other, can the rotation matrices for each individual rotation be multiplied together to form the overall rotation matrix (capturing the sequence of rotations)?

9. Is a composite rotation matrix consisting of a sequence of several rotations still an orthogonal matrix with determinant equal to 1?