Introduction to Robotics
Assignment #6

Caus Danu 7014833 | Massimo Innocentini 7016313 | Daniel Bremer 6834136

Task 6.1 (8 points) Configuration Space - Programming Task:

6.1.1 (6 points):

We performed this experiment with 2 different resolution settings: 1 degree (see figure 1) and 0.1 degree (see figure 2). Since at this stage, the program is much simpler than for the other subsequent tasks, the 0.1 degree case finishes in maximum 10 minutes on an average laptop. In the process of building the C-Space, we have shrunk the robot to a point and inflated the obstacle borders/deflated the start and goal areas correspondingly (so that the TCP overlaps fully with those areas).

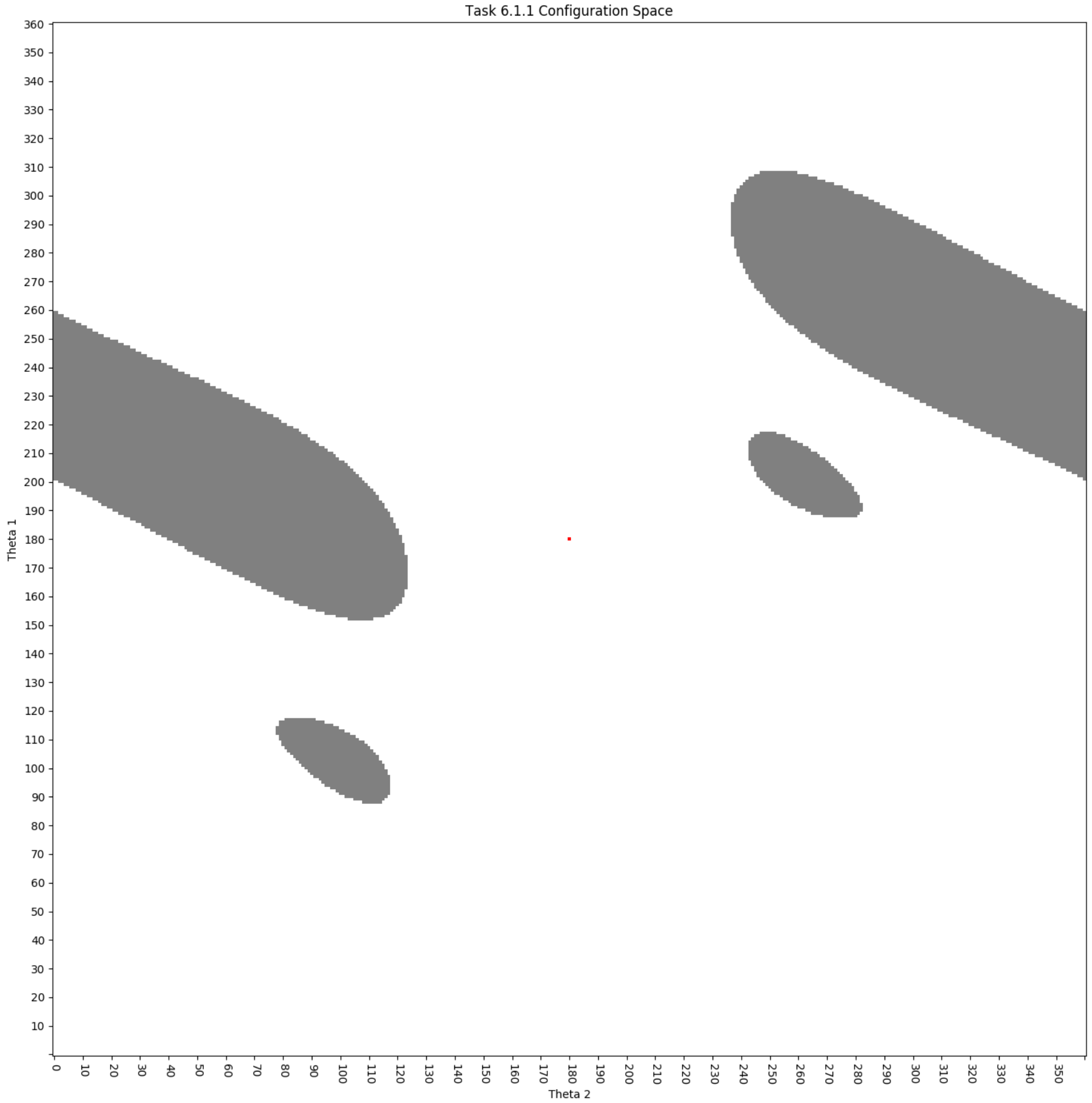

Figure 1: Configuration Space with 1 degree resolution. White: free space, grey: obstacle

