Human-in-the-loop Robot Learning

Jianming Xiao

1 Introduction

This is an approach for teaching a Robot how to clean up a table. In this approach I divided the whole process into 4 stages because in each stage the status conditions and actions are different, these stages are: moving to the trash; pick up the trash; moving to the dust bin; and put the trash in the dust bin.

For the whole approach, robot will have one reinforcement learning agent with human in the loop to provide the rewards to handle the different status and actions, and the learning algorithm will use Evaluation Reinforcement (TAMER) algorithm.

The following chapters will discuss about the detail about each stage and then how to translate this approach to use Cozmo and its cubes to simulate.

2 Actions and status

This chapter will describe the status of Robot and the possible actions on different situation.

2.1 Status

For this task we should know the following information from the Robot, Trash, and Dust Bin:

- Position and angle of Robot ($Pos_R \leftarrow (X_R, Y_R, Angle_R)$)
- Position and angle of Trash ($Pos_T \leftarrow (X_T, Y_T, Angle_T)$), so the distance between Robot and Trash can be easily calculated ($D_{R \to T} \leftarrow \sqrt{(X_R - X_T)^2 + (Y_R - Y_T)^2}$).
- Position and angle of Dust Bin Trash ($Pos_{DB} \leftarrow (X_{DB}, Y_{DB}, Angle_{DB})$), so the distance between Robot and Dust Bin can be easily calculated ($D_{R \to DB} \leftarrow \sqrt{(X_R - X_{DB})^2 + (Y_R - Y_{DB})^2}$), and the distance between Trash and Dust Bin should be ($D_{T \to DB} \leftarrow \sqrt{(X_T - X_{DB})^2 + (Y_T - Y_{DB})^2}$).
- Has Robot carried Trash ($B_C = \{ True, False \}$)
- Has Trash already in the Dust Bin ($B_G = \{ True, False \}$)

And of cause all these status information will have a probability to get the real data.

For the goal status, should be Robot near the Dust Bin and successfully put the Trash into the Dust Bin, that means in $getAction(s)$ function will have:

Partially Function $getAction(s)$

```
1. if B_C == False and B_G == True and D_{T→DB} ≈ 0
2.     exit('succes')
```

2.2 Actions

For simplify the description, let's assume the robot has the movement is easily have move toward and move further of an object. So, the action will have:

- Move to certain position $Pos_R \leftarrow (X_i, Y_i, Angle_i)$
- Pick up the Trash $B_C \leftarrow True$
- Put the Trash into Dust Bin $B_C \leftarrow False, B_G \leftarrow True$

Not all actions are available in all Approach Stages, please see the detail in next chapter.

3 Approach Stages

We already divided the whole process into the following 4 stages:

3.1 Moving to the trash

Condition: $B_C == False$ **and** $B_G == False$ **and** $D_{T \to T} > 0$

Support Actions:

- Move to certain position $Pos_R \leftarrow (X_i, Y_i, Angle_i)$

When Robot move toward the Trash, human should give the positive reward, and on other case, should return negative rewards from human. Result will become let Robot closer and closer to the Trash $\lim_{i \to \infty} D_{R \to T} = 0$.

3.2 Pick up the trash

Condition: $B_C == False$ **and** $B_G == False$ **and** $D_{T \to T} \approx 0$

Support Actions:

- Move to certain position $Pos_R \leftarrow (X_i, Y_i, Angle_i)$
- Pick up the Trash $B_C \leftarrow True$

When Robot pick up the Trash, human should give the positive reward, and on other case, should return negative rewards from human. Result will become after many times learning iteration Robot will directly pick up the Trash at this stage $\lim_{i \to \infty} P(B_c = True) = 1$.

3.3 Moving to the dust bin

Condition: $B_C == True$ **and** $B_G == False$ **and** $D_{T \to DB} > 0$

Support Actions:

- Move to certain position $Pos_{Ri} \leftarrow (X_i, Y_i, Angle_i)$

When Robot move toward the Dust Bin, human should give the positive reward, and on other case, should return negative rewards from human. Result will become let Robot closer and closer to the Dust Bin $\lim_{i \to \infty} D_{R \to DB} = 0$.

