

TDDC17 - LAB 1 - Report

The same logic is used for both Tasks 1 and 2. We have used a rule based approach for defining the actions that an agent must take. The approach makes use of randomness and data about the environment to take decisions in each iteration. This approach was chosen because it was easier to implement and requires lesser memory than alternative approaches like searches which requires to maintain a stack or a queue.

The main decision loop has been explained below,

1. The agent first checks if the complete world have been explored by comparing the number of unknown cells in its *world* data array and the size of the world if all the reachable regions are explore then it executes the code to go home by invoking the *go_home()* function.
 - *go_home()* function - The agents finds the x-y translations required to reach its home position by subtracting the current position from the agents home position. The agent then turns to the appropriate directions and moves forwards till it reaches home. If any bumps are encountered during this movement the agent move in random direction for 1-10 iterations. This is bad but simple way to fix the bump problem. The agent will finally reach home at some point.
2. If the full world has not been explored then the agent checks the percept data to do actions. The actions done when each percepts are given below:
 - i. If dirt is received as true the main loops updates the *world* data array that the current cell is dirty and returns *action_suck* to clean the dirt.
 - ii. If bump is received as true then the agent updates the *world* data array to denote the next cells in the direction that it is facing is a wall proceeds to turn by calling the *turn_vacuumAgent()* function.
 - *turn_vacuumAgent()* function - The agent always checks if the cells to its right and left have already been explored by using the *world* data array. If any one of the side has already been explored the agent turns to the other side. If both side are not explore or if both sides are explored the agent randomly turns to the right or left with a probability of 50%.
 - iii. If no dirt or bump has been encountered the agent checks if the next cell in the direction in which it is facing has been explore by using the *world* data array if not it moves forward else the agent randomly chooses between moving forward and turning with a probability of 50%, the turning is done using the *turn_vacuumAgent()* function mentioned above. The *world_data* array is also updated by marking the current cell as clear.

This approach suffers from problems like getting stuck in a local region when all the surrounding regions are explore. It will in turn escape and explore the complete state space but it will take time. A mix of searching and logic would have preformed better.