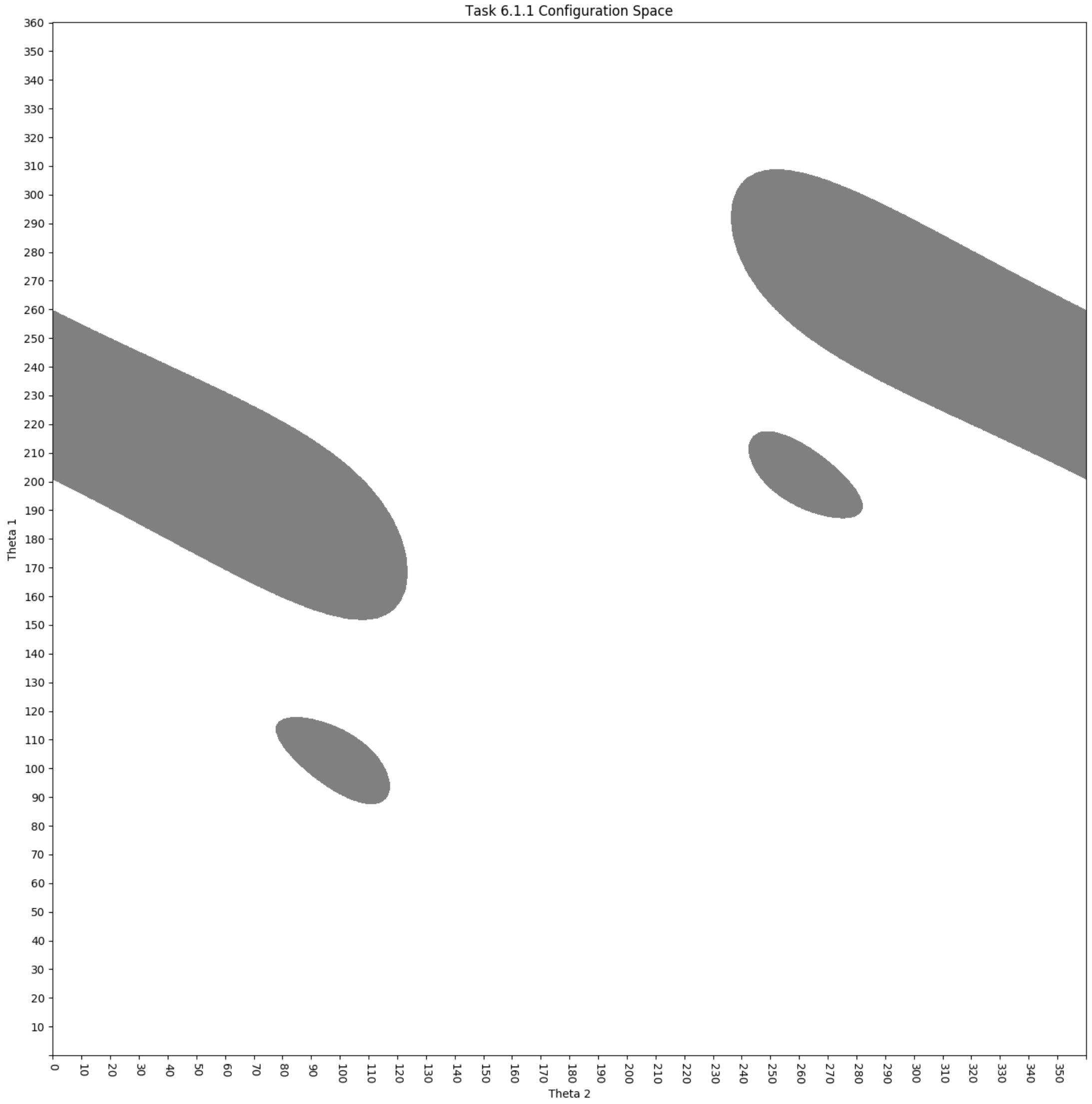

Figure 2: Configuration Space with 0.1 degree resolution. White: free space, grey: obstacle

6.1.2 (2 points):

For this task, we extend the previous one by using a graph data structure in order to store free nodes, while building the configuration space itself. Once we finish building the C-Space, we also have a graph on which we can search for possible paths between the start and goal areas and choose the shortest path.

