Motion Planning Assignment

Minh Quan DAO

1 Introduction

This report is to demonstrate how to do the motion planning for the Denso manipulator such that it can perform a pick-and-place task in the environment described by Fig.1. The robot's task is to pick the boxes inside the pink container, then place them in one of two destination bays denoted by the blue box. Robot's motion can be either free or under the constraint that the tilt angle of the picked box is below 1 degree during its motion. Nevertheless, robot's motion must be collision free and respect joints' velocity and acceleration limit.

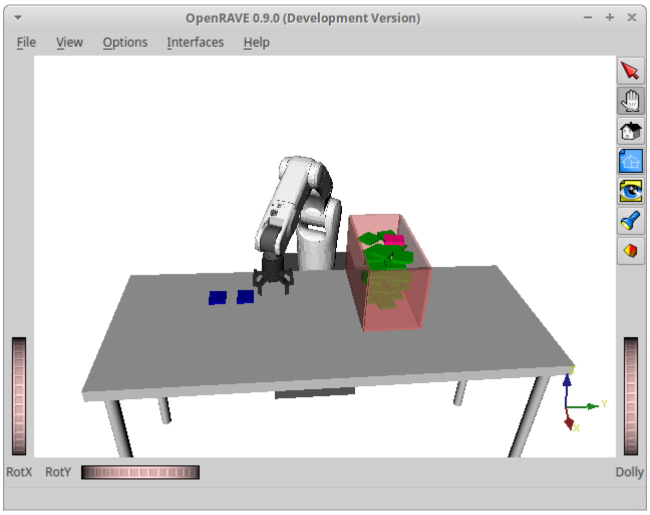

Figure 1: Environment of pick-and-place task

The flow chart of the pick-and-place program for each box is displayed in Fig.2

2 Pick and Place

2.1 Generate Pick and Place pose

A correct grasping pose is categorized by:

- the gripper is perpendicular to the surface of the box
- the finger pads are parallel to the edges of the box

The normal direction of the surface of the box is the z-axis of the box frame, while the z-axis of the gripper frame shown in Fig.3 is parallel to gripper finger pads and points outward. Therefore, for the gripper frame to satisfy the first condition, its z-axis must be in the reversed direction of the box z-axis.

As illustrated in Fig.3, the y-axis of the gripper is perpendicular to the finger pads. As a result, to grip a box along its x-axis or y-axis, the gripper y-axis must be parallel to box y-axis or x-axis, respectively.

