

Catching Fire – Exercise 1: Obstacle Avoidance

In Exercise 1, you learn how quadcopter determines its path around the obstacles in the scenario. The exercise consists of two steps: 1. Creating a scenario consisting of obstacles, and 2. Simulate a drone in the developed scenario in Step 1.

Step 0. Setup the environment

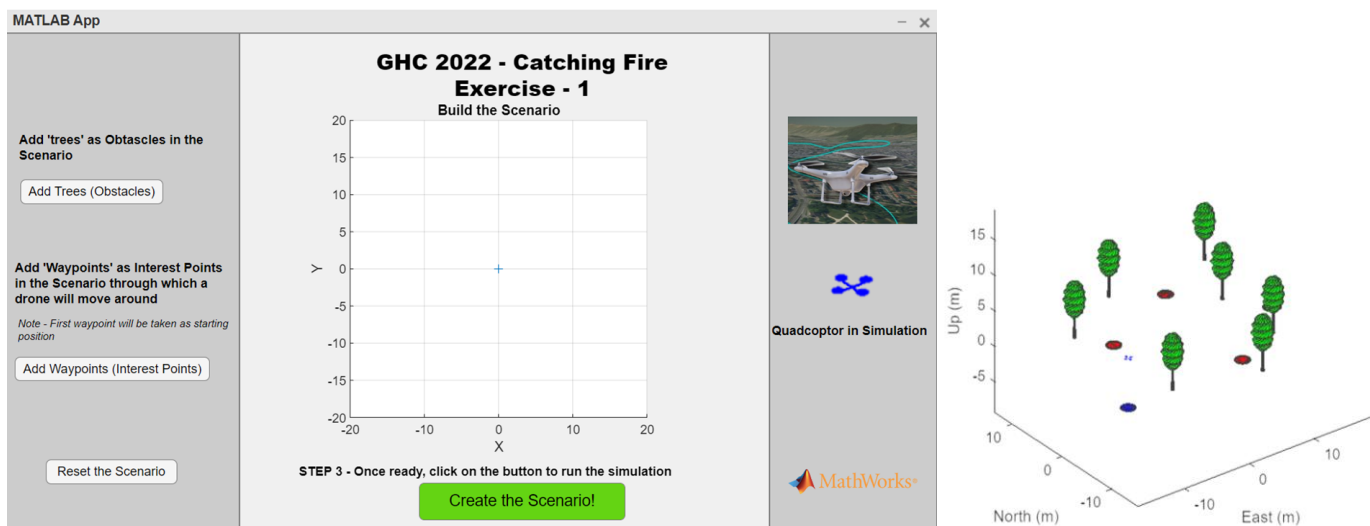
Before running the code for exercise 1, make sure to run the following command in the Command Window. This step adds the relevant folders to the path where the code for exercises is stored. This step also loads the model in the background for faster processing.

```
>> CatchingFire_Setup
```

Step 1. Generate the scenario

This step is an interactive step where you will be using a MATLAB App to add locations for obstacles and waypoints in the scenario to be used in the later step. Run the following command to open the app

```
>> Exercisel_Step1
```



