Rokos report – Alex Rutherford

This summer I spent 8 weeks (remotely) interning at the Oxford Robotics Institute with the Goal-Oriented Long-Lived Systems (GOALS) Lab. I was tasked with combining two existing algorithms, rapidly exploring random trees (RRT) and Monte Carlo Tree Search (MCTS) to create a new hybrid algorithm for navigating across an uncertain domain.

I began by understanding and implementing both of the existing algorithms. They share many similarities but have been designed with different applications in mind. MCTS is used to explore typically discrete state spaces with high branching factors (i.e. many possible actions at each state), random action outcomes and long planning horizons. Perhaps most famously, it was used by Google as a basis for their AlphaGo program which conquered the game Go a few years ago.

Meanwhile, RRT is used to quickly explore a continuous space, rapidly finding paths to a goal region. The method RRT uses to explore is incredibly efficient and it can be biased to explore new areas. These qualities make it ideal for movement planning tasks.

Both of these algorithms were implemented in Python, with the task of exploring a simple two-dimensional state space to find a goal. Their performance can be seen in the figure on the right (MCTS here is labelled as UCT). RRT took random samples from the search space, rejecting any sample that was within an obstacle and then extending the tree towards the new sample. On the other hand, for MCTS the state space was in essence discretised into a grid of nodes the MCTS chose actions which moved the robot between these nodes. ‘RRT star’ is a variant of RRT which finds the optimal path to the goal instead of just any path while ‘UCT ec’ is identical to MCTS but is biased towards the goal. Result across this domain were in line with expectations, RRT is quicker but UCT consistently produces optimal paths.

The search domain was then extended to include uncertainty. To achieve this, I used particle RRT (pRRT), a variant of RRT which samples many times at each extension of the tree, allowing for stochastic outcomes. This algorithm maintains a tree structure which becomes a disadvantage as the planning horizon increases as the probability of reaching nodes in the path quickly diminishes. It also biases the growth of the tree towards path that have a higher likelihood of being followed by the robot. This works well in some domains but in others with large trees the method used for biasing the growth results in many possible extensions being rejected.

To overcome these issues, I extended particle RRT with a graph structure to result more robust plans with a higher probability of reaching the goal. The resulting, novel, algorithm has two stages, a search phase and a dynamic programming phase. The search phase
creates a graph and then the dynamic programming finds the optimal path through the graph to the
goal. The search phase closely follows, rapidly exploring random graphs, a variant of RRT. A random
point is sampled from the search domain and then as in pRRT extensions toward this point are
sampled multiple times. The different outcomes of the samples, due to the stochastic nature of the
domain, are observed and then, like pRRT, they are clustered to form the new nodes for the tree.
This clustering groups similar samples to allow for substantively different paths that the robot could
take depending on the action outcome to be represented in the graph.

The progression from pRRT comes from adding additional connections to the new nodes. The
different can be seen in the Figure above, pRRT keeps a tree structure while pRRG has a graph
structure. This graph structure allows the dynamic programming phase to be run. In my
implementation I used Nested Value Iteration which optimised for the probability to reach the goal
firstly and then the distance of the path.

For my test domains I first took inspiration from the pRRT paper and used a simple domain where
the coefficient of friction was uncertain. I then used a pre-trained Gaussian process model based on
data from the Copernicus Marine Environment Monitor Service to create a domain where the robot
moved ‘underwater’ and the currents it experienced followed the GP and so action outcomes were
stochastic.

Overall, I really enjoyed my internship and it particularly exciting to learn about a branch of
engineering and maths that I knew nothing about before starting. Joining the weekly GOALS lab
reading group was also really interesting as each week a new paper was discussed with further
exposed me to new parts of robotics, beyond just what my project was focused on.