# Garbage collector robot implementation using Q-Learning

---


**Under the guidance of :**
**Dr. Shivashankar B. Nair**

**Members :**
**Vinod Kumar(164101027)**
**Subash(164101037)**
**Piyush Singhal (164101023)**
**Rachith Chandran(164101030)**

**Department of Computer Science & Engineering**

## Indian Institute of Technology, Guwahati

**Abstract** – An approach to the problem of autonomous mobile garbage collecting and sorting robot using reinforcement learning is proposed in this paper. Q-learning is one kind of the reinforcement learning algorithm, which we used for learning the actions based on rewards. Two types of reinforcement learning algorithms are employed in garbage collecting and sorting task, namely single agent and multi-agent reinforcement learning algorithms.

# 1 Introduction

The Garbage is a huge problem in the world, where people does not follow the rules and throw the garbage in wrong places. We propose an approach for cleaning the garbage by using autonomous mobile garbage collecting and sorting robot using reinforcement learning algorithm, which collect the garbage and clean the area. In addition to that, the robot separates the garbage by sorting them into appropriate bins(place where the garbage should be dropped). The location of the bins is known to the robot, but what kind of garbage should be dropped is not known prior. The robot is not hard coded with which garbage to be thrown in which bin. We use Reinforcement learning algorithm namely Q-learning. This algorithm will pick each garbage and drop into any of the corner and based on the reward it will learn to drop the garbage into correct bin. The robots learn the environment and share the Q-Values in case of Multi-agent reinforcement learning method. The experimental results which is mentioned in further part in the paper shows that multi-agent reinforcement learning algorithm does a better job by reducing the number of error in terms of dropping garbage object into wrong bin and learns faster than single-agent reinforcement learning algorithm.

# 2 Problem statement

Autonomous garbage collection and garbage segregation, where the environment has different garbages scattered around, the robot should move around this environment and pick this objects(garbage) and drop it into appropriate bins, while keeping energy in check, if energy is getting lower than threshold, the robot should go to charing station and recharge its energy.

# 3 Algorithm

We have a garbage collection robot, which collects the garbage object and drops them into the corresponding bins. Initially, the robot doesnt know where to drop each garbage, within the given garbage bins. The robot does mistake and learns from the environment. Finally, the robot will learn to drop each object in the corresponding bins.
We define each discrete step of environment as states for the robot. We have the following states for our problem.
States = [ Open, RedBall, YellowBall, GreenBall, CornerA, CornerB, CornerC, CornerE, Batter, Wall, Object ]
We have defined the following actions for our robot to achieve the goal.
Action = [ GotoA, GotoB, GotoC, GotoE, Forward, Backward, Left, Right, Grab ]
We define desirability(Q-Value) for each state and action pair. For a given state, how much desirable is each action. For a given state, the robot refers this desirability and chooses the most desirable action. For each action, the environment either rewards the robot, if the robot has taken a right action or penalizes the robot for choosing an undesirable action. Our Q-learning algorithm, takes this state, action and the reward as input and updates the desirability for the given state and action. Thus, next time when an action is chosen, the action which was the reason for penalization will be less desirable.
In our problem, we have to pick the garbage object and drop it in a bin. In our environment, the bins are kept at the corner of the arena. The corners are defined with respect to the direction. Imagining one side of the arena to be the North, one of the corner would be North East. We cannot expect at least one side of the arena to be perfectly aligned towards North. Hence, we calibrate the direction of one side of arena to North, at the beginning of experiment. It takes this side as North and starts the garbage collection work.
To achieve the assumption, we have initialised the Q-value for many state action pairs, with the domain knowledge of the problem. The Q-value of the state action pair gets updated in the following manner:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(reward + \gamma \ max_{a'} Q[s',a'] - Q[s,a]) \tag{1}$$