3.4 Put the trash in the dust bin

Condition: $B_C == True$ **and** $B_G == False$ **and** $D_{T \to DB} \approx 0$

Support Actions:

- Move to certain position $Pos_R \leftarrow (X_i, Y_i, Angle_i)$

- Put the Trash into Dust Bin $B_C \leftarrow False, B_G \leftarrow True$

When Robot throw the Trash into, human should give the positive reward, and on other case, should return negative rewards from human. Result will become after many times learning iteration Robot will directly put the Trash into Dust Bin at this stage $\lim_{i \to \infty} P(B_c = True) = 1$.

4 Cozmo Simulation & Approach

When work with Cozmo we can use the cubes as the Trash and Dust Bin, and by tapping the different cubes, will give Cozmo positive and negative rewards.

4.1 Translate the concept

Cozmo support some building actions which can help us to reduce the work of implement the robot control in the entire approach.

4.1.1 Cubes

For the 3 cubes, they will have different meaning and usage in the Approach:

- Cube 1, blue light, as the Trash.
- Cube 2, red light, after each action Cozmo take, tapping this cube will means the negative rewards; also, as the Dust Bin.
- Cube 3, green light, after each action Cozmo take, tapping this cube will means the positive rewards.

4.1.2 Actions

4.1.2.1 Move to certain position

Cozmo has built in action (**go_to_pose**) to go to a certain position:

```
1. go_to_pose(Pose(100, 100, 0, angle_z=degrees(45)), relative_to_robot=True)
```

But in order to reduce the learning time, we can use another action (**go_to_object**) in the simulation:

```
1. robot.go_to_object(cube1, distance_mm(70.0))
```

And we only allow robot to do 4 moving related actions:

- Move closer to trash
- Move further to trash
- Move closer to dust bin
- Move further to dust bin

```
1. robot.go_to_object(cube1, distance_mm(D_{R→T} − 50))     # move closer to trash
2. robot.go_to_object(cube2, distance_mm(D_{R→DB} − 50))    # move closer to dust bin
```

For 2 move further actions we simplify use a hack way to do it

```
1. # move further to trash
2. robot.go_to_object(cube1, distance_mm(D_{R→T} − 50))
3. robot.drive_straight(distance_mm(-50), speed_mmps(50))
```

And

```
1. # move further to dust bin
2. robot.go_to_object(cube2, distance_mm(D_{R→DB} − 50))
3. robot.drive_straight(distance_mm(-50), speed_mmps(50))
```

4.1.2.2 Pick up the trash.

As we previously assumption, cube 1 is the object to present the trash. So, for Cozmo, it can pick it up the cube by this action.

```
1. robot.pickup_object(cube1, num_retries = 3)
```

4.1.2.3 Put the Trash into Dust Bin

As we previously assumption, cube 3 is the object to present the trash. So, for Cozmo, when it had already picked up the cube 1 (trash), when it place the cube 1 on top of cube 3, that means it put the Trash into the Dust Bin.

The following action can do that:

```
1. robot.place_on_object(cube2, num_retries = 3)
```

4.1.3 Rewards

When doing the learning, Cozmo will ask if that action is correct or wrong. Because of the Cozmo SDK don't support detect 2 cube's tabbing at same moment, it will ask "Is that correct step?" first, and during this time, cube2's light (red) will off. If no one tab cube3, Cozmo will ask "So is that wrong step?", and at this moment, cube3's light (green) will off.

For the positive reward (tab cube3), will give +1 rewards. On the other hand, negative reward (tab cube2) will be -1.

4.1.4 stage

4.2 Code

The code can be found in the GitHub repository: https://github.com/BigEgg/csep590robotics-labs, in the lab14 folder.

In the code there has 3 files:

- humanInLoopReinforcementLearningAgent.py: contains the learning agent logic.
- cleanTableRobot.py: a wrapper robot class for cleaning the table.
- main.py: the logic to start Cozmo and let it learn the process twice, and then do it by it own.

4.2.1 Running Program

By starting it, run ***python main.py*** will be enough.