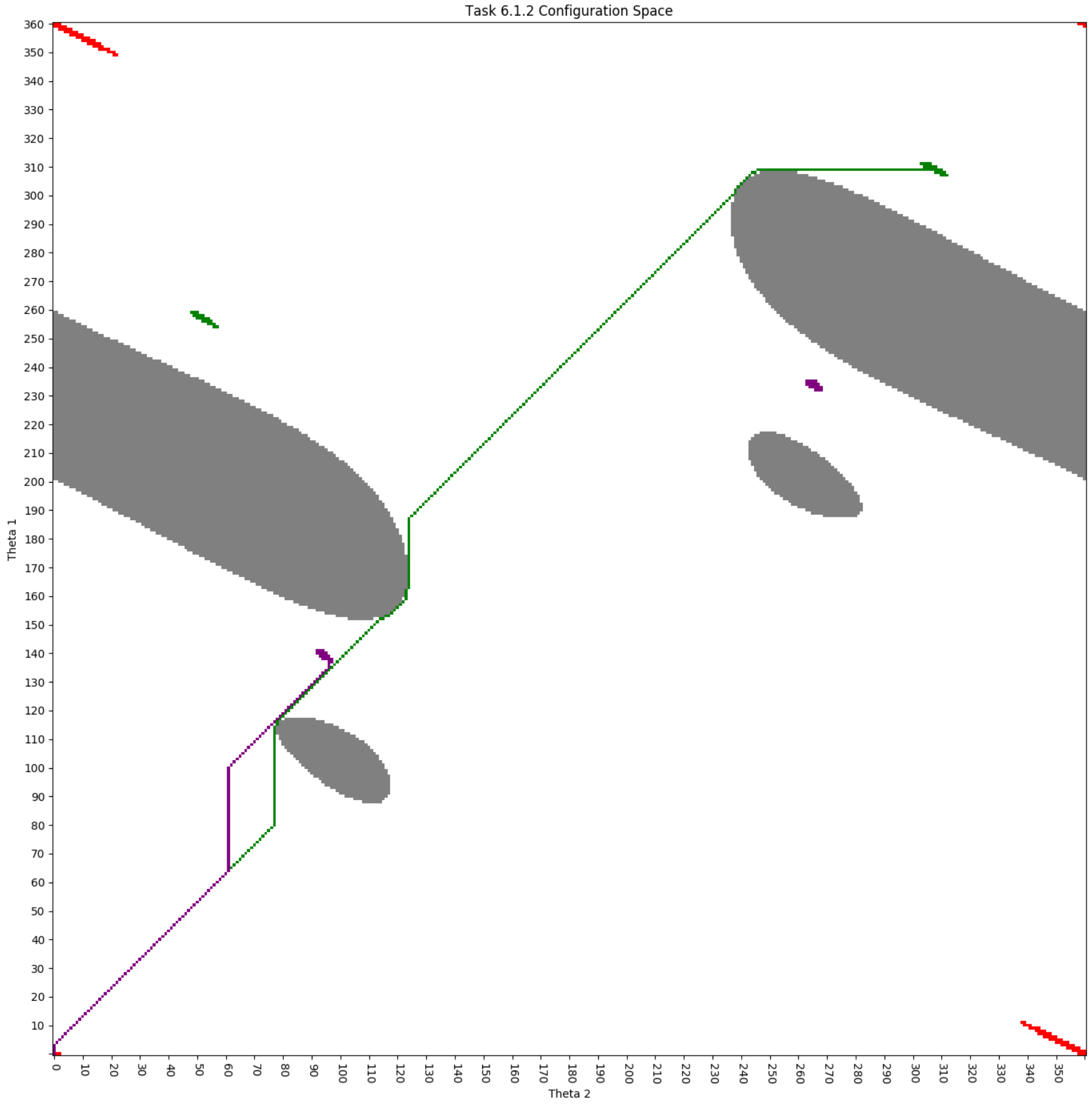

Figure 3: Configuration Space with drawn paths from start to the 2 goal areas. Red blob: start area, green blob: goal 1, purple blob: goal 2

To draw the workspace presented in figure 4, we've created helper functions that manipulate a 1000 by 1000 matrix: draw_circle(...), draw_square(...), and draw_path(...)