Our problem statement includes, adaptation to change of environment. Though the bins are interchanged in position, robot should understand the change in environment and learn the correct position. To achieve this requirement,

we check the previous reward for same state action pair. I.e. for each of the state of the Q-learning, we are expecting same reward. When the reward is changing then, it means that the environment has changed. We adapt by reverting the Q-value to initial values for the corresponding state action pair. Thus the robot will be able to adapt to new environment fast. If we use multiple robot inside arena, we can update our Q-value with respect to change in other robot. The message which we send to the other robot resembles the following format.
[ state, action, qval, previousReward, currentReward ] to all neighboring robots by broadcasting the message.
After receiving the QValue from the neighbour robot, we accept the QValue with a constant Beta, if the neighbour has new environment information(change in reward value).

$$Q[s,a] = \beta * Q[s,a] + (1-\beta) neighbourQ[s,a] \tag{2}$$

# 4 Experiments and Results

## 4.1 Assumptions

- Different garbages are modelled as hexagonal prism shaped colored objects red, green , yellow.
- Actual battery level of the robot is not considered for energy, we have defined energy as 100 initially internally and then each operations will consume some energy , which will be reduced from the internal value to mimic the actual energy consumption.
- Only 3 colors of objects exists in environment namely red, green and yellow.
- There are 3 colored bins(square colored paper in floor as mentioned in robot map below in this paper) namely red,green,magenta. Where red object is dropped in red bin in floor, and green object to green bin and yellow object to magenta bin.
- The environment as mentioned in environment design is placed in any direction but the calibration is required.
- The calibration is where the robot is placed in the environment to correct the error from the north(i.e the degree of deviation from actual north should be given to robot to correct it actual direction).

- We have to learn only the object drop location, other intuitions like avoiding the wall and charge itself during low charge is known to the robot.

## 4.2 Technologies Used

- Software
  1. 32 bit jdk with version 8
  2. Nxt development software
- Hardware
  1. Lego Nxt Kit
  2. Raspberry Pi 2
  3. Battery

## 4.3 Robot Configuration

- The robot is designed with following sensors
  1. Color Sensor - We used two color sensor for floor and object.
  2. Compass Sensor - Required to find direction , and used it for navigation.
  3. UltraSonic Sensor - Required to find obstacle in front of the robot.
- Actuators used
  1. Motors - two motors used for driving and one motor for claw control.
  2. Claw is used to pick the object and drop color object.
- Robot Communication
  - For extra computation power we used Raspberry Pi , which is used to communicate with other robot and wifi dongle is used for communication with other robots.
- Raspberry Pi Usage in Experiment
  - By placing raspberry pi on lego enables us to use the extra processing power of pi and also use the wifi features available in the pi for our robot communication.
- Below is the robot

Figure 1: With Raspberry Pi mounted on lego(Side View).

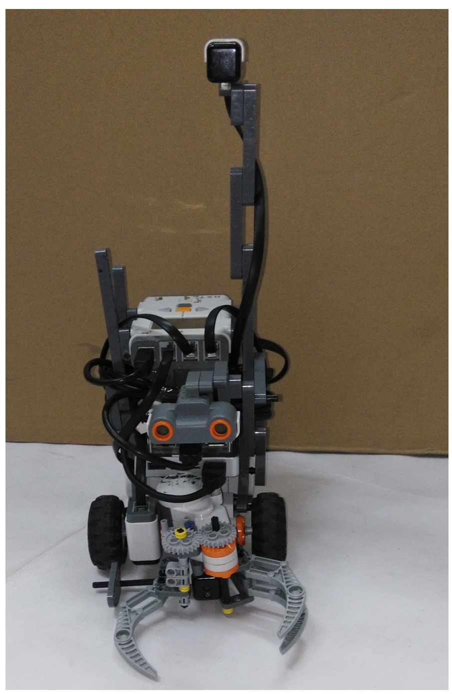

Figure 2: With Raspberry Pi mounted on lego(Front View).

Figure 3: Without Raspberry Pi(Side View).

Figure 4: Without Raspberry Pi(Front View).

### 4.4 Environment Design

The Environment has four station, Station A has Red floor, Station B has Green floor, Station C has Magenta Floor. While one of the colour sensor aids in identifying the color of the object in front of the robot, the other mounted beneath, facilitates detection of the color on the surface of the arena. The ultrasonic sensor installed in front of the robot aids in identifying obstacles encountered on its path. The compass sensor helps the robot to navigate within a 2 m x 2.1 m arena.

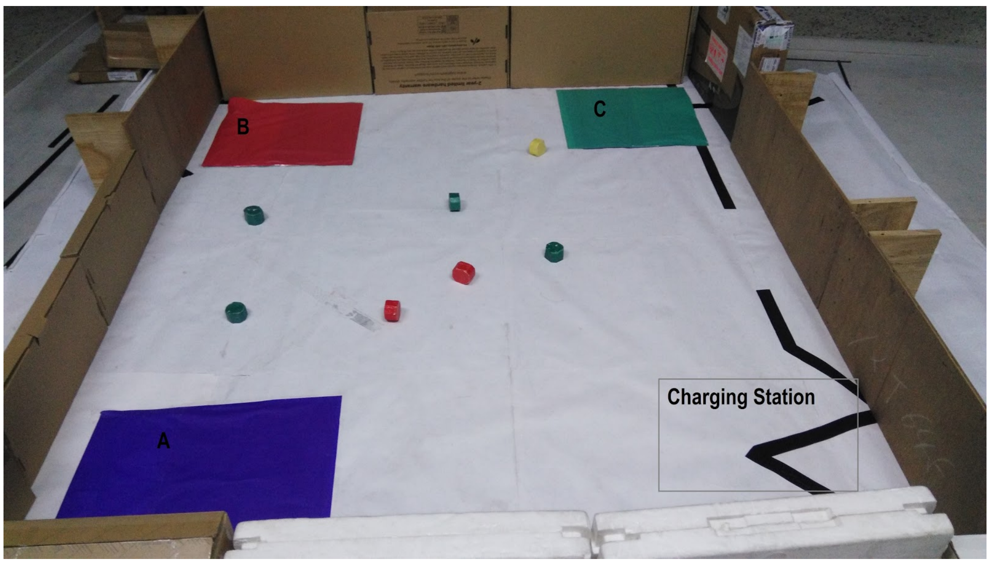


Figure 5: Without Raspberry Pi(Front View).

### 4.5 Experiment Details

Two mode of experiments were conducted , namely

1. Single Lego Garbage collector
   In this mode single robot will be running Q-learning algorithm and pick colored object and drop it into one of A,B,C corner. Though the algorithm dropped few garbage objects in wrong bin at the learning phase, the robot learned to drop the corresponding object to respective bins.

2. Multi Lego Garbage Collector
   In this mode multiple robot will be running Q-learning algorithm and

pick the colored object and drop it into one of A,B,C corner, here Raspberry Pi will be mounted on the lego for extra computation. Though the algorithm dropped few garbage objects in wrong bin at the learning phase, the robot learned to drop the corresponding object to respective bins.

## 5 Limitations

- Compass Sensor was giving faulting direction many times , which caused robot to move in wrong direction. An solution to this problem is to use two Compass Sensor and take their average value for more accuracy.
- Communication within the robots had problems of wifi router dropping packets, due to security reasons, for this problem we configured our own wifi router for communication.
- The Actual energy could be considered for future development, where robots actual battery level should be considered for energy drain and on low energy it should go to charging station.

## 6 Conclusions and future research work

- Placing Led in the robot to indicate the station it is moving, like Red Led for station A, Green Led for Station B, Yellow Led for Station C and Blue Led for charging station.
- The robots doesnt know the actual location of other robots, if this information is given to the robot it can optimize the location to scan for garbage.
- In our case we are not using any effective search we are just following random path, if some efficient search algorithm is applied for garbage scanning the energy of the robot could be conserved.