Lab - 5 Report

1. In the report, a) describe your choices of state and reward functions, and b) describe **in your own words** the purpose of the different components in the Q-learning update that you implemented. In particular, what are the Q-values?

Answer:

We had used 20 states. The idea behind using 20 states was based on dividing the region into quadrants. To increase the precision in each state we had chosen to divide it further 5 times which gave (4* 5 = 20 states). Dividing the quadrants five more times were got by experimentation, dividing the quadrants five more time give the optimal results than below them.

The reward function we had used was,

int MAX_ANGLE = 3;
double reward = MAX_ANGLE - Math.abs(angle);

The maximum possible angle value received from the sensors were 3, so this reward function subtracts the maximum angle value from the current angle value. To maximise the rewards the angle value should be around 0.

The Q Value denotes the quality value i.e it represent how useful an action on a state will be in gaining future rewards. The different components of the Q-Leaning update are,

1. The Q-Value of the current state - the update happens on top of the current Q Value
2. The learning rate(Alpha) - This limits the update that is happing to the current Q value. It donates how well the Q-learning update accepts new values when compared to old values. In the current implementation it is done such that the learning rate decreases with the number of times a node is visited
3. Reward for a state, action
4. The discount factor - It denotes the agents preference over current rewards and future rewards. If the discount factor close to 0 then the agents prefers current rewards and the rewards in the future are considered insignificant. If the discount factor is close to 1 then the agent waits for long term rewards.
5. The maximum reward in the new state that is reached

2. Try turning off exploration from the start before learning. What tends to happen? Explain why this happens in your report.

Answer:

The agent seems to be stuck in a particular action which was usually pretty bad and seem to not learn anything. This is because the agents has not checked other possibilities in the environment as is stuck in a local maximum.

Part 3 - Report

The rewards and states used for the angle controller task were mentioned above. The hover task also was implemented based on a similar intuition, We used 6 states each of angle, vx and vy. This was selected based on experimentation sufficiently good observations were received with this combination with short durations of training. The reward functions that were used was also similar to the angling task. The complete reward functions are given below,

```
int MAX_ANGLE = 3;
int MAX_VY = 1;
int MAX_VX = 1;

double reward_angle = (MAX_ANGLE - Math.abs(angle))/MAX_ANGLE;
double reward_vx = (MAX_VX - Math.abs(vx))/MAX_VX;
double reward_vy = (MAX_VY - Math.abs(vy))/MAX_VY;

double reward = (reward_angle+reward_vx+reward_vy);
```

This reward function was also on the same intuition as the angle controller. The reward of *vx* and *vy* was was calculated by subtracting the current value from a maximum value. Even though the max value for *vx* and *vy* was around 15 we gave the *max_vy* value as 1 so as to punish larger values in velocity. Larger values in *vx* and *vy* in these cases will lead to negative values in rewards for *vx* and *vy* which will cause the rewards to go down. This gave us pretty good results when compare to setting the *max_vx* and *max_vy* as 15 which was causing the rocket to move up at larger speeds.