Monte Carlo TLDR: Try different actions and see what happens, Remember how good or bad each action was, Average out how good or bad each action was, Adjust future actions based on averages

Monte Carlo: Used to estimate value function of a policy by averaging returns from many episodes

1. Generate episodes where agent interacts with environment
2. Computes returns for each action based on rewards
3. After an episode, update value function based on returns (averaging returns for each state pair)

- Monte Carlo could be computationally expensive due to needing to generating many episodes for value function estimation however is usually straightforward and effective
- Model free so they do not require a model of the environment
- Could use parallelization for episodes

State Representation: Can include UAV's current position, velocity, orientation, distance to target, obstacles, etc.
Action Space: Changing altitude, Changing velocity, Changing orientation, etc.
Reward Designs: Can be distance to target, safety, mission completion time