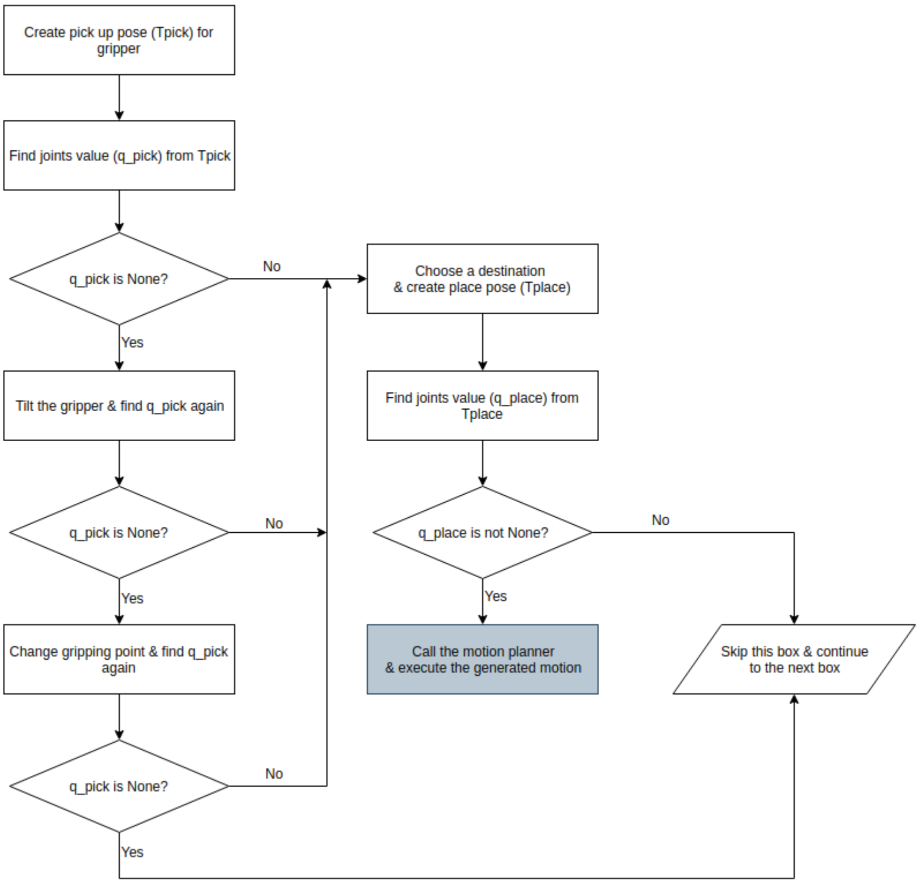

Figure 2: Pick-and-place flowchart

With the reasoning above, the picking orientation is defined by

$$
\begin{aligned}
z_{pick} &= -z_{box} \\
y_{pick} &= y_{box} \quad or \quad x_{box} \\
x_{pick} &= y_{pick} \times z_{pick}
\end{aligned}
\tag{1}
$$

The gripping position is initially chosen to the centroid of the box. This position is subjected to change in case the inverse kinematics (IK) solution to the picking pose does not exist.

The placing pose is defined using the same method as the picking pose. However, the z-coordinate of the placing pose is offset by the height of a green box ($10(mm)$) plus $1(mm)$ every time a green box is successfully placed.

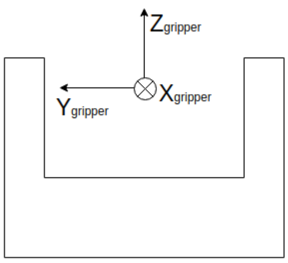

Figure 3: Gripper frame

2.2 Modifying picking pose

There is a possibility that there is no collision free IK solutions to the picking pose generated by the previous subsection. In such situation, to pick up the targeted green box, one can try either tilting the gripper or sliding the gripping position.

2.2.1 Tilting the gripper

The tilted gripper is defined by a rotation of the gripper frame around its own y-axis.

$$T_{gripper}^{tilted} = T_{gripper} \cdot Rot_y(\alpha) \tag{2}$$

with $Rot_y(\alpha)$ defined by

$$Rot_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The tilt angle α starts from 0 and is increased by a small angle (e.g. $5(deg)$) each iteration until IK solutions exist or the maximum number of iterations is reached. The pseudo code for tilting the gripper is in algorithm 1.

Algorithm 1 Tilt gripper

procedure TILTGRIPPER()
    $it \leftarrow 0$
    repeat
        $q_{pick} \leftarrow$ collision free IK solution to T_{pick}
        if q_{pick} is None **then**
            $T_{pick} \leftarrow$ rotate T_{pick} around its y-axis by α
            $it \leftarrow it + 1$
        else
            break
    until $it = it_{max}$
    $tilt\ angle \leftarrow \alpha \cdot it$
    return q_{pick}, $tilt\ angle$

If a collision free IK solution still does not exist after tilting the gripper to the maximum angle, one can try changing the axis of the box along which the gripping is performed, then tilt the gripper again to find q_{pick}.

It is worth noticing that when the gripper is tilted by *tilt angle*, the placing pose must be tilted by the same angle.

2.2.2 Sliding the picking position

When tilting gripper fails to produce any collision free IK solutions, the gripping position can be slided along the gripper frame's x-axis by a distance d in attempts on finding a T_{pick} that robot can reach.

$$T_{gripper}^{slided} = T_{gripper} \cdot \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The sliding distance d starts from 0 and is increased by a small amount (e.g. $10(mm)$) until IK solutions exist or the gripping position is no longer on the box. The implementation of sliding the picking position is similar to Algorithm 1. In addition, like the tilted gripper, when the picking position is changed, the placing position need to changed with the same amount.

3 Boxes' tilt angle

Once a box is gripped, its normal direction and the gripper's z-axis are displayed Fig.4 If the gripper is not tilted, the tilt angle of the box is equal to the angle between the reversed of gripper z-axis and the world z-axis. If the gripper is tilted, the z-axis of the gripper frame needs to be rotated around gripper y-axis by the negative amount of gripper tilt angle, before being reversed and used for calculation of the box's tilt angle.