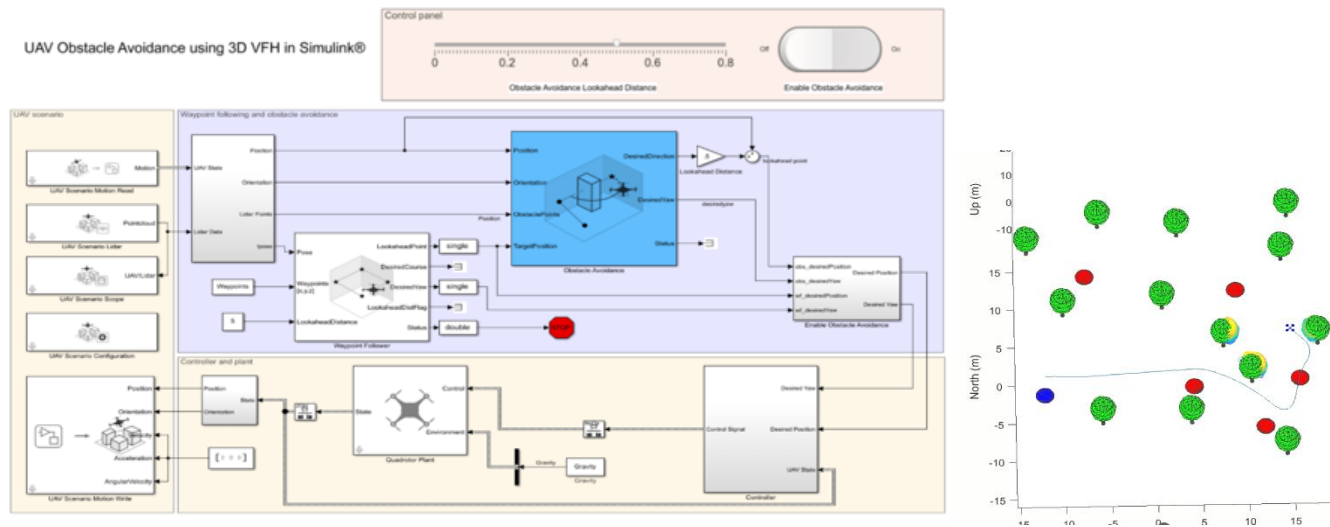
- **Add Trees** - Click on this button first to add obstacles in the plot provided on the right. Click in the plot region in multiple places. The round circles will appear wherever you click to show the position of the obstacles. Make sure to add obstacles with some space between them to allow drone to move freely.
- **Add Waypoints** - Click on this button next to add waypoints in the plot provided on the right. Click in the plot region in multiple places. The cross with the dashed line connecting to the waypoints will appear showing the path. The first waypoint will be treated as a initial position of the drone.
- **Reset the Scenario** - This is an optional button to be used if you would like to reset the plot region. This allows you to restart adding obstacles and waypoints.
- **Create the Scenario** - Once the locations of obstacles and waypoints are added to the plot region, the green button will be enabled. Click on this button to create a scenario, such as shown on the right above. This shows trees as the obstacle, red marks as the waypoint and blue as the initial position of the drone. The created scenario will be used for Step 2.

You can close the app now by clicking on the 'X' on the top right corner. Observe the location of the developed scenario opened as a separate window in the MATLAB Online.

Step 2. Simulate the drone path by following obstacle avoidance

This step uses Simulink model to simulate the drone (quadcopter in this case) path by utilizing an obstacle avoidance algorithm (3D Vector Field Histogram). Type the following command to open the model to run this step

>> Exercisel_Step2



The simulation (as shown in the right picture) will automatically pop-up and start once the Simulink model gets opened. The model consists of four main subsystems:

- **Scenario Building** - The scenario block configures the scenario and visualizes the obstacles, trajectory, and the lidar point cloud data.
- **Control Panel** - The switch enables or disables the updates to the lookahead point from the Obstacle Avoidance block. The slider updates the lookahead distance used to compute the lookahead point.
- **Waypoint Following and Obstacle Avoidance** - The Waypoint following, and obstacle avoidance subsystem finds the obstacle-free desired position and the desired yaw according to the current UAV state and point cloud data.
- **Controller and plant** - The Controller and plant subsystem generates the control commands and updates the UAV state based on the lookahead point.

Step 3. Monitor and observe the results

Monitor the path of the drone through the obstacles and waypoints. The parameters can be updated:

- The top right corner of the Simulink model provides a control panel with a button to on and off the use of obstacle avoidance. You can use this to see how the path of the drone changes in the simulation in real-time. Are you able to see the updated path?
- You can restart the exercise from step 1 and add more obstacles and change the waypoints. Follow the steps to see how the drone's path changes in the updated scenario.
- Once more time permits, you can review the code to update the parameters associated with drone and the sensor mounted on the drone to analyze the change in behavior.

Learn More

- UAV Toolbox: [mathworks/uav](https://mathworks.com/products/uav/)
- UAV Obstacle Avoidance Documentation Example: [mathworks/uav-obstacle-avoidance-in-simulink](https://mathworks.com/help/uav/ug/uav-obstacle-avoidance-in-simulink.html)
- Vector Field Histogram Algorithm: [mathworks/controllervfh-system-object](https://mathworks.com/help/controllervfh/ug/vector-field-histogram-algorithm.html)
- Modeling and simulation: mathworks.com/discovery/modeling-and-simulation
- Simulink and Simulink Online: mathworks.com/learn/tutorials/simulink-onramp
- MATLAB App Designer: [mathworks/app-designer](https://mathworks.com/products/app-designer/)