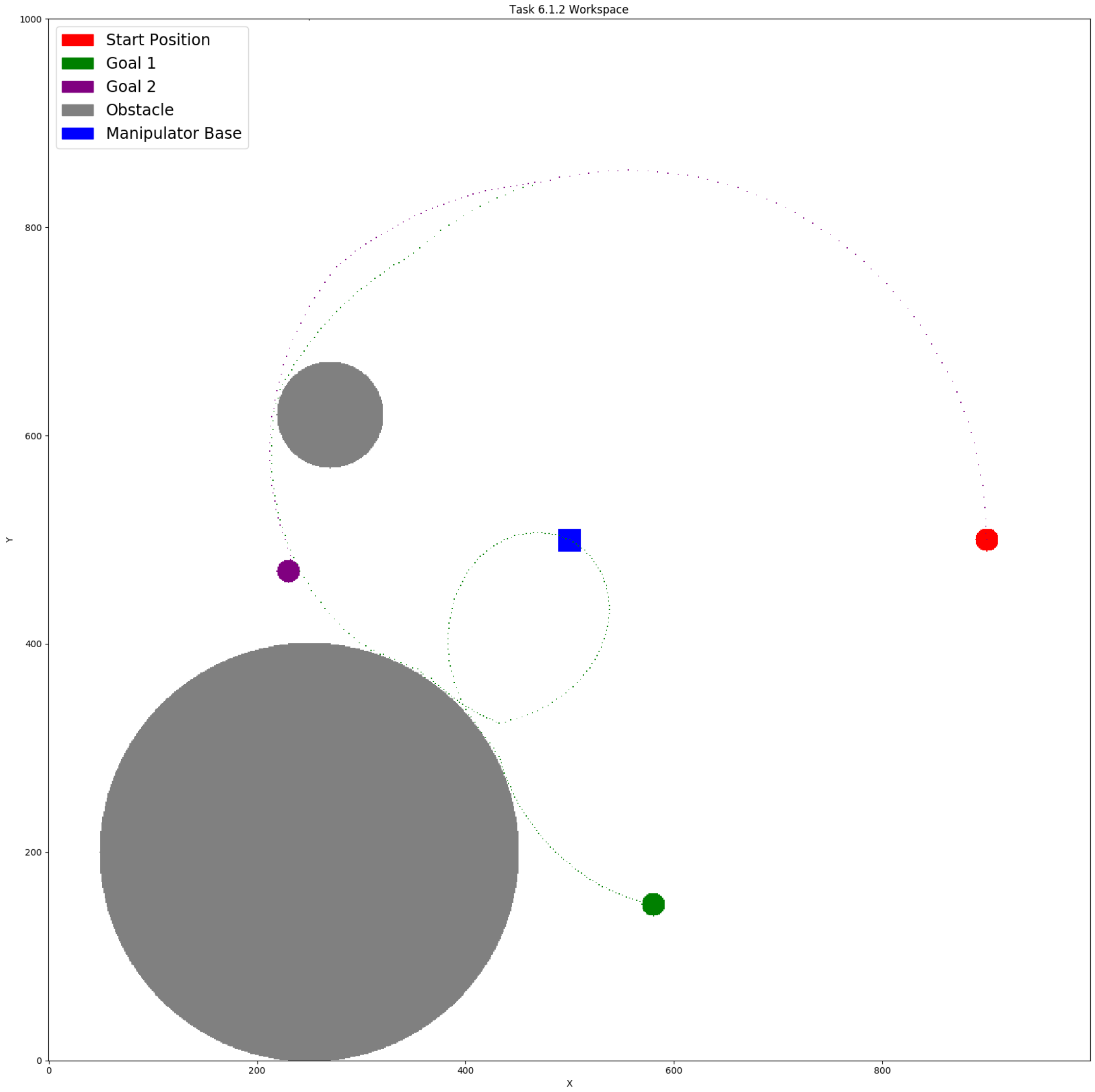

Figure 4: Workspace with drawn paths from start to the 2 goal areas

If we allow the 2 links to rotate with positive, as well as negative angles, we get a different shortest path solution presented in figure 5. We implement this using bidirectional edges between the graph nodes.

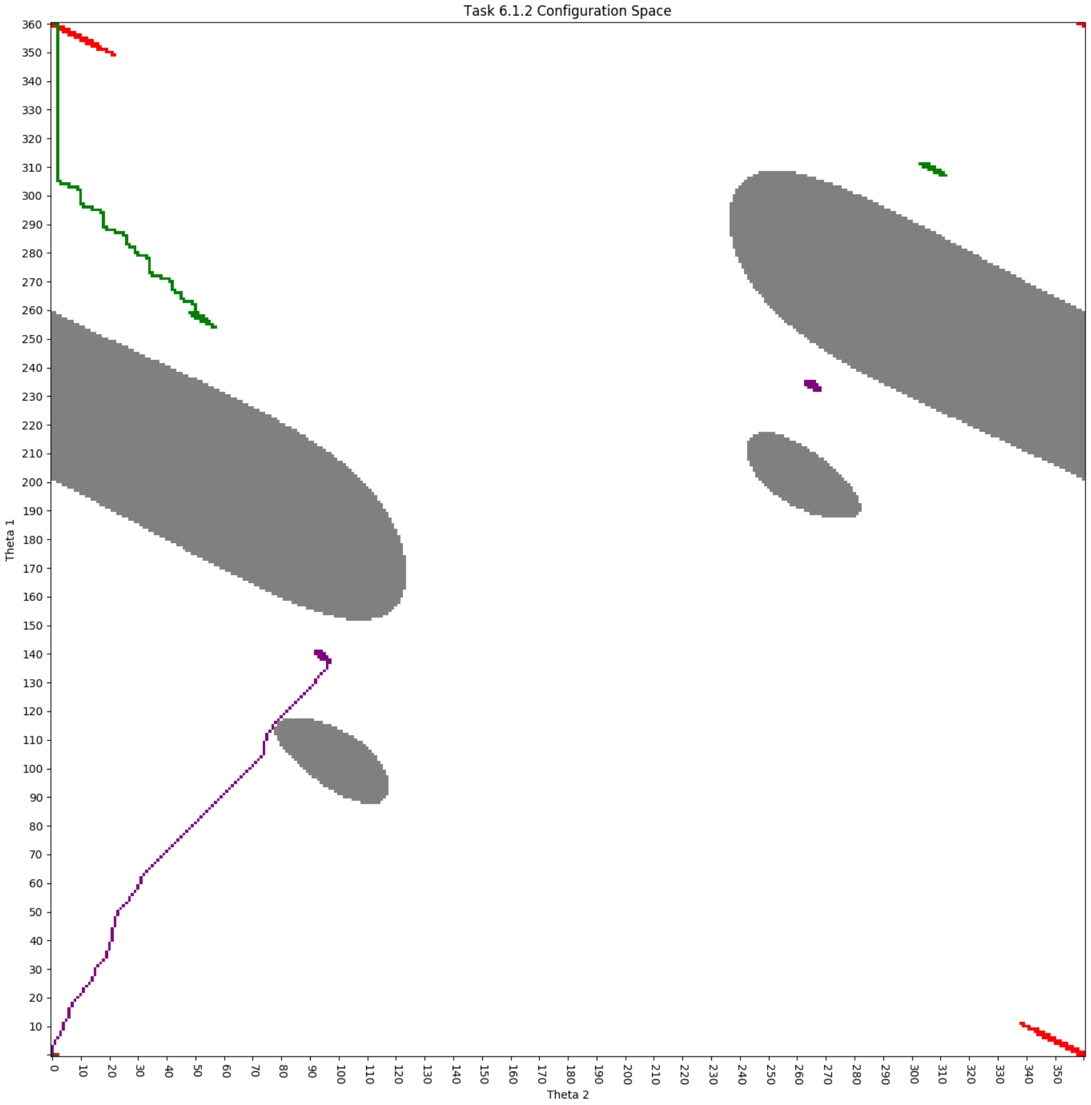

Figure 5: Configuration Space with drawn paths from start to the 2 goal areas. Links are allowed to rotate in both directions. Red blob: start area, green blob: goal 1, purple blob: goal 2

Task 6.2 (7 points) Configuration Space with Links - Programming Task:

The application can build a configuration space of 1 degree resolution and search for a path in approximately 10 minutes (on a PC with i7 processor and 8 GB of RAM). There is no path found for a resolution of 1 degree. The same negative outcome is achieved for a 0.5 degree resolution. For a 0.1 degree resolution, the program should run a lot more time and we did not perform the experiment due to time constraints. Since the exercise asked us to use a grid method instead of a probabilistic method, a potential solution to speed up the application is to use multi-resolution grids. In other words, create a coarse 1 degree grid in large free spaces, and finer 0.1 degree grids in other areas of interest. This might indeed help, since, as we can see in figure 6, the goal in purple is on the edge of an obstacle. Another point to note is that we are trying to exhaustively find the shortest path, instead of any possible path. That is another possible explanation for why this might be running so long.

