Assignment - 07

1. ANS:-

State Space (S):

The state space represents the drone's current situation, including its position, velocity, and surroundings. We can define S as a tuple of:

1. GPS coordinates (x, y, z): The drone's current location.

2. Velocity (vx, vy, vz): The drone's current speed in each direction.

3. Obstacle detection: A vector indicating nearby obstacles (buildings, trees, other flying objects) within a certain radius, represented as binary values (0 = clear, 1 = obstacle).

4. Battery level: The drone's current battery percentage.

5. Package status: A binary value indicating whether the package is still on board (0) or delivered (1).

Action Space (A):

The action space represents the drone's possible movements and actions. We can define A as a set of:

1. Movement commands: {up, down, left, right, forward, backward, hover}

2. Speed adjustments: {increase speed, decrease speed, maintain speed}

3. Yaw adjustments: {turn left, turn right, maintain heading}

Reward Function (R):

The reward function encourages efficient and safe deliveries. We can define R as a combination of:

1. Delivery reward: +10 for successful package delivery

2. Time penalty: -0.1 for each time step taken to deliver the package

3. Safety penalty: -5 for each obstacle collision or near-miss (within a certain distance)

4. Battery penalty: -0.5 for each percentage point of battery drained

5. Smooth flight reward: +0.5 for smooth flight (minimal changes in velocity and yaw)

By defining the state space, action space, and reward function, we've formulated the RL task for our autonomous drone delivery system. Now, we can use RL algorithms to train the drone to navigate the urban environment efficiently and safely!

2. <u>ANS:-</u>

In Reinforcement Learning, the trade-off between exploration and exploitation involves balancing the need to explore new actions to discover potentially better rewards (exploration) and the need to choose the best-known actions to maximize immediate rewards (exploitation).

The ϵ\epsilonϵ-greedy strategy addresses this by:

- **Exploitation**: With probability 1−ϵ1 - \epsilon1−ϵ, the agent selects the best-known action.
- **Exploration**: With probability ϵ\epsilonϵ, the agent randomly selects an action.

Typically, ϵ\epsilonϵ starts high (favoring exploration) and decays over time (shifting towards exploitation), ensuring the agent explores sufficiently early on while focusing on the best actions later.

3. <u>ANS:-</u>

Let’s solve this using the Bellman equation. The Bellman equation for the value of state (s) is:

$v(s) = \sum(p(s'|s,a) * (R(s,a,s') + \gamma v(s')))$

We have two possible outcomes, so we'll calculate the expected value as:

$v(s) = 0.4 * (10 + 0.5 * 5) + 0.6 * (2 + 0.5 * 3)$

First, calculate the value for each outcome:

1. $0.4 * (10 + 0.5 * 5) = 0.4 * (10 + 2.5) = 0.4 * 12.5 = 5$

2. 0.6 * (2 + 0.5 * 3) = 0.6 * (2 + 1.5) = 0.6 * 3.5 = 2.1

Now, add the values:

v(s) = 5 + 2.1 = 7.1

So, the expected value of state s is approximately **7.1**.