COMP550 Project G: Planning Under Uncertainty

Joshua Han (jh273) & Alan Huang (ah212)

I: Project Statement

In the real world, the mechanics of a robot are not perfect and the response of the robot to a command can vary. This variation is known as actuation error, and it can affect a planner's ability to find a path that satisfies the constraints of its environment. This is most critical in problem domains that prioritize avoiding collisions. In these environments, a feasible path that ignores uncertainty can be dangerous, and can lead to the robot colliding into obstacles. Thus, the development of motion planners that are robust to small errors in the movement of the robot is very practical. The purpose of our project is to implement and analyze the Stochastic Motion Roadmap, a robot motion planner which incorporates uncertainty in the robot's motion.

II: Our Approach

Our project will focus on the Stochastic Motion Roadmap proposed by Alterovitz et al. (2007). Like PRM, the Stochastic Motion Roadmap samples points uniformly at random in the state space as vertices for a roadmap. It models uncertainty by sampling random motions starting at each vertex from a discrete set of possible controls to estimate state transition probabilities between vertices. We assume the robot's random motion comes from a Gaussian noise distribution, which is different for different controls. These state transition probabilities act like the edges of our roadmap, and we can use them to represent P(t | s, u), the probability to get from vertex s to vertex t while executing control u.

Unlike deterministic planners which seek to find a feasible path, the Stochastic Motion Roadmap wants to find an optimal policy π: Q → U, which describes the best action in U to take for each state in Q. The SMR models a Markov decision process, where Q are the roadmap's vertices and U is the discrete set of controls. A reward function represents the rewards given to states in the goal region, where the reward function returns 1 if the state is in the goal region and 0 otherwise. It then uses a dynamic programming method known as value iteration to compute the value function $p_s(i)$, which represents the expected probability of success for the state i to reach the goal region. The optimal policy is extracted from this iterative process by selecting the actions that maximize the chance of success for each state.

Once the optimal policy is obtained, we can query a path from a given start state to the goal region. We will model a steerable needle robot, just like Alterovitz et al., which behaves identically to a Dubins bang-bang car. The robot's motion will be random based on Gaussian noise distribution. There are only two possible control states for the control u used in this problem, moving forward and left (u=0) and and moving forward and right (u=1). Iteratively, we execute the best action for our current state in the path according to the optimal policy. We then conduct experiments to see how often the robot can reach the goal with the Stochastic Motion Roadmap planner and how the probability of success increases with the number of vertices sampled.

III: Experiment Description

We conducted three experiments with Stochastic Motion Roadmaps. First, we generated several interesting environments and then we created SMRs on these environments with n=50,000 sample size and the number of sample transitions m=20. We chose these parameters since the original paper by Alterovitz et al. had good results with them. Once we found the optimal policies for each of these environments, we then simulated many paths and selected the ones that gave the most insight to how SMR behaves.

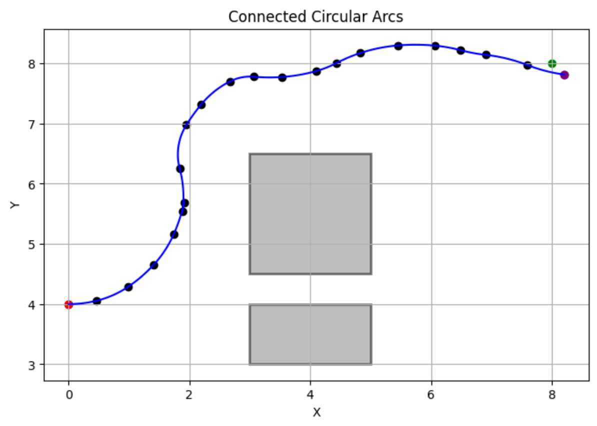
The second experiment we conducted was to change the variance of the robot's random motion in its Gaussian noise distributions. We tested the SMR's behavior and examined the actual and expected probabilities that the start could reach the goal region when the variance is low (low uncertainty), high (high uncertainty), and when the variance differed between controls.