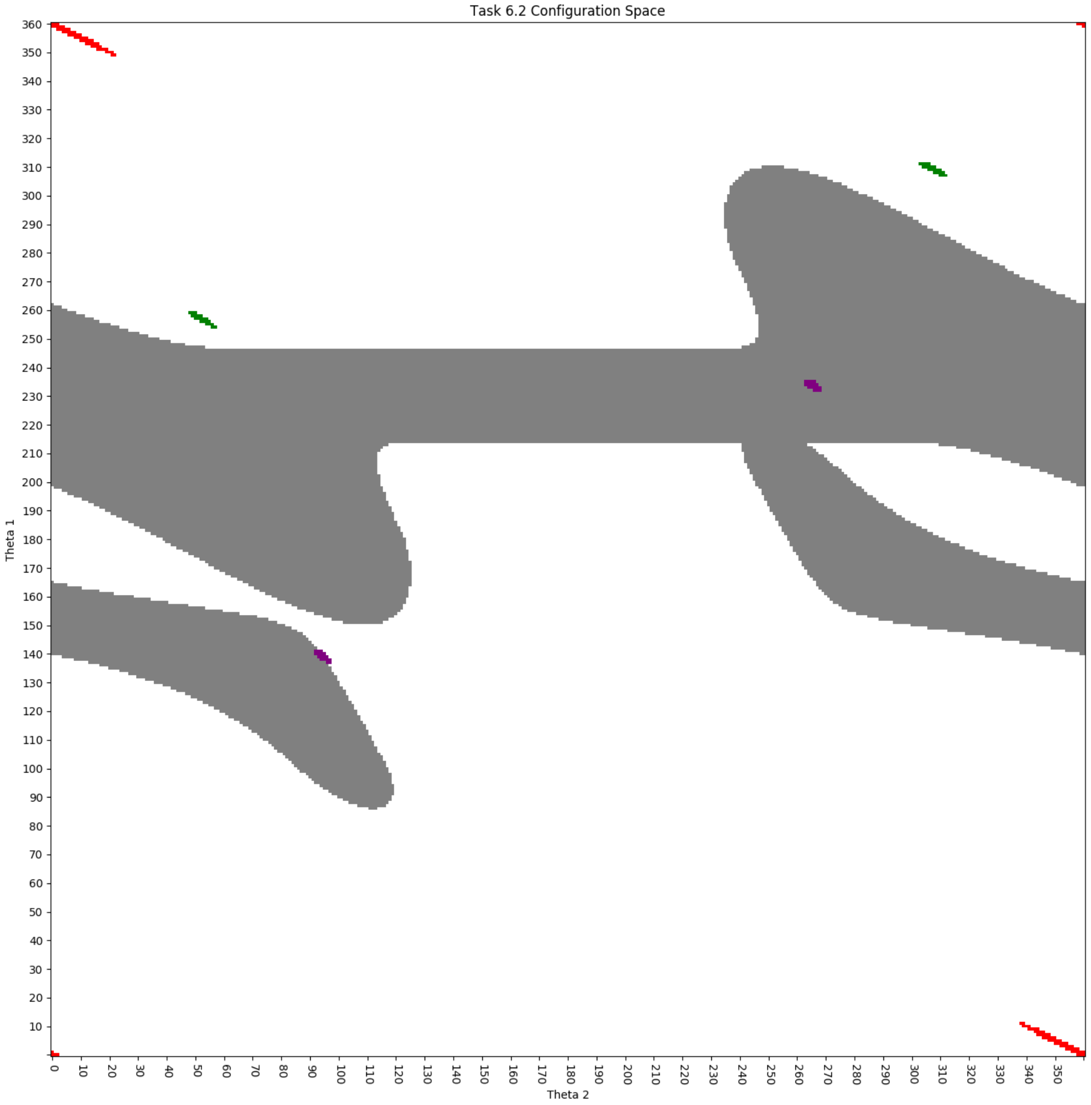

Figure 6: Configuration Space with manipulator links. Red blob: start area, green blob: goal 1, purple blob: goal 2

We also want to stress that in this task we originally assumed only **POSITIVE** rotation angles, and this significantly reduces the dynamic range of the manipulator when it tries to avoid obstacles. If we allow negative rotations, then both goals are very easy to reach. We can convince ourselves of this by extending the code to use bidirectional links between the graph nodes. If we do that and run for a resolution of 1 degree, we get the paths shown in figure 7

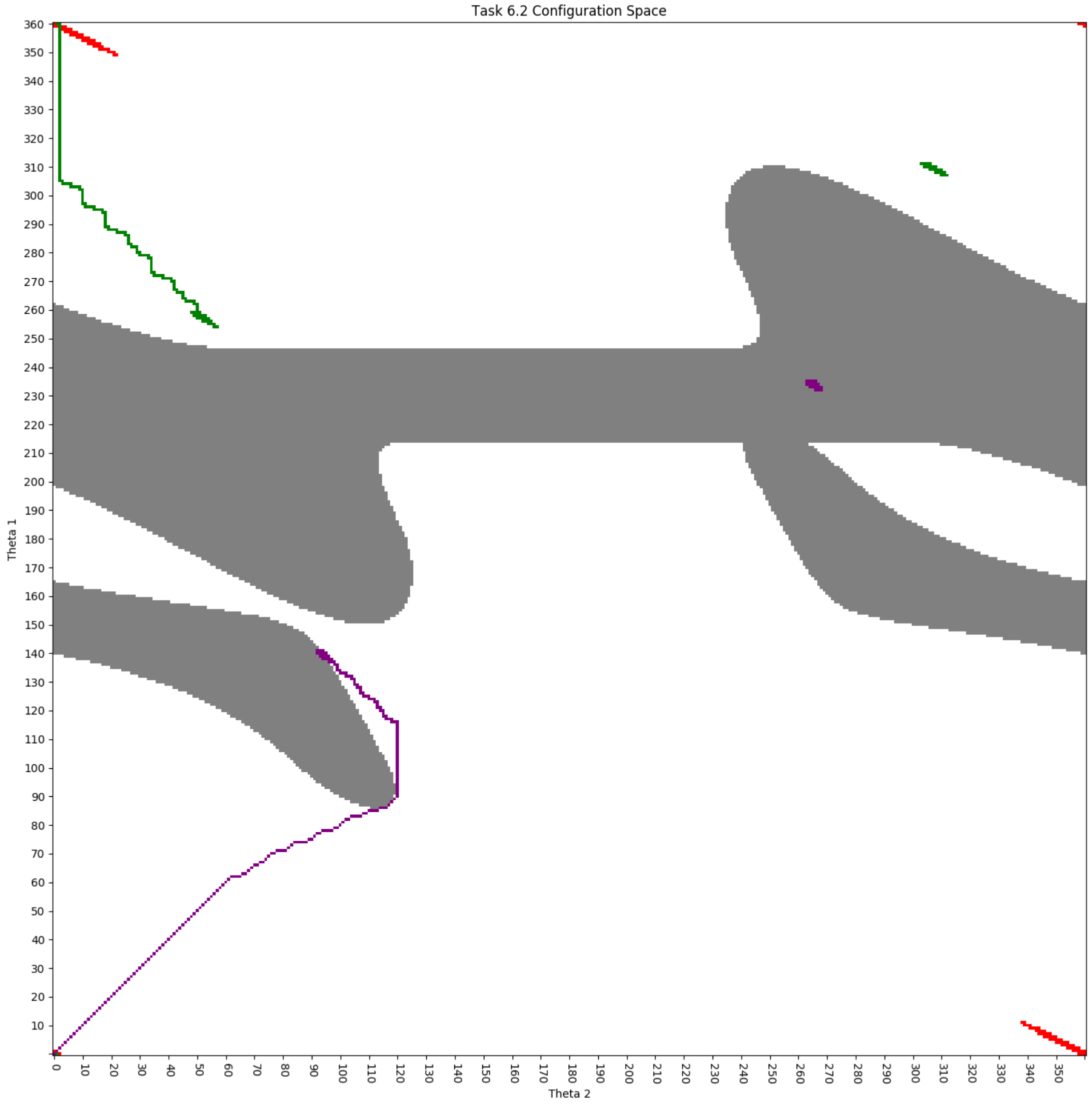

Figure 7: Configuration Space with manipulator links. Paths were found using bidirectional graph edges. Red blob: start area, green blob: goal 1, purple blob: goal 2.

Task 6.3 (5 points) Arbitrary C-Obstacles - Programming Task:

For this task, the workspace looks as drawn in figure 8. The corresponding C-Space is presented in figure 9. Just like in the previous task, we ran the experiment for 1 degree resolution and 0.5 degree subsequently, which resulted in no path found. Just as previously, we are trying to find the shortest path, not just any path.

