

sim_rrt

October 17, 2021

1 RRT Sampling-Based Motion Planning

```
[1]: # The autoreload extension will automatically load in new code as you edit ↵
      ↪files,
      # so you don't need to restart the kernel every time
      %load_ext autoreload
      %autoreload 2

      import numpy as np
      import matplotlib.pyplot as plt
      from P2_rrt import *

      plt.rcParams['figure.figsize'] = [8, 8] # Change default figure size
```

1.0.1 Set up workspace

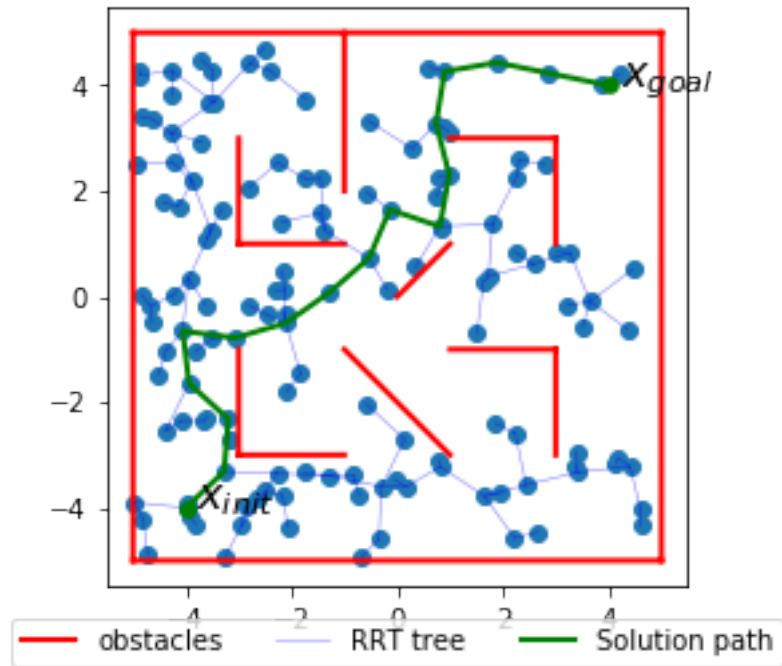
```
[2]: MAZE = np.array([
      (( 5, 5), (-5, 5)),
      ((-5, 5), (-5,-5)),
      ((-5,-5), ( 5,-5)),
      (( 5,-5), ( 5, 5)),
      ((-3,-3), (-3,-1)),
      ((-3,-3), (-1,-3)),
      (( 3, 3), ( 3, 1)),
      (( 3, 3), ( 1, 3)),
      (( 1,-1), ( 3,-1)),
      (( 3,-1), ( 3,-3)),
      ((-1, 1), (-3, 1)),
      ((-3, 1), (-3, 3)),
      ((-1,-1), ( 1,-3)),
      ((-1, 5), (-1, 2)),
      (( 0, 0), ( 1, 1))
])

      # try changing these!
      x_init = [-4,-4] # reset to [-4,-4] when saving results for submission
      x_goal = [4,4] # reset to [4,4] when saving results for submission
```