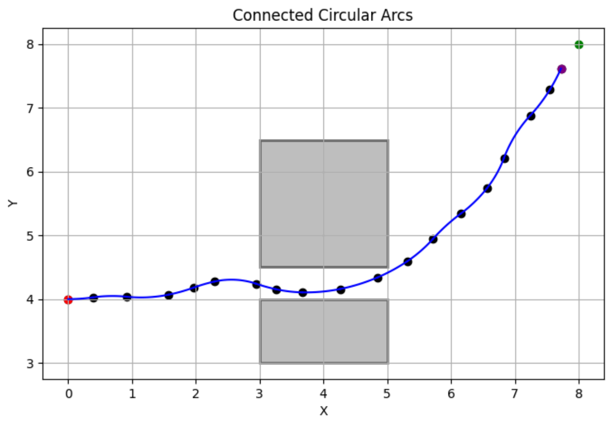
The third experiment we conducted was to examine how sample size affects the difference between the actual and expected probabilities of success. For four different sample sizes (1e3, 1e4, 1e5, 1e6), we generated an SMR on environment 1 and simulated 1000 paths using the SMR's optimal policy. To estimate the actual probability of success, we tallied up the number of collision-free paths that reached the goal and divided them by 1000. We then computed their difference with the expected probability of success obtained from value iteration. Finally, we repeated these steps 20 times for each sample size and computed the mean and standard deviation of the differences between the actual and expected probabilities of success. We use the sample means as point estimators and graph them on a logarithmic scale for the sample size.

