Programming Swarm Robotics

Occupancy Grid Maps:

* Ways to represent what’s occupied and what is not – obstacles occupy space and free spaces don’t occupy any spaces
* You split the map into individual cells, and light up the discrete cells if they are occupied. Even if a cell (pixel?) is partially occupied, it is represented by lit pixel (pixel =1), and the rest are pixel =0.
* These are also called binary maps – occupied/not occupied by something
  + Probabilistic maps – different probability that a given cell is occupied or not (0.5 = 50% probability that a given cell (pixel) is occupied or not.

Plant Model for the vehicle:

* A single-track car that runs by providing different wheel speeds to the two wheels. As the speeds on either side of the robot change, the vehicle turns in a certain direction.
* The vehicle inputs given are typical of a vehicle

Controller:

* The controller used is a pure pursuit controller, wherein the robot continuously pursues a point in space. The logic behind this algorithm is similar to the way humans drive – they figure out a point where they would like to steer/accelerate a car to and pursue that point until the vehicle reaches that point. The pursuing point keeps changing based on the look-ahead distance available in front of us.