MPHY0054 Robotic Systems Engineering
Coursework 2: Jacobian, Inverse Kinematics and Path Planning

Dr. Agostino Stilli
Department of Medical Physics and Biomedical Engineering
University College London

December 10, 2025

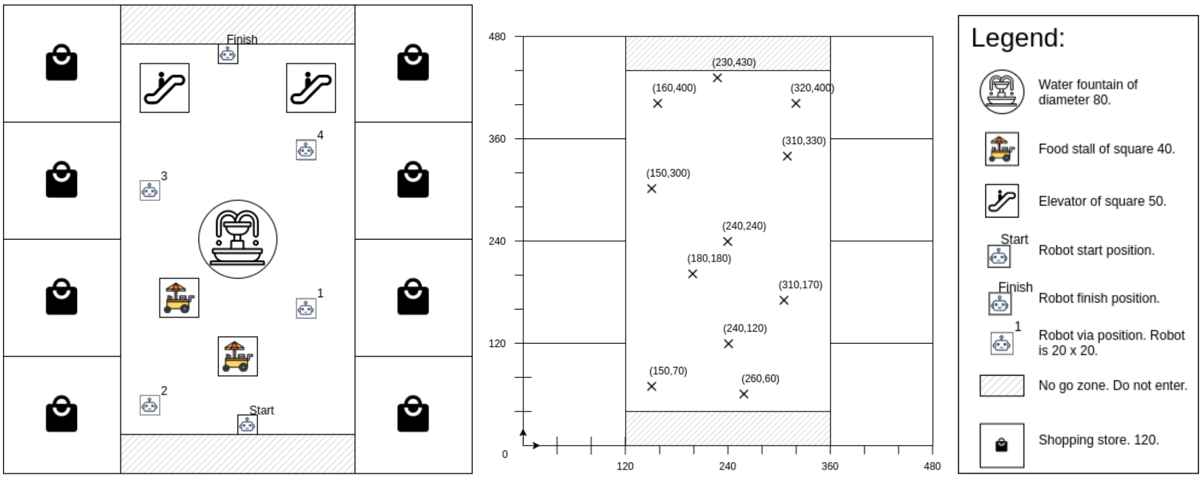
To get full credit for an answer, you are *required* to provide a .pdf report, and a fully working coding solution by filling in the provided code templates. These templates provide additional information on how to implement each script. **Do not remove anything from the templates and try to only fill in the code in the specified fields.** For the coding questions, you are also expected to include a simple breakdown of your algorithms in the report. When ready, *upload* your 'cw2' package on Moodle along with your submitted coursework report, in .zip or .rar extension. The necessary ROS packages are available on the course's *GitHub repository*.

Path and Trajectory Planning

1. Assume the following scenario: A shopping mall is testing autonomous cleaning robots to clean the floor of a section of the mall. They have given you the following floor map defining no-go zones and cleaning via points, and obstacles. Cleaning will occur at night, so no dynamic obstacles will be present. Consider only passing through via points, not total floor coverage. They also allow external cameras and tracking, so perfect odometry can be assumed. Present a robotic solution by choosing a drive system to complete the task, addressing path planning and trajectory planning. You can assume the position and orientation of your chosen robot is given as $q = (x, y, \theta)$ where x and y describe the position and θ describes the orientation of the robot. You have a perfect controller that can control your wheel velocities to attain the given position and orientation. When choosing the drive system, consider that the vacuum is located at the back of the robot, while the brush is at the front, therefore, depending on how the robot handles turns and rotations, the vacuum may miss water pickup. Specifically, your solution should address the following. (The recommended answer is up to 50 words per sub-question and the word count for the entire question should not exceed 300 words.)

   a. Present your drive system configuration and explain your reasoning. Is it holonomic or non-holonomic and why? What is the configuration space? [report - 5 pts]

   b. Describe how you would find a path that goes from the starting point to the final point while passing through all the via points in order. How would planning change if the ordering did not matter? [report - 5 pts]



(a) Map (b) Coordinates (c) Legend

Figure 1: Map, coordinates, and legend of shopping mall cleaning area.

c. Describe a time scaling function you would use when constructing a trajectory. Consider initial and final accelerations as well as at the via points. [report - 5 pts]

d. The robot is designed with a water vacuum at the back of the robot, meaning during turns and rotations, the vacuum may miss water pick-up. What ways can you design your path or trajectory to prevent left-over water? [report - 5 pts]

e. Describe a path-planning approach for full floor coverage while avoiding obstacles. [report - 5 pts]

[25 pts]

2. Complete the following question by filling in the 'cw2q2/src/cw2q2_node.py' python code templates to perform path planning via the shortest path. A simple code breakdown in the report is required for all subquestions, except subquestion e, which is code only. You are given target joint positions in the bagfile 'cw2q2/bags/data_ros2/data_ros2.db3'. Your task is for the Youbot end-effector to reach each target Cartesian check-point associated with each target joint position, via the shortest path in Cartesian space. To solve the question you need to:

   a. Implement the "load_targets()" method that loads the target joint positions from the bagfile and calculates the target end-effector position. [report - 2 pts, code - 3 pts]

   b. Implement the "get_shortest_path()" method that takes the checkpoint transformations and computes the order of checkpoints that results in the shortest overall path. [report - 3 pts, code - 5 pts]

   c. Implement the "decoupled_rot_and_trans()" and "intermediate_tfs()" methods that take the target checkpoint transforms and the desired order based on the shortest path sorting, and create intermediate transformations by decoupling rotation and translation. [report - 2 pts, code - 5 pts]

d. Implement the "ik_position_only()" and "full_checkpoints_to_joints()" methods that take the full set of checkpoint transformations, including intermediate checkpoints, and compute the associated joint positions with position-only inverse kinematics. [report - 2 pts, code - 8 pts]

e. Implement the "q2()" method, the main method of this question, where other methods are called in order to perform the path planning task. [code - 5 pts]