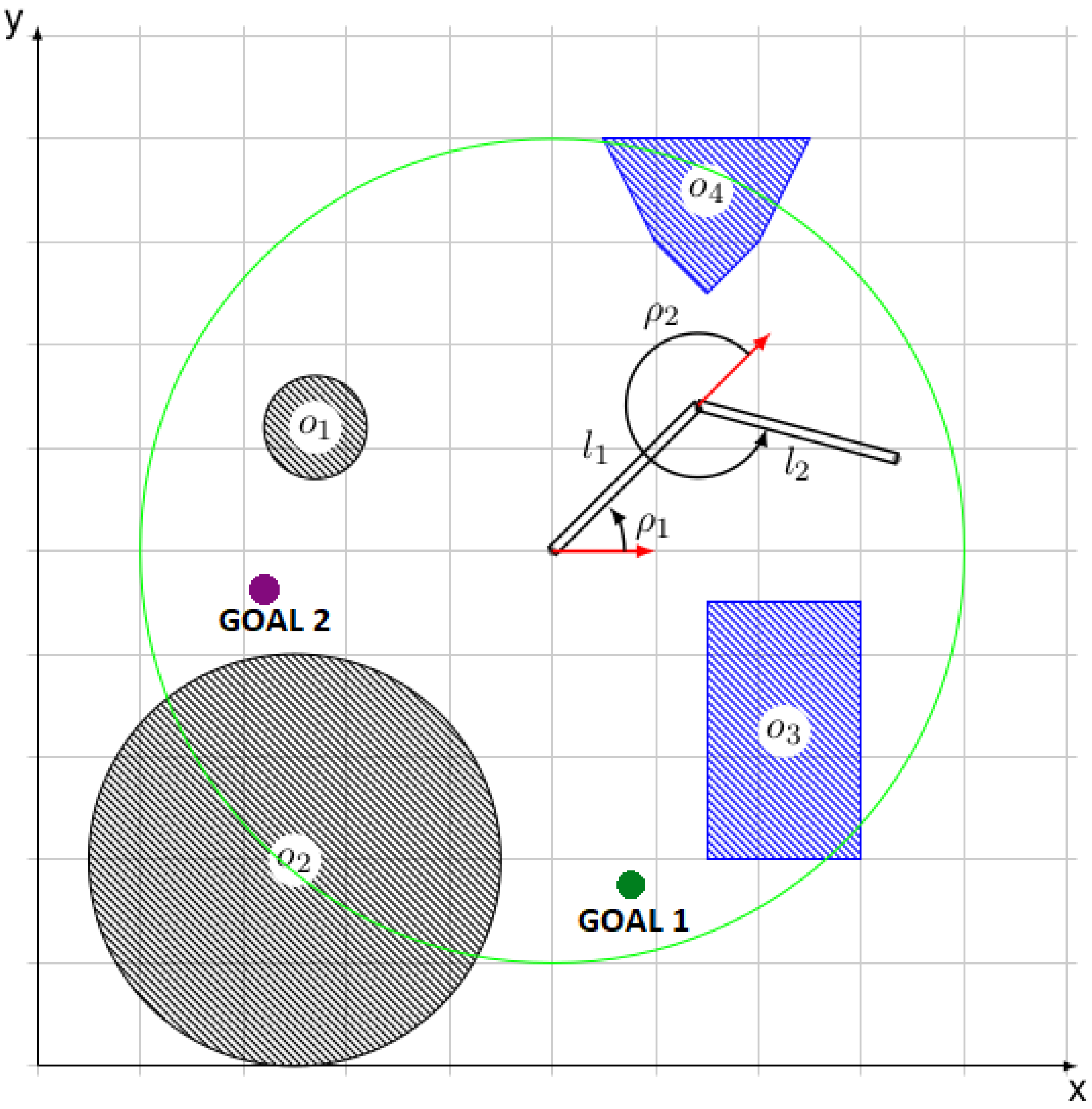

Figure 8: Workspace with more obstacles: O_1 to O_4

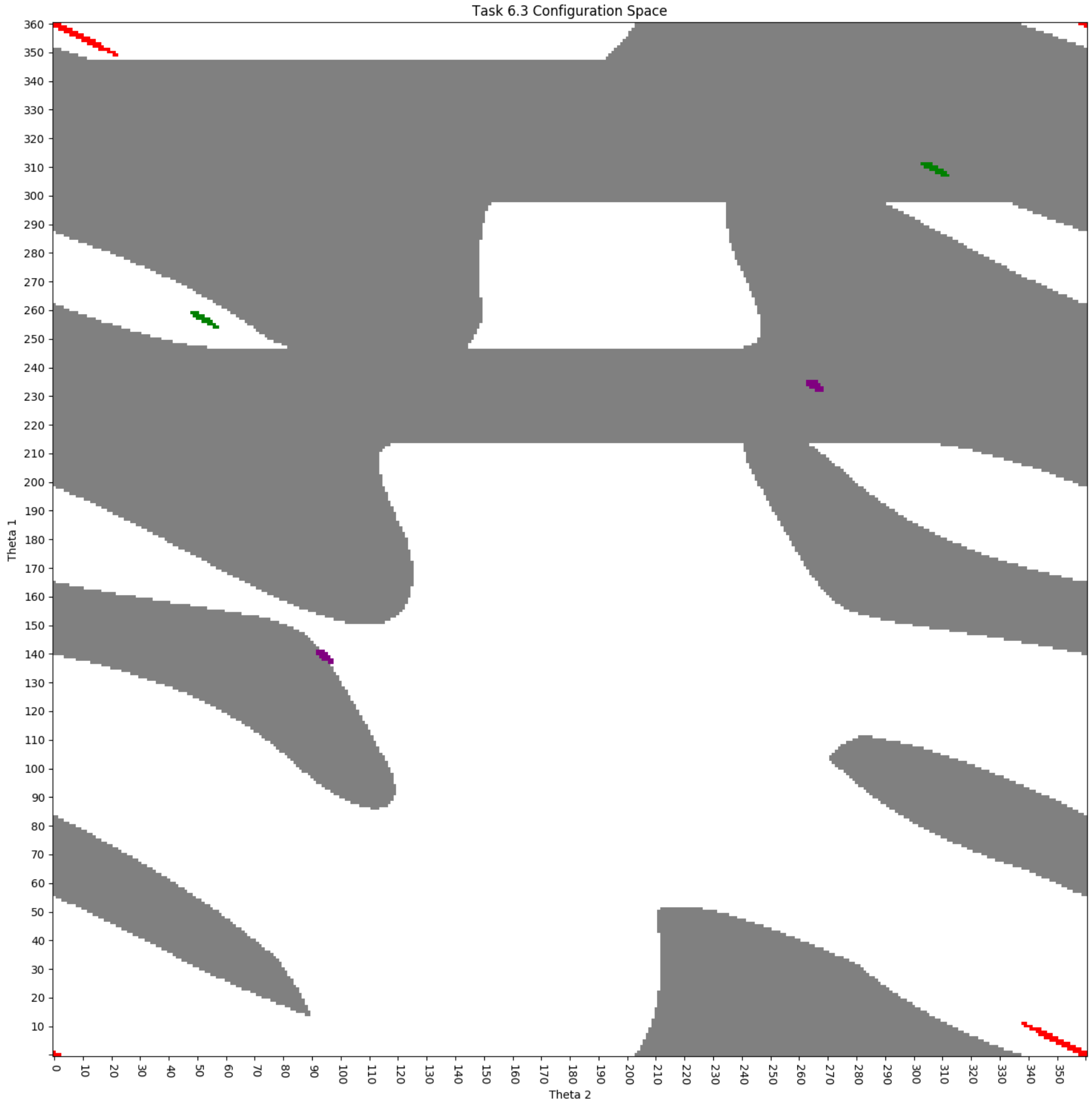

Figure 9: Configuration Space with more obstacles. Red blob: start area, green blob: goal 1, purple blob: goal 2

Like in the previous task, we assumed originally the constraint of having only **POSITIVE** rotations. The difference is that, even with negative rotations, this task looks like it would not have a solution for goal 1. As can be seen in figure 8, it is very hard for the manipulator