IV: Analysis

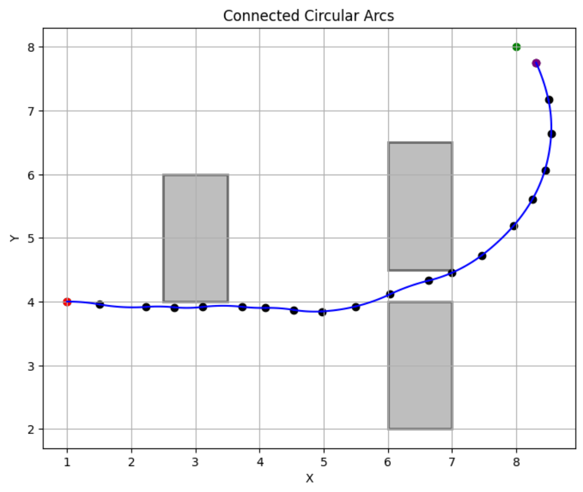
Different Environments

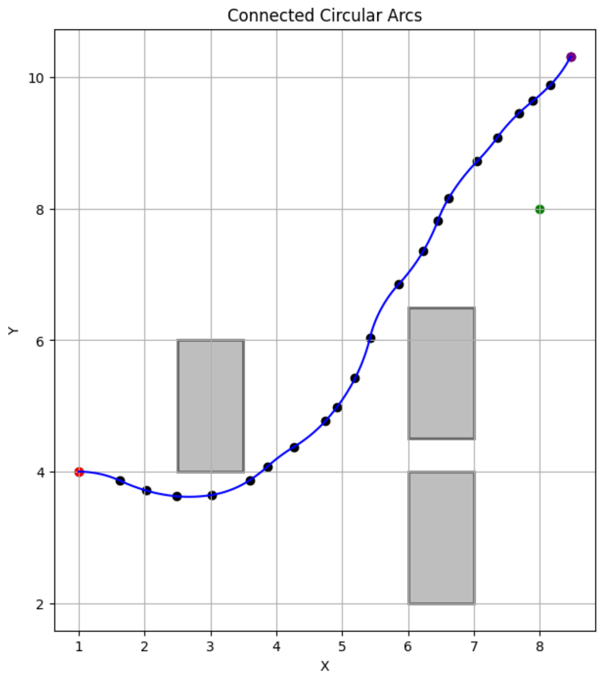
Environment 1

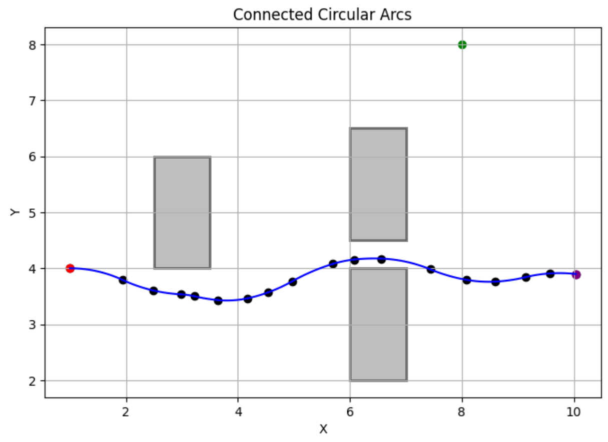


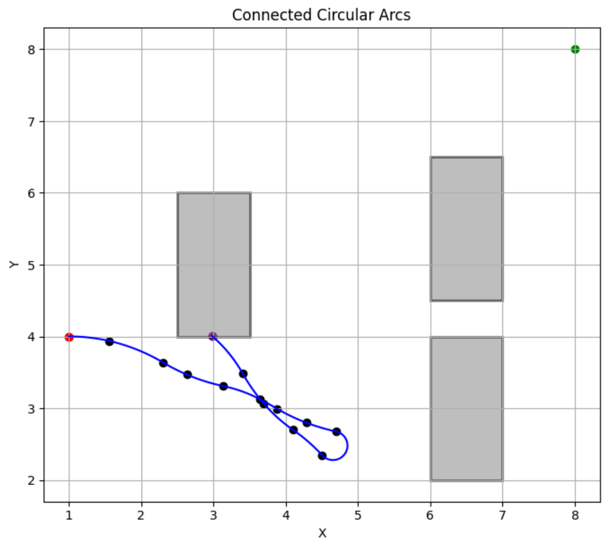


Environment 2

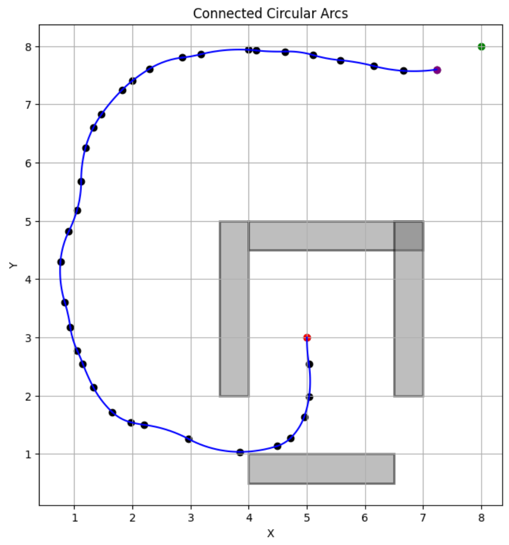


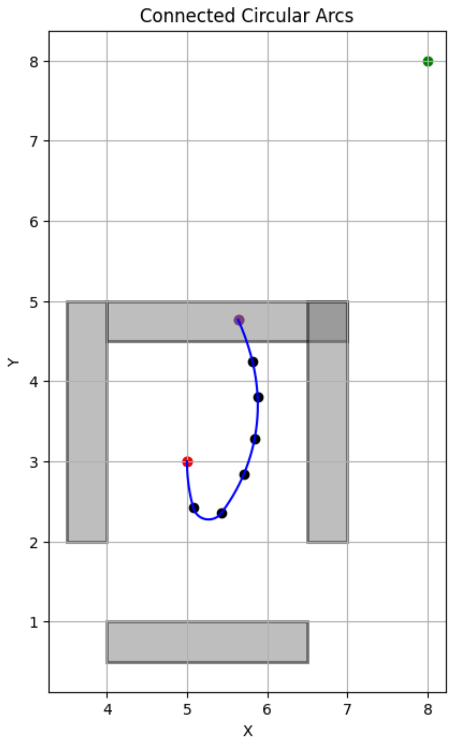






Environment 3





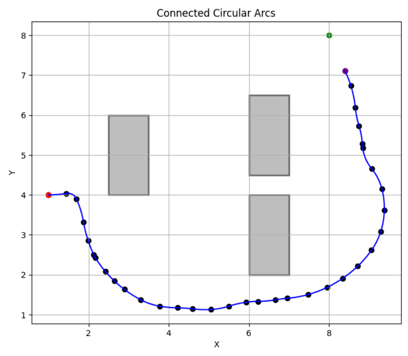
When planning under uncertainty, the algorithm generates different paths each time a plan is executed. This variability is particularly fascinating because, even though it might not always reach the goal, it often discovers multiple solutions due to its inherent randomness.