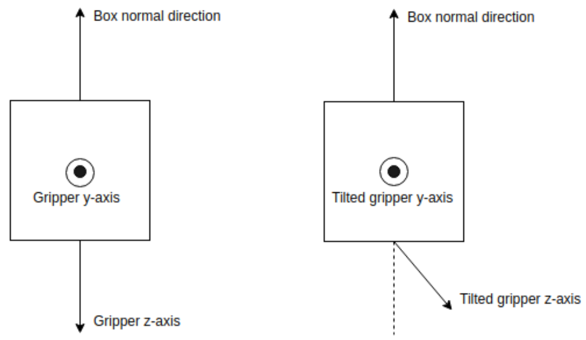

Figure 4: Box's normal direction and gripper z-axis

The pseudo code for calculating boxes' tilt angle is displayed in algorithm 2.

Algorithm 2 Compute boxes tilt angle

procedure BOXTILTANGLE(*trajectory*)
  $times \leftarrow$ time stamp of the trajectory
  $qvect \leftarrow$ joints angle on the whole trajectory
  $box\ tilt\ angles \leftarrow \emptyset$
  for every point in $times$ **do**
    $T_{ee} \leftarrow$ end-effector pose resulted by joints angle at this point of $times$
    if $tilt\ angle > 0$ **then**
      $T_{ee} \leftarrow T_{ee} \cdot Rot_y(-tilt\ angle)$
    $box\ tilt\ angles \leftarrow box\ tilt\ angles\ \cup \angle($ z-axis of T_{ee} , global z-axis)
  return $box\ tilt\ angles$

4 Constraint Motion

Initially, every box has its tilt angle equal to 0. Therefore, for them to have small tilt angle (less than $1(deg)$) during the pick-and-place task, the motion of the gripper must not be made of the rotation around global x- or y-axis. In other words, robot's motion only includes rotation around z-axis and translation along x-, y-, or z-axis (all axes are belong to world frame).

The implementation of the constraint motion is based on the section "moving mug without global XY rotation" in [1]. In addition, to reduce the planning time, each box after being gripped is pulled close to the robot, lifted up, then translated outside of the pink container toward the destination bays. All of these motions are along a straight line. Once this sequence of straight motions is finished, the constraint motion planner is invoked to generate the final trajectory to move the gripped box to the chosen destination.

The pseudo code for constraint motion is displayed in algorithm 3.

The tilt angle of the first 5 boxes (initialized in the same environment) moved without and with X, Y rotation constraint are respectively displayed in Fig.5 and Fig.6. It can be seen from these figures that the tilt angles during the constraint motion is much smaller, and their value are below 1 degree, except for box 1.

Algorithm 3 Move one box

procedure MOVEONEBOX($q_{pick}, q_{place}, T_{place}$)
Require: q_{pick}, q_{place} are not None
    $up\ direction \leftarrow \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$
    $close\ direction \leftarrow \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$
    $side\ direction \leftarrow \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$
    Move all joints to position q_{pick}
    Move gripper along a straight line in *up direction*
    Move gripper along a straight line in *close direction*
    Move gripper along a straight line in *side direction*
    $trajectory \leftarrow$ move gripper to T_{place} under the constraint of no X, Y rotation
return *trajectory*

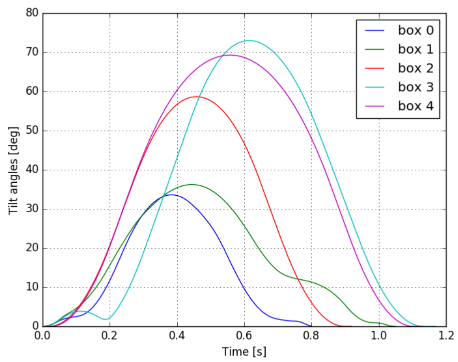

Figure 5: Boxes tilt angles during motion without constraint

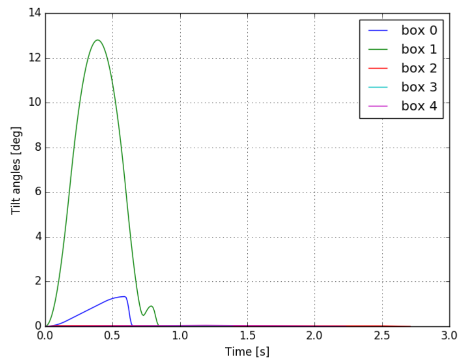

Figure 6: Boxes tilt angles during motion with constraint

References

[1] http://openrave.org/docs/0.8.2/_modules/openravepy/examples/constraintplanning/#ConstraintPlanning