to avoid obstacle 3, no matter if rotating positively or negatively. It is possible to reach goal 2 however if we use bidirectional angles. Adding edges to the graph in both directions will result in the shortest path solution presented in figure 10. This experiment was performed using a resolution of 1 degree.

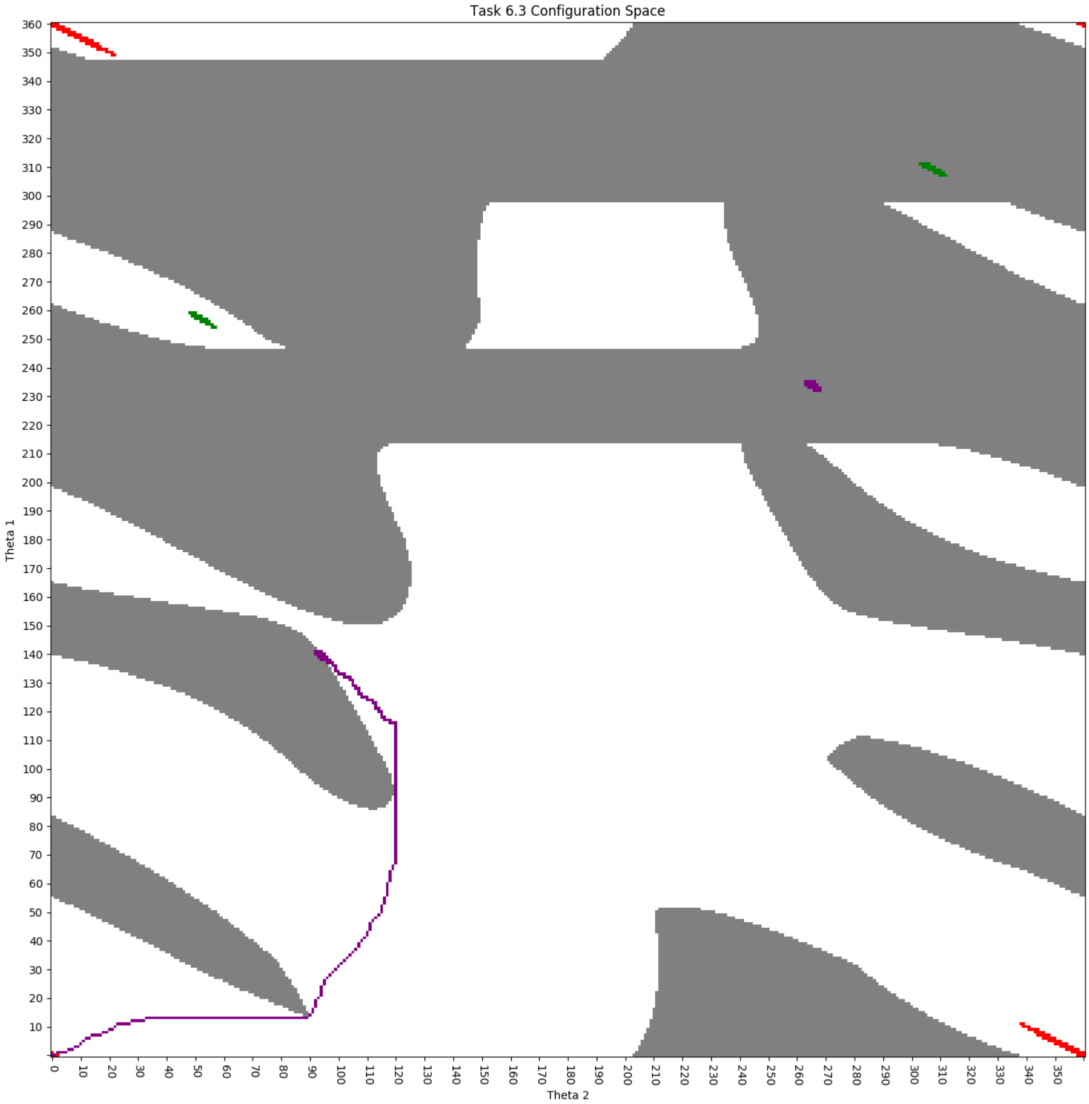

Figure 10: Configuration Space with more obstacles. Path to goal 2 was found using bidirectional graph edges. Red blob: start area, green blob: goal 1, purple blob: goal 2