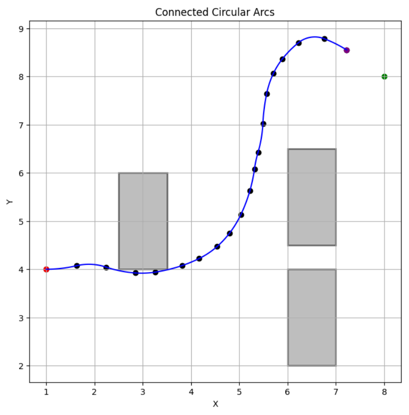
For instance:

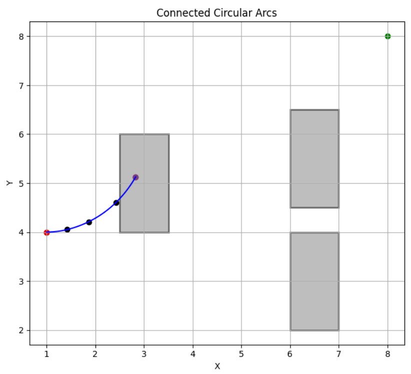
- In Environment 1, the algorithm finds two distinct solutions: navigating around an obstacle to reach the goal or passing through a narrow passage.
- In Environment 2, it explores different strategies, such as taking the larger vertical passage in one instance and the narrow horizontal path in another.
- In the third box environment, the limited space restricts the number of possible solutions. However, the uncertainty allows the algorithm to adapt, eventually finding a way out by taking a wide circular path.

This behavior highlights how planning under uncertainty can leverage randomness to explore diverse and often creative solutions, even in constrained or complex environments.

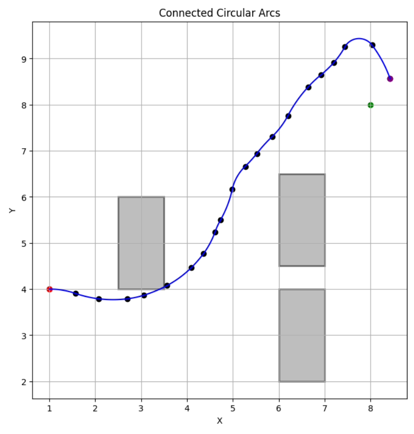
Sigma
High uncertainty (Left 100, Right 100)
Finished in 32 iterations.

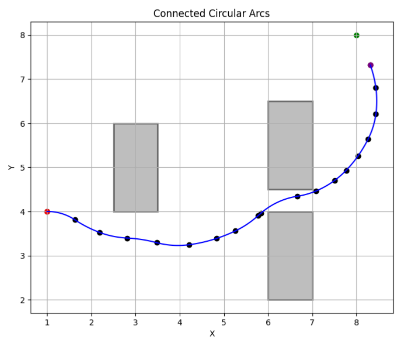


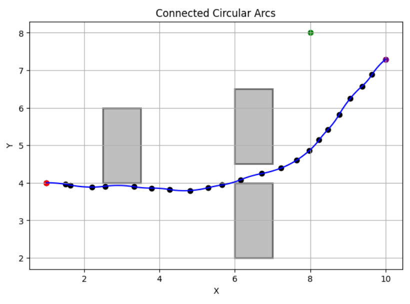




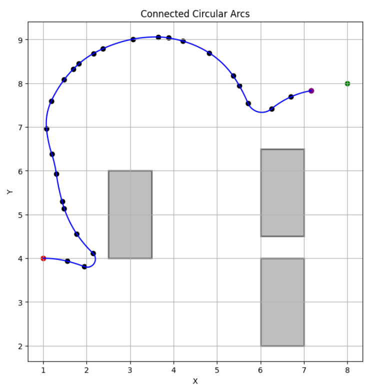
Low uncertainty (Left 0.0001, Right 0.0001)
Finished in 251 iterations.

