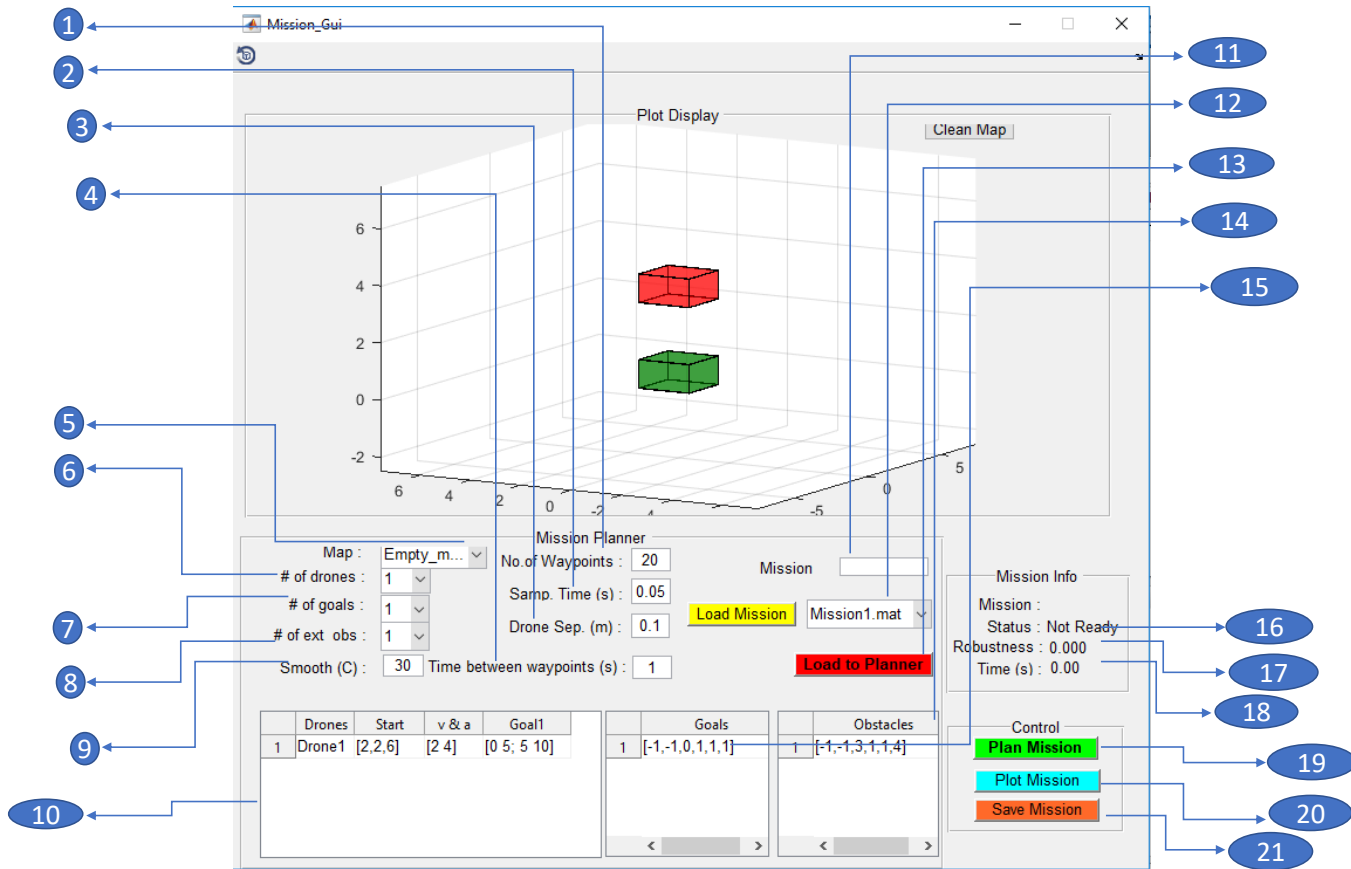


Fly By Logic - GUI User Manual

December 15, 2018

1 User Interface



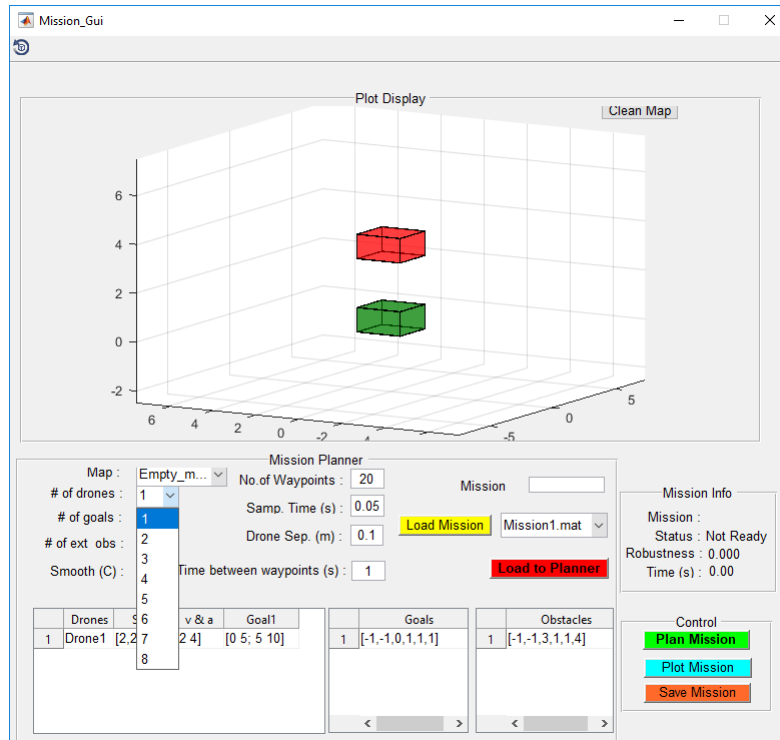
1.

2 GUI Nomenclature

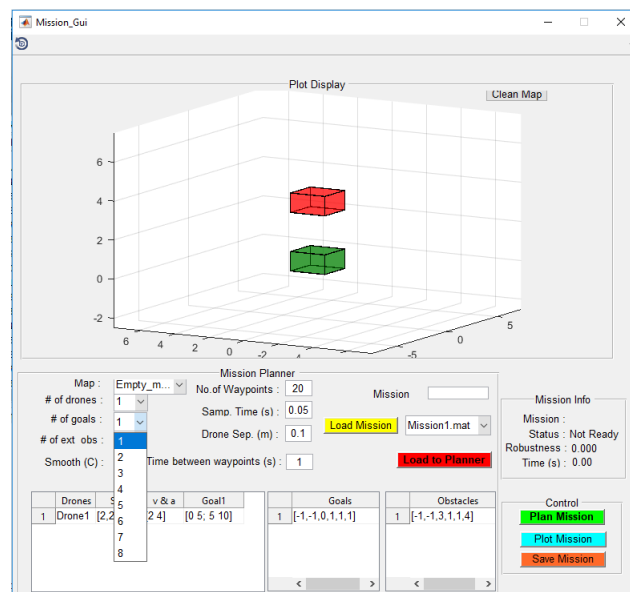
With reference to the indexing on the GUI Layout, each feature is explained below

1. **No. of Waypoints** - Given the start and goal positions, the planner generates waypoints between these two. And this parameter denotes the number of high samples waypoints to be generated between each two waypoints given by the planner.
2. **Samp. Time (s)** - Abbreviated to Sampling Time. This depicts the rate at which the high sampled points are generated.

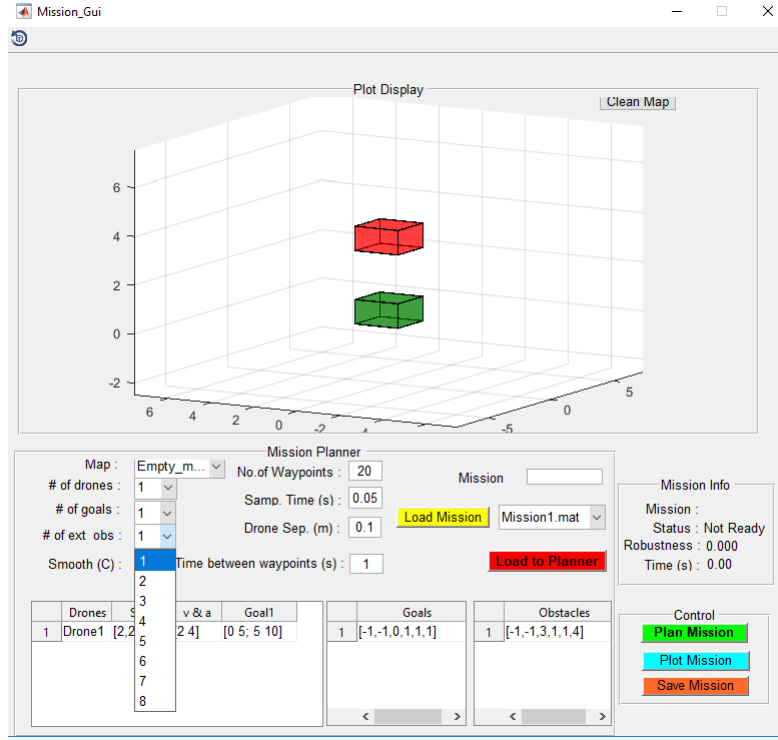
3. **Drone Sep. (m)** - The minimum distance that the drones to be maintained between each other.
4. **Time between waypoints (s)** - The time taken by the drone to travel between any two way points generated by the planner.
5. **Map** - Loads the Map. These maps can be designed and saved by the user to load to the planner.
6. **# of drones** - Number of drones required to perform a specific mission.



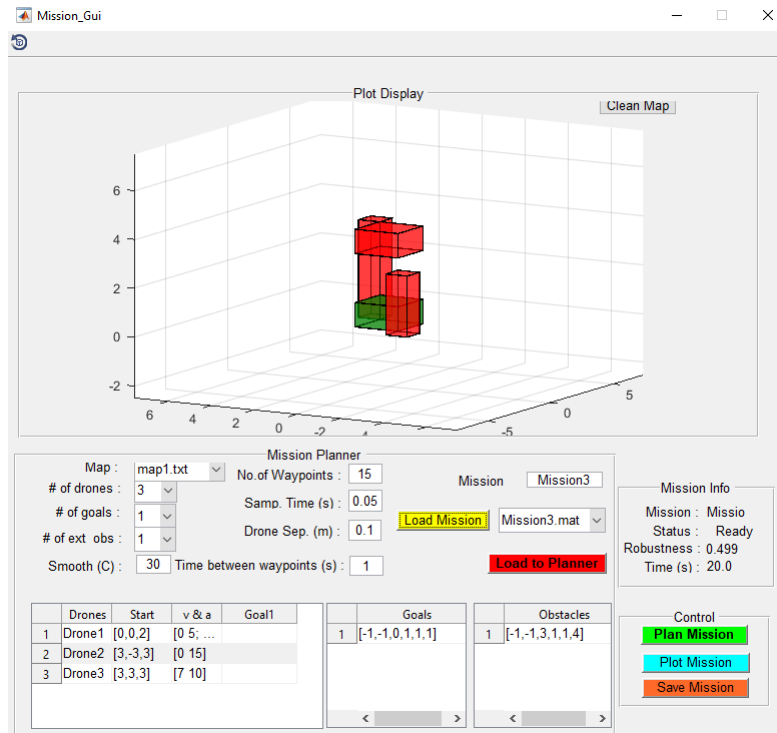
7. **# of goals** - Number of goal sets for the drones to oblige in the given time interval for a specific mission.



8. **# of ext obs** - Number of external obstacles that the drones need to avoid while executing the mission.



9. **Smooth (C)** - Smoothing parameter that is used to smooth the robustness function to solve the optimization problem.
10. **Mission Table** - This table gives the core Mission parameters. It has three + n columns where n is the number of Goal time intervals. The first three columns of the table are **Drones**, **Start**, **v & a**. The Drones column shows the Drone number. The Start column gives the 3D Cartesian of the start position of the respective drone. This should be in the format $[x,y,z]$. The v & a column gives the velocity and acceleration bounds on the respective drone. This should be in the format $[v,a]$. v is in $\frac{m}{s}$ and a is in $\frac{m}{s^2}$. The rows indicate these parameter with respect to **# of drones**. The rest of the n columns gives the Goal time intervals that the respective drone need to obey for a specific mission. n is equal to the **# of goals** parameter. The format for the goal time interval is an array $[t_1 t_2; t_3 t_4]$. Consider an example where the goal time interval is $[0 5; 10 15]$ and there is one goal. This says that, the respective drone should reach the goal set eventually in $[0 5]$ time interval and it should reach goal set again in $[10 15]$ time interval. All the numbers in the goal time interval are in seconds.
11. **Mission** - Name of the custom mission to be saved. Custom missions are the missions designed by the user to be used in future.
12. **Load Mission** - This enables user to load the previous missions that are saved by the user.



13. Load to Planner - TO BE WRITTEN

14. **Obstacles** - This table gives the Lower and Upper bounds of the x, y, z coordinates of the obstacles. The number of obstacles are decided by the **# of ext obs** parameter. Consider an example where the obstacle is denoted by $[-1, -1, 3, 1, 1, 4]$. This says that the Lower bound on x-coordinate is -1 and the Upper bound on x-coordinate is 1. Similarly, the Lower bound on y-coordinate is -1, the Upper bound on y-coordinate is 1 and Lower bound on z-coordinate is 3 and the Upper bound on z-coordinate is 4. This gives the shape of the obstacle to be a cuboid.
15. **Goals** - This table gives the Lower and Upper bounds of the x, y, z coordinates of the Goals. The number of Goals by the **# of goals** parameter. Consider an example where the obstacle is denoted by $[-1, -1, 0, 1, 1, 1]$. This says that the Lower bound on x-coordinate is -1 and the Upper bound on x-coordinate is 1. Similarly, the Lower bound on y-coordinate is -1, the Upper bound on y-coordinate is 1 and Lower bound on z-coordinate is 0 and the Upper bound on z-coordinate is 1. This gives the shape of the Goal to be a cuboid.
16. **Status** - This lets the user know if the mission is **Ready** to be planned. If the user loads one of the saved missions, this parameter lets the user know that the mission is **Ready** to be planned after the loading has completed. Else, this parameter shows **Not Ready**
17. **Robustness** - This gives the robustness value of the mission after solving the optimization problem which is obtained by Planning the Mission through **Plan Mission** button.
18. **Time** Time taken to plan the mission.
19. **Plan Mission** - This button lets the user decide to plan the mission once all the parameters to the mission are given to the GUI.
20. **Plot Mission** - This button lets the user to plot the mission once the mission has been planned.
21. **Save Mission** - This button lets the user save the mission which is custom designed by the user. The user can choose to save the mission on a file name which can be given as an input in **Mission** parameter of the GUI.

3 Example Mission with Parameters

