COMP0037

Report

Planning in Uncertain Worlds

Group AS

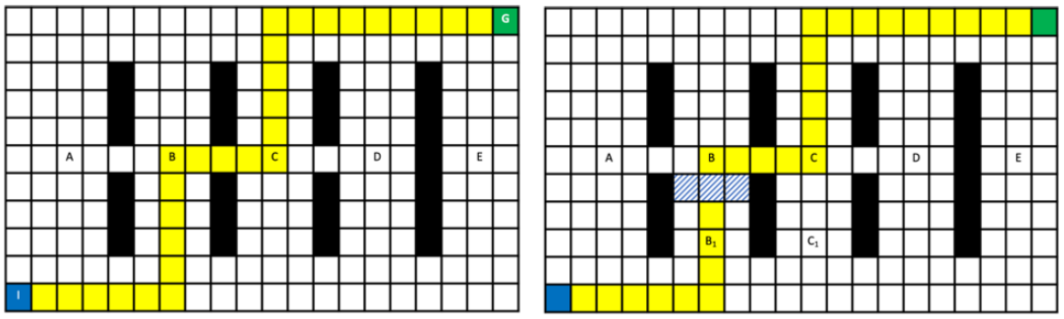
Student Name	**Student number**
Arundathi Shaji Shanthini	16018351
Dmitry Leyko	16021440
Tharmetharan Balendran	17011729

Department:	Department of Electronic and Electrical Engineering
Submission Date:	28th of April 2020

Contents

1 Decision Re-Plan Policy

1.1 Policy Selection when Obstacle is Observed



(a) The original planned path form I to G going through Aisle B and C.

(b) An obstacle in aisle B obstructs the planned path of the robot.

Figure 1: Illustration of case where robot observes an obstruction to it's planned path.

The scenario that we will be analysing is the case shown in Fig. 1a. The robot is required to go from a cell I to a cell G. These cells are marked blue and green in Fig. 1a respectively. The figure also shows the original planned path that the robot computed going down aisle B. However, once the robot turns into aisle B it observes that the aisle is blocked. This observation is done at the point when the robot reaches the cell labelled B_1. At this point the robot can either decide to wait until the obstruction clears or it can re-plan a path. Once the robot observes the obstacle, the time the robot must wait for the obstacle to clear may be represented by the expression in Eq. 1.

$$T = \frac{0.5}{\lambda_B} + \widetilde{T} \tag{1}$$

The wait time is dependent on λ_B and a random variable $\widetilde{T}$. The random variable $\widetilde{T}$ is sampled from a exponential distribution with a rate parameter of $0.5\lambda_B$. The probability density function (PDF) for $\widetilde{T}$ is shown in Eq. 2.

$$f(t) = \begin{cases} 0.5\lambda_B e^{-0.5\lambda_B t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2}$$

As previously mentioned, the robot has two options to choose from: to wait for the obstacle to clear, or to re-plan and execute the new path. The two are different policies the robot must choose from. We use the symbol π to denote a policy. A policy is a mapping from the world state to an action the robot can execute.

Let us assume that if the robot decides to wait the total path length (number of cells) will be K_1 while if the robot decides to re-plan and execute the total path length will be K_2. We let the quantity K equal to the larger value between K_1 and K_2. Now we may write the policy for the robot to wait as π_K^1 and the policy for re-planning as π_K^2. These policies are padded correspondingly to produce actions $\mathbf{u}_K^1$ and $\mathbf{u}_K^1$ that are padded with zero-cost state preserving actions.

To see which policy is better on average, we consider the expected value of the cost function for both cases. The case when policy π_K^1 is chosen is characterized by the inequality shown in Eq. 3.

$$\mathbb{E}\left[L\left(\pi_K^1\right)\right] \leq \mathbb{E}\left[L\left(\pi_K^2\right)\right] \tag{3}$$

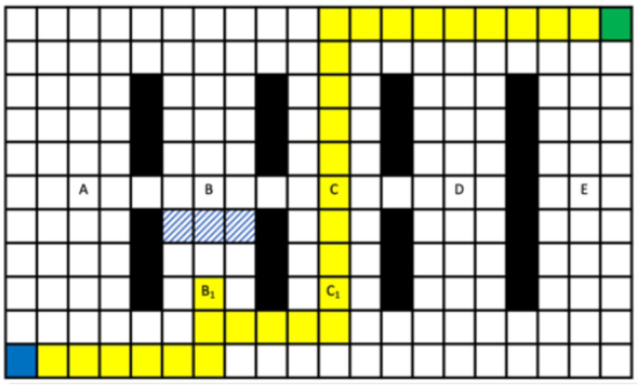


Figure 2: The path for the re-plan policy π_K^2 which bypasses aisle B and goes down aisle C.

We can see from Fig. 1a that the cost of the original planned path is given by the expression in Eq. 4. In this equation, the terms L_{XY} denote the cost of the shortest path between cell X and cell Y. Additionally, the term L_W represents the cost of waiting 1 unit of time.

$$L\left(\pi_K^1\right) = L_{IB_1} + TL_W + L_{B_1B} + L_{BC} + L_{CG} \tag{4}$$

From Fig. 2 which shows the re-planned path, we can also see that the cost of this path is equal to the expression in Eq. 5

$$L\left(\pi_K^2\right) = L_{IB_1} + L_{B_1C_1} + L_{C_1C} + L_{CG} \tag{5}$$

Substituting the expressions in Eq. 4 and Eq. 5 into Eq. 3. We obtain the inequality shown in Eq. 6

$$\begin{aligned}
\mathbb{E}\left[L_{IB_1} + TL_W + L_{B_1B} + L_{BC} + L_{CG}\right] &\leq \mathbb{E}\left[L_{IB_1} + L_{B_1C_1} + L_{C_1C} + L_{CG}\right] \\
L_{IB_1} + \mathbb{E}\left[T\right] L_W + L_{B_1B} + L_{BC} + L_{CG} &\leq L_{IB_1} + L_{B_1C_1} + L_{C_1C} + L_{CG} \\
\mathbb{E}\left[T\right] &\leq \frac{L_{B_1C_1} + L_{C_1C} - L_{B_1B} - L_{BC}}{L_W}
\end{aligned} \tag{6}$$

The quantity $\mathbb{E}\left[T\right]$ is the expected value for the time the robot has to wait for the obstacle to clear. As we know the distribution that the variable is sampled from we can compute the expected value. The expected value for the time taken is given by the expression found in Eq. 7

$$\begin{aligned}
\mathbb{E}\left[T\right] &= \mathbb{E}\left[\frac{0.5}{\lambda_B} + \widetilde{T}\right] \\
&= \frac{0.5}{\lambda_B} + \mathbb{E}\left[\widetilde{T}\right] \\
&= \frac{0.5}{\lambda_B} + \int_0^\infty t0.5\lambda_B e^{0.5\lambda_B t} dt \\
&= \frac{0.5}{\lambda_B} + \left[(t)\left(-e^{0.5\lambda_B t}\right)\right]_0^\infty - \int_0^\infty e^{0.5\lambda_B t} dt \\
&= \frac{0.5}{\lambda_B} + \left[-te^{0.5\lambda_B t}\right]_0^\infty - \left[-\frac{1}{0.5\lambda_B} e^{0.5\lambda_B t}\right]_0^\infty \\
&= \frac{0.5}{\lambda_B} + \left[-te^{0.5\lambda_B t} + \frac{1}{0.5\lambda_B} e^{0.5\lambda_B t}\right]_0^\infty \\
&= \frac{0.5}{\lambda_B} + \lim_{x \to +\infty}\left[-te^{0.5\lambda_B t} + \frac{1}{0.5\lambda_B} e^{0.5\lambda_B t}\right] + \frac{1}{0.5\lambda_B} \\
&= \frac{0.5}{\lambda_B} + \frac{1}{0.5\lambda_B} = \frac{2.5}{\lambda_B}
\end{aligned} \tag{7}$$

Substituting the expression from Eq. 7 into Eq. 6 we obtain the expression in Eq. 8. The right-hand side of the inequality in Eq. 8 represents the smallest possible value for λ_B for which the waiting policy π_K^1 is a better option than the re-plan policy π_K^2. The inequality also takes into consideration the constraint that $\lambda_B > 0$ and negates the solution when $\lambda_B < 0$.

$$\begin{aligned} \frac{2.5}{\lambda_B} &\leq \frac{L_{B_1C_1} + L_{C_1C} - L_{B_1B} - L_{BC}}{L_W} \\ \lambda_B &\geq \frac{2.5L_W}{L_{B_1C_1} + L_{C_1C} - L_{B_1B} - L_{BC}} \end{aligned} \tag{8}$$

1.2 Policy Selection at Start

1.3 Considering the Probability of the Obstacle Being Present

1.4 Considering Multiple Obstacles

2 Implementation of the System in ROS

2.1 Planning path via an Aisle

In the given map of the environment (Fig. 1a), we can see that the map has 5 labelled aisles - A, B, C, D and E. For the given system, it might be desirable to be able to control the path planning such that the planned path goes down an aisle.
To implement this system, the function *planPathToGoalViaAisle()* in *comp0037_reactive_planner_controller/src/reactive_planner_controller.py* was completed. To ensure that the path planned goes down a particular aisle, a multiple destination path planning approach was adopted such that the first destination is the mid point[1] of the aisle the path has go down and the second and final destination is the goal. The function *planPathToGoalViaAisle()* first checks to see if the class variable *aisleToDriveDown* is None. If this is true, it assigns the value of the function parameter **aisle** to **aisleToDriveDown** and then the coordinates of the midpoint of the aisle is found. Once, the aisle cell coordinates are obtained path planning is done by calling **planner.search** twice. First the path is planned from the starting cell (*startCellCoords*) to the mid point of the aisle (*aisleCellCoords*). Then, a path is planned from the mid point of the aisle(*aisleCellCoords*) to the goal cell (*goalCellCoords*). Finally, the two individual paths are concatenated together and the concatenated path is plotted on search grid so that the resulting planned path can be observed.
The resulting search grid to demonstrate the performance of this function can be seen in Fig. 3 below.

[1]The walls dividing the aisles are disjoint in the middle this means that choosing the mid point would ensure this algorithm does not result in adding to a higher cost than is required to be able to plan a via an aisle. Hence, the mid point was chosen.

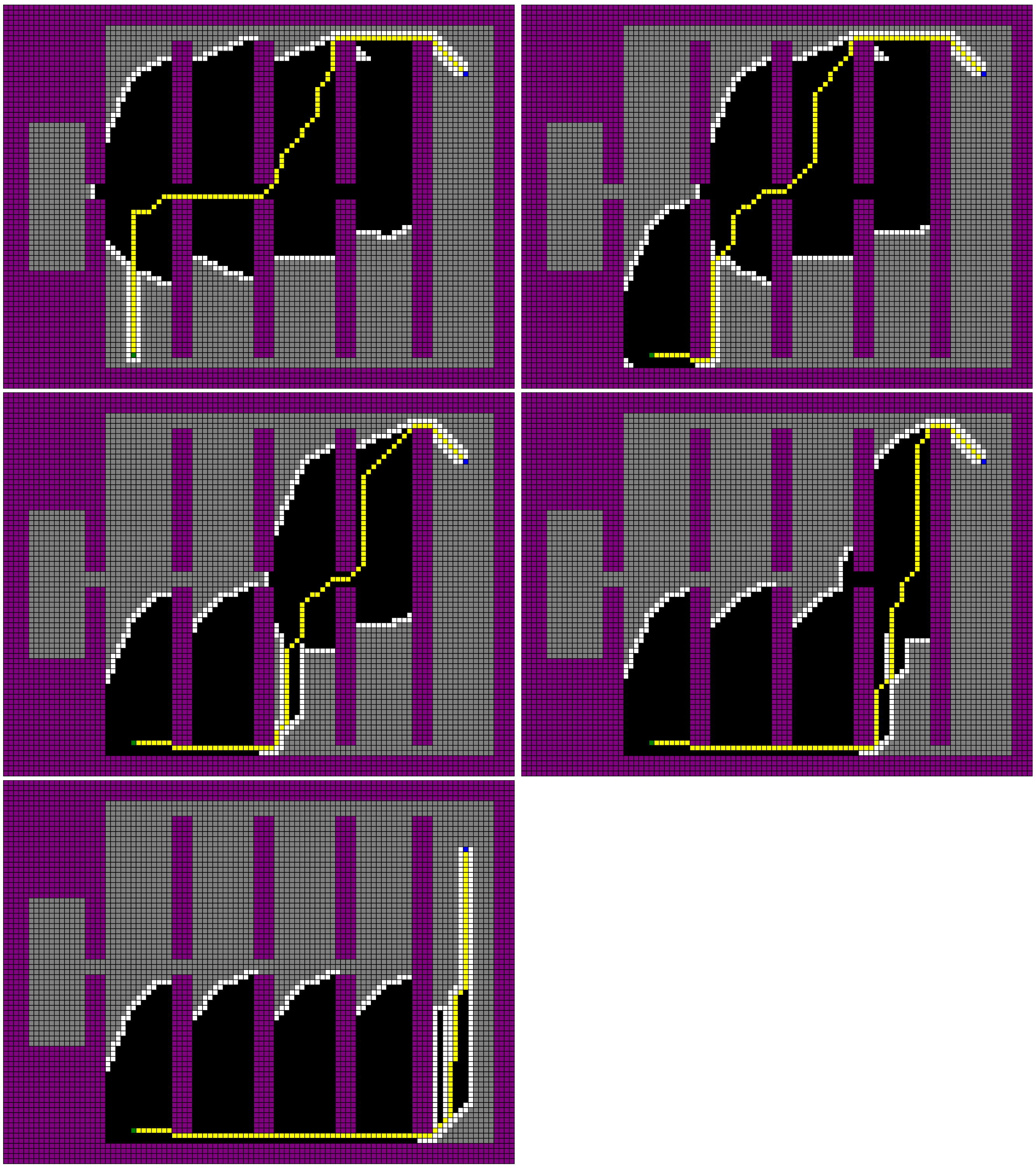


Figure 3: The demonstration of results of planning via each of the aisles.

References

Appendices