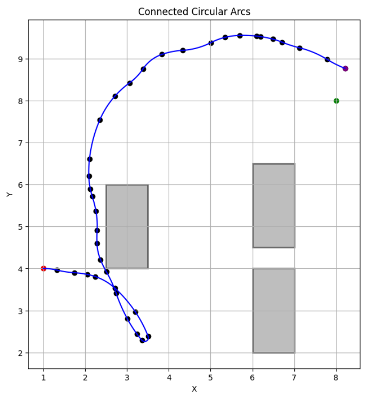


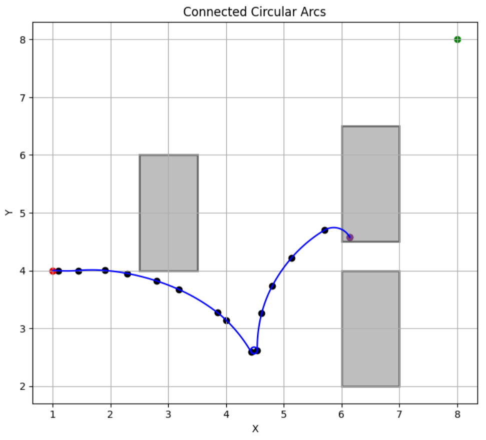




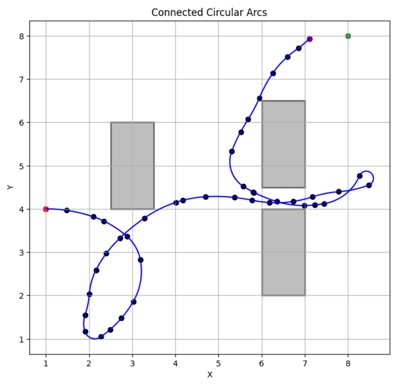
Unbalanced uncertainty(Left 100, Right 0.001)
Finished in 182 iterations

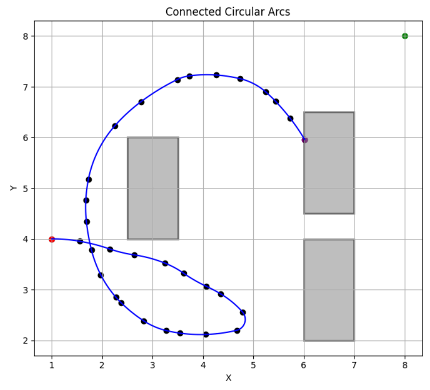


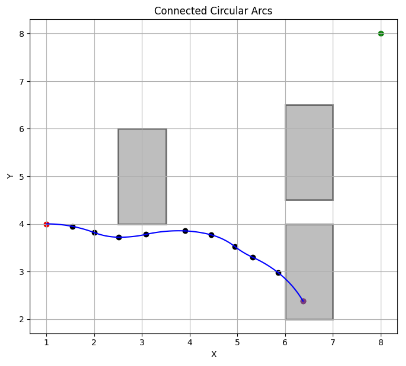




Unbalanced uncertainty (Left 0.0001, Right 100)
Finished in 176 iterations.







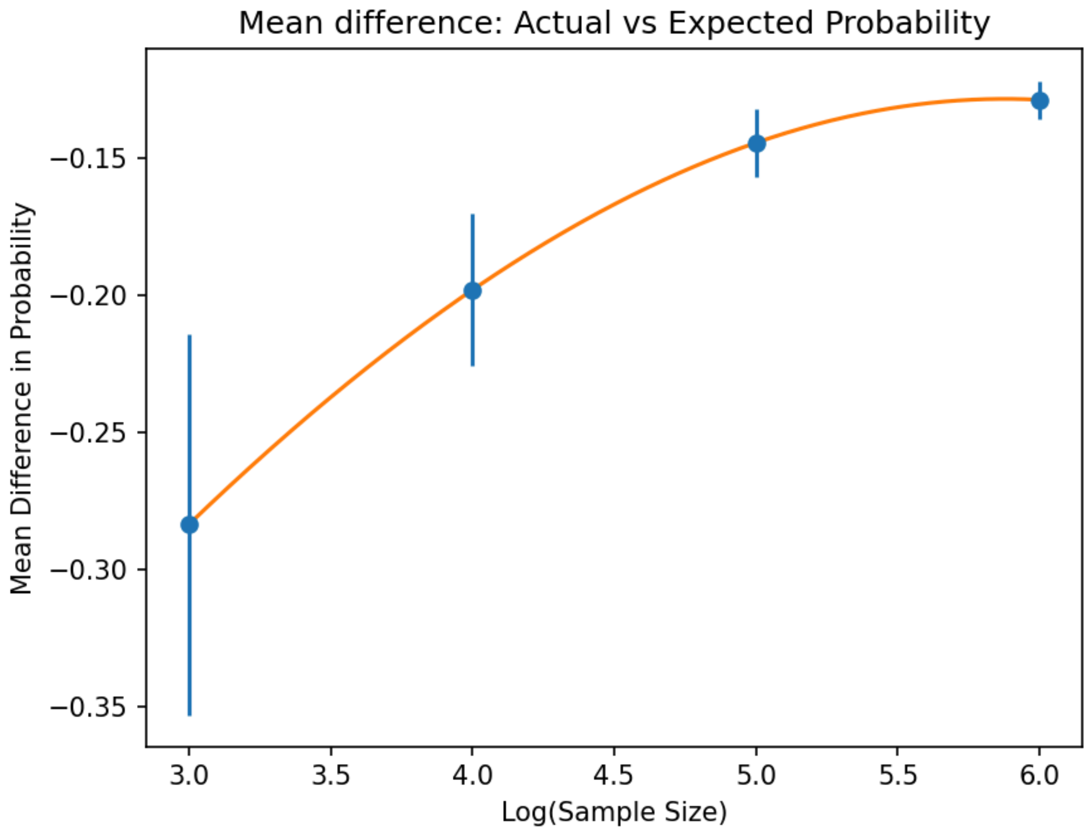
When adjusting the standard deviation to control the needle's turning behavior (left or right), we observed several interesting dynamics:

- High Uncertainty: With greater uncertainty, the task often finishes more efficiently and requires fewer iterations. This level of randomness also reveals solutions that may not have been initially apparent, such as looping under the graph instead of navigating through the narrow passage like most other strategies.
- Low Uncertainty: Lower uncertainty results in a higher iteration cost to find a solution. However, it increases the likelihood of solving the problem and produces paths that appear more optimal and controlled. These paths often align closely with the shortest or most direct route.
- Imbalanced Uncertainty: When one control (e.g., left or right) has high uncertainty and the other has low uncertainty, the resulting behavior becomes particularly intriguing. For example, the path may start to form loops or take on unusual shapes. If the goal is to the left of the start point and the left control has higher uncertainty, the system is more likely to navigate novel paths along the left side, providing additional solutions. Conversely,

when the right control has high uncertainty, the probability of reaching the goal decreases, and the system frequently experiences misdirection or inefficient routes.

These findings underscore the trade-offs between efficiency, solution diversity, and path quality based on the level and distribution of uncertainty. By carefully tuning the uncertainty in each control, it's possible to influence the system's behavior to explore unconventional paths or prioritize more reliable solutions.

Sample Size



The results of our experiment are shown above. We can see that the relationship between the logarithmic sample size and the mean difference in actual and expected probability is a logarithmic growth curve. The mean difference increases with higher sampler sizes but with diminishing returns. Increasing the sample size from 1000 to 10,000 leads to an increase of about 0.08 in the mean difference and increasing the sample size from 10,000 to 100,000 leads to an increase of about 0.05. The standard deviation of the differences, represented as error bars in the graph, also show a trend. As the sample size increases, the standard deviation decreases. This is because with higher sample sizes, the stochastic motion roadmap can better approximate the underlying state-action space. It is also noteworthy that the mean difference is always negative and it seems to plateau at around -0.14. This means that the actual probability for the start to reach the goal is consistently smaller than the expected probability calculated from value iteration. We expected this difference to be smaller and we suspect the problem lies in how we implemented path simulation.

Conclusion

In conclusion, planning under uncertainty introduces variability that can uncover diverse and creative solutions, often revealing paths not initially considered. High uncertainty enhances exploration, reduces iteration costs, and discovers novel solutions, such as unconventional routes like looping under obstacles. Low uncertainty provides more optimal and reliable paths but at the cost of increased iterations. Imbalanced uncertainty creates unique behaviors, influencing path diversity and success rates depending on the control's uncertainty level. Across different environments, the system adapts to constraints, such as navigating narrow passages, taking wide circular paths, or finding unexpected routes. These findings highlight the importance of tuning uncertainty to balance exploration, efficiency, and adaptability in diverse and complex navigation tasks.

The analysis of sample size effects reveals that increasing sample sizes improves the approximation of the state-action space, with diminishing returns as size grows. Larger samples reduce the variability in the difference between actual and expected probabilities, which plateaus around -0.14, indicating consistent underestimation of success probabilities. This highlights the need for further refinement in simulation methods to align actual performance with theoretical expectations.

V: Task Reflection

Task	Difficulty (1 to 10)
Deliverable #1: Implement SMR and simulate in many environments.	7/10. The algorithm itself is easy to understand, but the implementation of it was time-consuming. Writing optimal code was important in making sure we could get experimental results in a reasonable amount of time. There were multiple parts to implement for the SMR algorithm, like the learning phase, query phase, and path simulation. We also made sure to write multiple scripts for visualization to ensure correctness. Once the algorithm was implemented, testing the behavior of the SMR planner on different algorithms was not too difficult.

Deliverable #2: Experimenting with sample size.	4/10. Not too difficult once the SMR algorithm was implemented. The results were interesting and only required a little bit of Python code to graph the results.