**Package overview:**

- Inside `catkin_ws/src`, the main package is `scara_robot`. It does not directly contain any nodes or launch files, but is a way to organize all of the other nodes.
  - New package:
    - The `scara_pd_controller` package implements a proportional and derivative controller for the three controllable joints: joint 1 (revolute), 2 (revolute), and 5 (prismatic joint). The controller functions by reading the current joint position using the `gazebo/get_joint_properties` service, calculating the necessary input into the joint, and applying the input force using the `gazebo/apply_joint_effort` service. The controller receives the desired reference position using a custom service message under `scara/JointControlReference`.
  - Old packages used (from PA #1):
    - The `scara_gazebo` package includes the launch files for the gazebo world.
    - The `scara_description` package includes the URDF files for the robot as well as the rviz launch files.

**Problems:**

1. Velocity Level Kinematics: Implement a node with two services. One takes joint velocities and converts them to end effector velocities, and the second one takes end effector velocities and converts them to joint velocities.
2. Extend the position controller in Part 2 to all the joints. (don't forget to revert the joint types.) Move the robot to a position that is significantly away from singular configurations using you position controllers
3. Write velocity controllers for all the joints. For tuning the controller gains, you might need to fix the joints rather than the joint of consideration. Don't forget to revert the joint type to movable ones once you are done.
4. Give a constant velocity reference in the positive $y$ direction of the Cartesian space. Convert this velocity in to the joint space using your Jacobian and feed it as a reference to your velocity controllers. This should make the robot move on a straight line in the $+y$ direction. Record the generated velocity references together with the actual velocity of the system over time, and plot via Matlab.