[35 pts]

Actuators and Mechanisms

3. A conveyor belt inside a candy factory carries colorful hard candy of quasi-spherical shape (e.g. Skittles, M&Ms, etc.). The candy is assumed to be stationary with respect to the conveyor, which moves at a constant velocity that cannot be altered. The conveyor belt is approximately 20cm wide and the candy appears randomly along its width. Assume the weight and shape of the candy are known. You are asked to implement a robotic system in addition to the conveyor belt to extract a candy of any given color and place it in a separate basket. The system should be able to perform the task as fast as possible. Ensure that you are designing a minimum viable system to solve the task, hence, avoid for example redundancy in your manipulator design as well as your actuator and sensor choices. Propose a robotic system to perform the task. In your answer, address the following. (**Recommended answer is up to 50 words per sub-question.**)

   a. Required degrees of freedom of your manipulator in the Cartesian space of the end effector. [report - 2 pts]

   b. Manipulator topology (serial/parallel) and design. You are free to reference commonly utilized designs and robot types. Also, state a reasonable choice of end-effector. [report - 2 pts]

   c. Choice of actuators (e.g. stepper motors, AC motors, brushed or brushless DC motors, pneumatics, hydraulics, etc.) and transmission. This will vary greatly with your manipulator design. [report - 2 pts]

   d. Choice of sensors (both, required for driving the manipulator as well as determining the correct object). [report - 2 pts]

   e. Discuss a component of the proposed system (e.g. joints, actuators, sensors) which would be prone to wear from repeated task execution. How could the wear be minimized? [report - 3 pts]

   f. Assume, only candy with the previously specified color above a certain weight threshold (e.g. > 5 grams) should be placed in the basket. How could you modify your system to be able to consider this additional factor without modifying the conveyor? [report - 4 pts]

[15 pts]

Robot Dynamics

4. Complete the following tasks by filling in the "cw2q4/src/cw2q4/iiwa14DynStudent.py" python class code template, to compute the dynamic components for the KUKA LBR iiwa14 manipulator. A simple code breakdown in the report is required for all subquestions. In the cw2q4 folder you can find three files.

   - iiwa14DynStudent.py: This is the class template for the questions below.
   - iiwa14DynBase.py: This class includes common methods you may need to call in order to solve the questions below. You should not edit this file.
   - iiwa14DynKDL.py: This class provides implementations to the questions below in KDL in order to check your own solutions. You should not edit this file.

   a. Write a script to compute Jacobian at the center of mass for the iiwa14 manipulator. [report - 2 pts, code - 3 pts]

   b. Fill in the appropriate class method to compute the dynamic component $\mathbf{B}(\mathbf{q})$ for the iiwa14 manipulator. [report - 2 pts, code - 8 pts]

   c. Fill in the appropriate class method to compute the dynamic component $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ for the iiwa14 manipulator. [report - 2 pts, code - 6 pts]

   d. Fill in the appropriate class method to compute the dynamic component $\mathbf{g}(\mathbf{q})$ for the iiwa14 manipulator. [report - 2 pts, code - 5 pts]

   [30 pts]

5. Describe the Huygens-Steiner theorem. Your answer should include a description of its hypothesis and of its derivation. Explain why it is important in robotics applications. [report - 7 pts]

6. Define the forward and inverse dynamics problems, and highlight their main applications. Describe what are the difficulties of each problem. [report - 8 pts]

7. In this question, you are tasked with computing the joint accelerations throughout the trajectory defined in the bagfile "cw2q7/bag/data_ros2/data_ros2.db3" using dynamic components. You should also plot the computed joint accelerations. You only need to edit the "cw2q7.py" file, and the corresponding launch file. Note that a coding template is **not** provided for this question. For the dynamic components, you can use either your own implementation from Q2, or the corresponding KDL class.

   a. Load the bag from the bagfile. What type of message does the bagfile contain? How many messages does the bagfile have, and what is the content of the messages? [report - 3 pts, code - 4 pts]

   b. Is this a problem of forward or inverse dynamics? [report - 3 pts]

c. Publish the trajectory to the appropriate topic to see the robot moving in simulation. [report - 2 pts, code - 3 pts]

d. Subscribe to the appropriate topic, and calculate the joint accelerations throughout the trajectory using dynamic components. [report 3, code - 12 pts]

e. Plot the joint accelerations as a function of time in your python script. [report - 5 pts, code - 5 pts]

Synchronizing ROS rate, bagfile reading, rviz executing the trajectory, and the function call can prove tricky. Thus, adding artificial delays in parts of the script is allowed.

[40 pts]

END OF COURSEWORK