The IRB 910 SC (pictured in Fig. 1) is a SCARA robot manufactured by ABB (Zurich, Switzerland). The robot has four axes, and it is rated for a maximum payload of 6 kg. The schematic in Fig. 1(b) shows the robot in its <u>home configuration</u>.

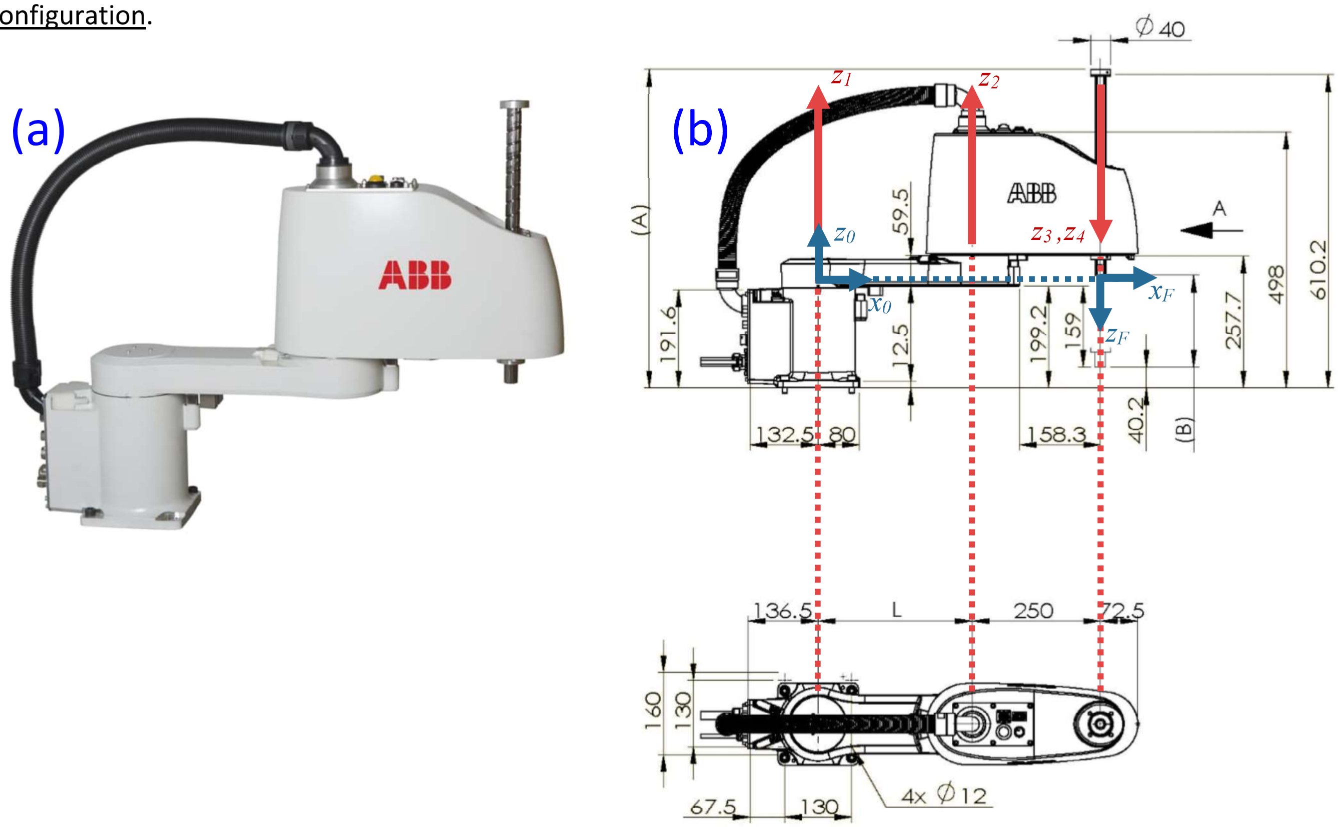

Figure 1: (a) The IRB 910 SC robotic arm; (b) Robot schematic with dimensions. All dimensions are in millimeters. Frame {0} is the space frame. Frame {F} is the end-effector frame. joint axes ($z_1, \ldots, z_4$).

<u>IMPORTANT</u>: The IRB 910 SC robot comes in three different variants. In this exam, we will model the IRB 910SC-3/0.55, whose lower arm is 300 mm long (see dimension labeled as "L" in schematic in Fig. 1(b)).

1. **Calculate each of the screw axes $\xi_i = (\omega_i, v_i)$ with respect to the space frame.**
   Show all the vector calculations needed to derive the screw axes.

2. **Calculate the homogeneous transformation matrix M representing the home configuration of the robot.**

3. **Create a model of the robot's kinematics in MATLAB.**
   Open the starting code that was provided to you. Find the script called `poe.m`, open it, and run it – this will display the robot. Complete this script to calculate S and M, that is, a list of screw axes, and the robot's home configuration.

4. **Implement a pick-and-place task in simulation.**
   Open the starting code that was provided to you. Find the script called `ik.m`, and run it. A new window will appear (see Fig. 2), showing (1) the robot in its home configuration, and (2) a path that we want the robot to follow to pick an object located at its right and move it to a location at its left. Your goal is to

implement the inverse kinematics of the robot. Use an inverse kinematics method of your choice among the ones we have covered in class. The code is expected to create a 10×4 `qList` matrix where each row is a set of joint variables necessary to trace the path.

Hint: As this inverse kinematics problem only requires solving for the end effector position, you may find it convenient to use the Analytic Jacobian.

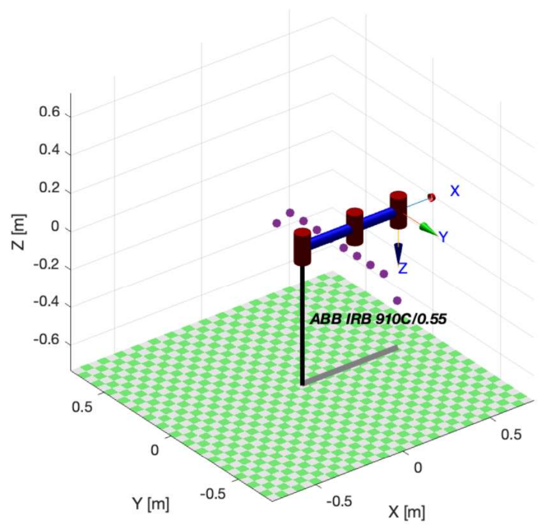

Figure 2: Kinematic chain of the robot, and path that we wish to trace (purple dots).

To demonstrate that your inverse kinematics solution is correct, create an animation in MATLAB using the following line of code:

```
robot.plot(repmat([qList; flip(qList)], 10, 1), ...
           'trail', {'r', 'LineWidth', 5});
```