Matthew Zheng
HW 2 Report

The algorithm implemented in my work utilizes both Model-based and Model-free Reinforcement Learning. In both algorithms, the discount factor is used to decrement the changes from the past when being used to update the new utility values. Epsilon is used to determine when the learning will end; specifically, the maximum change in either transition probability or utility value is recorded per iteration and then compared against epsilon. When this change is less than epsilon, the learning is terminated. To consider exploration vs. exploitation, the number of times an action is taken from a given state is recorded. While this count is under a defined threshold, the action is considered a possibility and added to a list of potential actions to take. After evaluating every action, those that produce the minimum utility (as we are trying to minimize strokes) are chosen from randomly and the state is moved forward. The count is then updated.

1. How did changing the initial value for exploration (starting close to 1 versus starting closer to 0) change the resulting computed policy?

   Given my implementation, this took a slightly different manifestation; however, the effects were the same. With a higher coefficient for exploration allowing for a greater amount of exploration did allow for the algorithm to more accurately portray the system. This can especially be seen in the Model-Based Learning as the probabilities reported more accurately reflect the real probabilities. However, this only worked up to a certain point. Increasing the coefficient would generally result in better results, but would begin to plateau once higher numbers were reached.

2. How did you decide when to stop learning?

   Through the use of the provided epsilon value, the algorithm's end condition was based on the maximum experienced change between every time the algorithm started over. Each time the algorithm reset the process to the Fairway, it would check what the largest change that had occurred since the last reset (in the case of Model-Based it would check the greatest delta in probability, in the case of Model-Free the greatest delta in Q-Value.)

3. How did changing the discount value (starting close to 1 versus starting closer to 0) change the resulting policy?

   As expected, the lower the discount value the fast the algorithm would terminate, but less accurately. Again looking at the Model-Based approach, it could be seen that with discount values close to 1 the algorithm would find probabilities with errors averaging 0.05 but this quickly increased as the discount value fell; however, the number of iterations would also decrease resulting in a faster run time. As with the value for exploration, this eventually plateaued as well.

4. How did changing the value for epsilon change the resulting policy?

   Epsilon controlling when the learning stopped meant that the lower the value of epsilon the longer the run time; however, interestingly enough it appears that within a certain margin the changes to epsilon would not change much. As epsilon decreased this

generally improved the accuracy of the policy, but would generally plateau immensely for despite orders of magnitude decrease past the 3rd or 4th order.