1.1 Geometric Planning

```
[3]: grrt = GeometricRRT([-5,-5], [5,5], x_init, x_goal, MAZE)
grrt.solve(1.0, 2000)
```

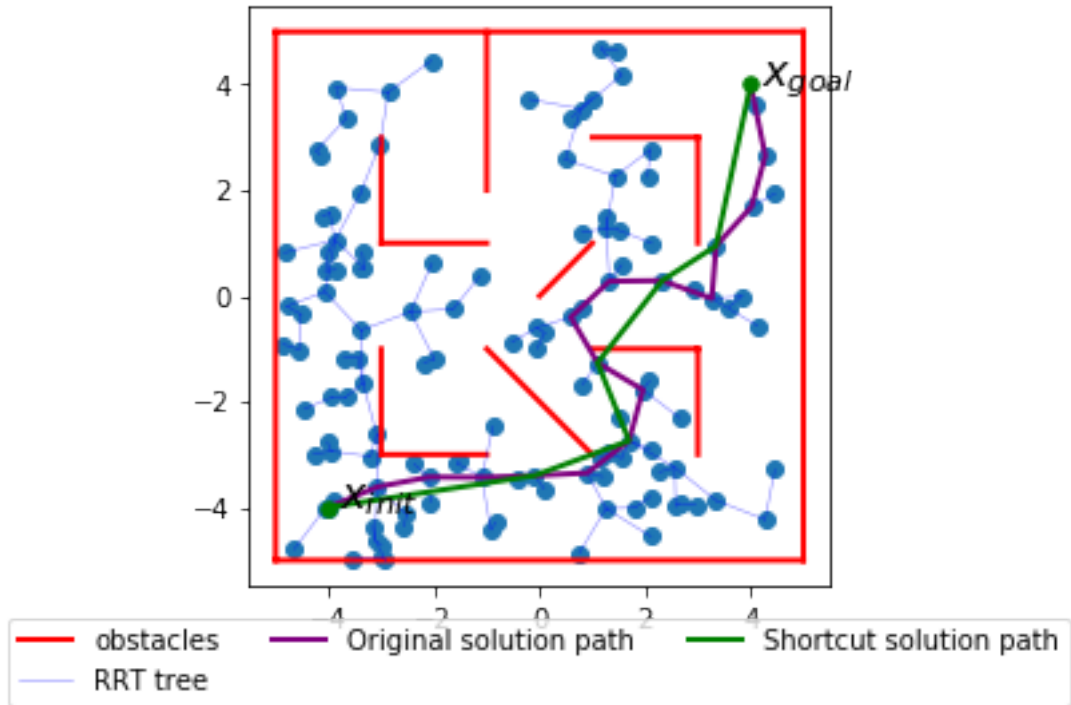
[3]: True



1.1.1 Adding shortcutting

```
[4]: grrt.solve(1.0, 2000, shortcut=True)
```

[4]: True

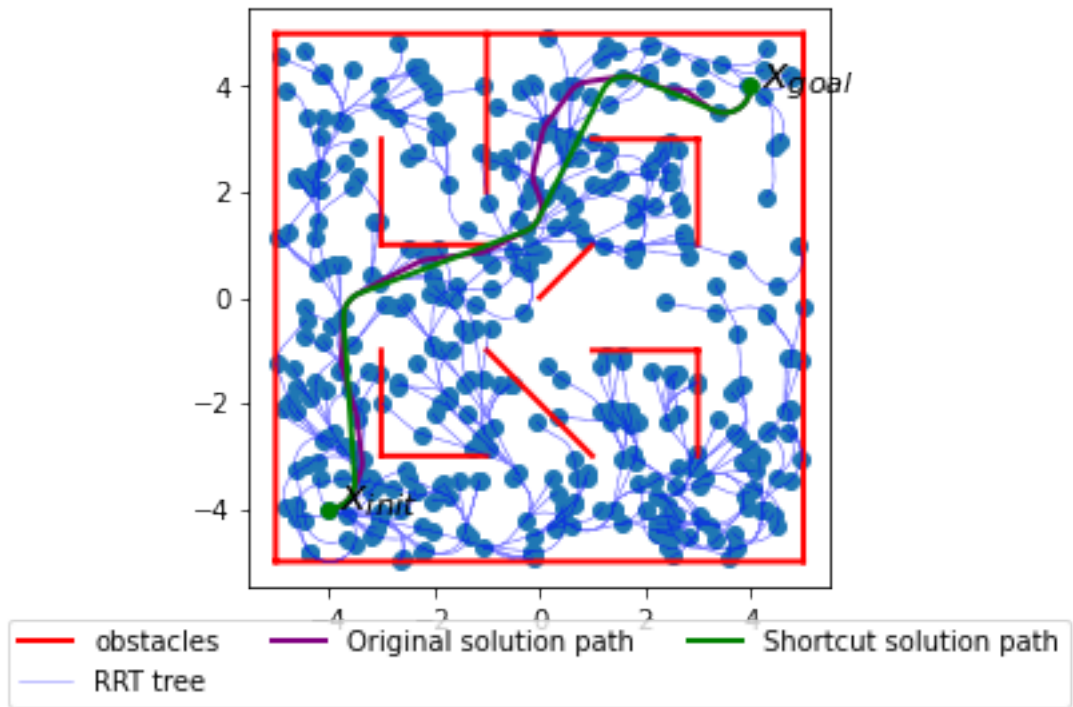


1.2 Dubins Car Planning

```
[5]: x_init = [-4,-4,0]
     x_goal = [4,4,np.pi/2]

     drrt = DubinsRRT([-5,-5,0], [5,5,2*np.pi], x_init, x_goal, MAZE, .5)
     drrt.solve(1.0, 1000, shortcut=True)
```

```
[5]: True
```



[]: