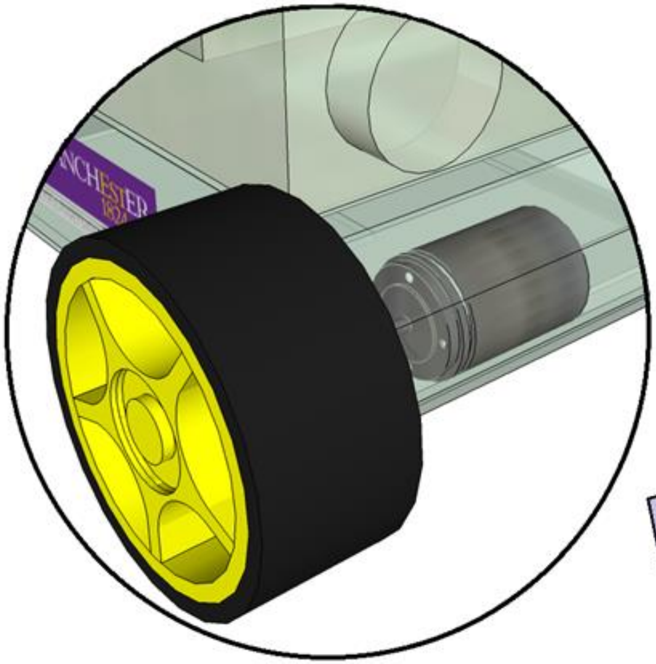
Autonomous Systems
- The 2nd level of autonomy -

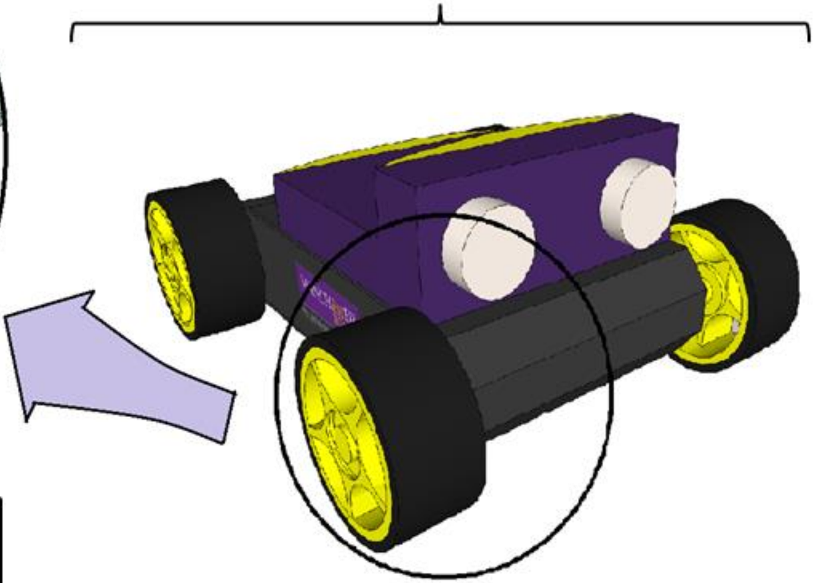
Dr Alexandru Stancu

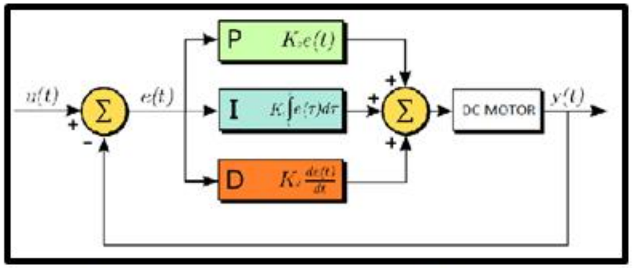
Dr Mario Martinez

Low Level Control

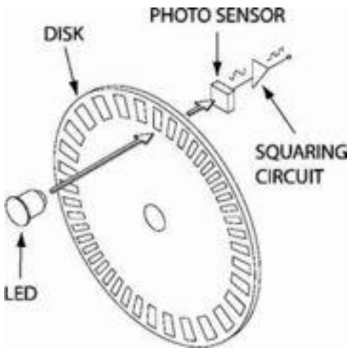


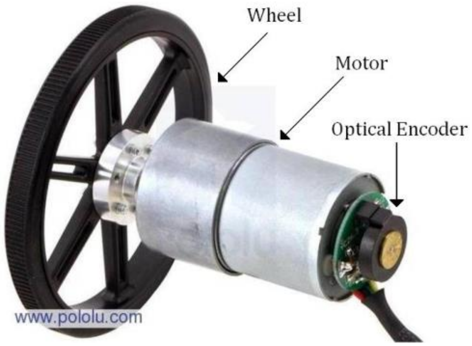
Dynamic Nonlinear Control





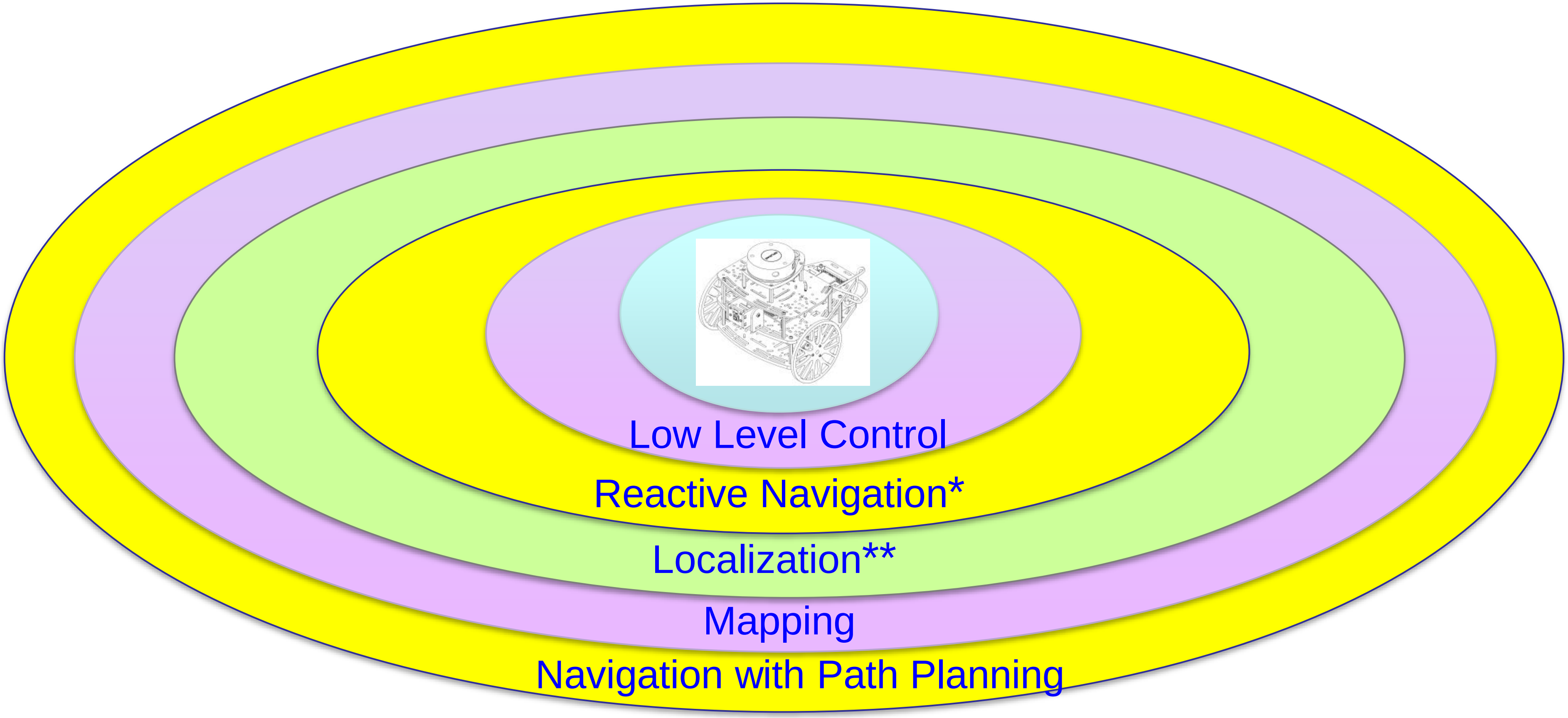
Wheel velocity control using PID





The hierarchy of the autonomy

How much information and support must be provided by human to ensure that the robot is able to achieve its goals.



2nd level of autonomy

The 2nd level of autonomy can be referred as:

- Reactive navigation, or
- Navigation with obstacle avoidance, or
- Short term navigation, etc.

Local obstacle avoidance is the process of modifying the robot trajectory by using the information provided by the sensors (exteroceptive). The resultant motion depends on different factors, such as **instantaneous sensor readings**, **goal position** and **relative location of the robot** in the environment. In this section, different algorithms that solve the obstacle avoidance problem will be presented.

Reactive Navigation (Obstacle Avoidance)

- Robot needs to navigate through the environment without running into obstacles.
- Robot needs to utilize exteroceptive sensors to identify obstacles.
- Example of exteroceptive sensors are: camera, LiDAR, sonar, etc...

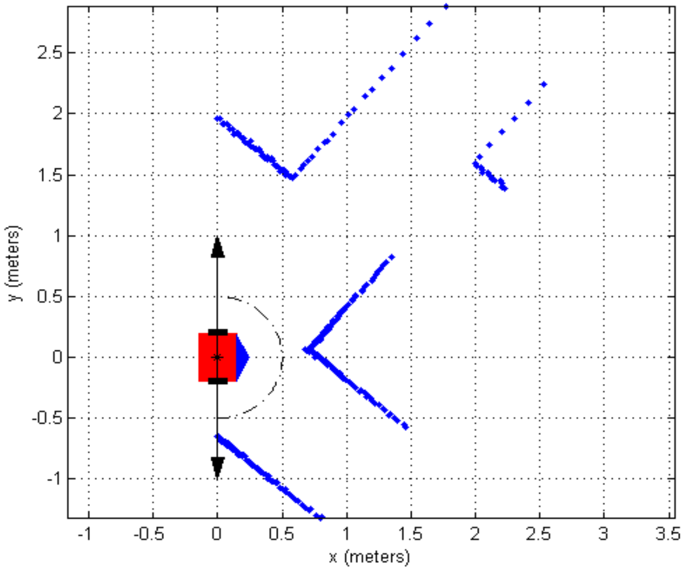
Kinect Sensor © Microsoft.

LiDAR © Hokuyo Automatic Inc.

Sonar © SparkFun Electronics.

Exteroceptive sensors
- Laser Rangefinders -

The laser rangefinder uses the principle of time-of-flight of electromagnetic wave to measure the distance between the sensor and obstacles in the environment. By rotating the sensor horizontally, it is possible to detect all obstacles around the robot within the range of the sensor. These measurements can also be used to build maps of the environment, which are essential for localisation and navigation.



Obstacle Avoidance algorithms

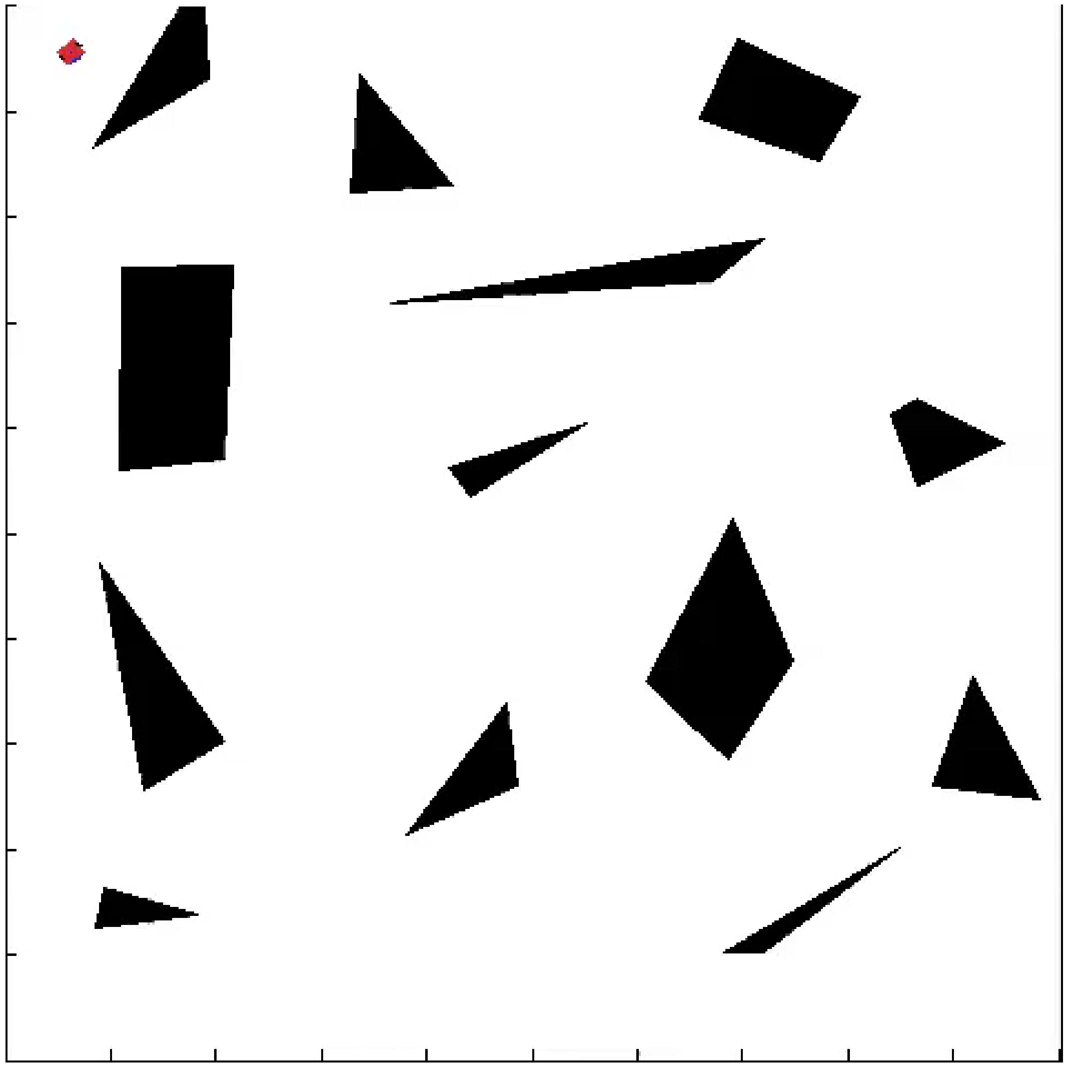
Bug algorithms assume only local knowledge of the environment and a global goal.

Bug behaviours are simple:
1) Move in a straight line toward goal

2) Follow a wall (right or left)

In other words, one can characterize the robot’s motion in terms of two states, one that involves **moving toward the goal** and a second that involves **moving around the contour of an obstacle**.

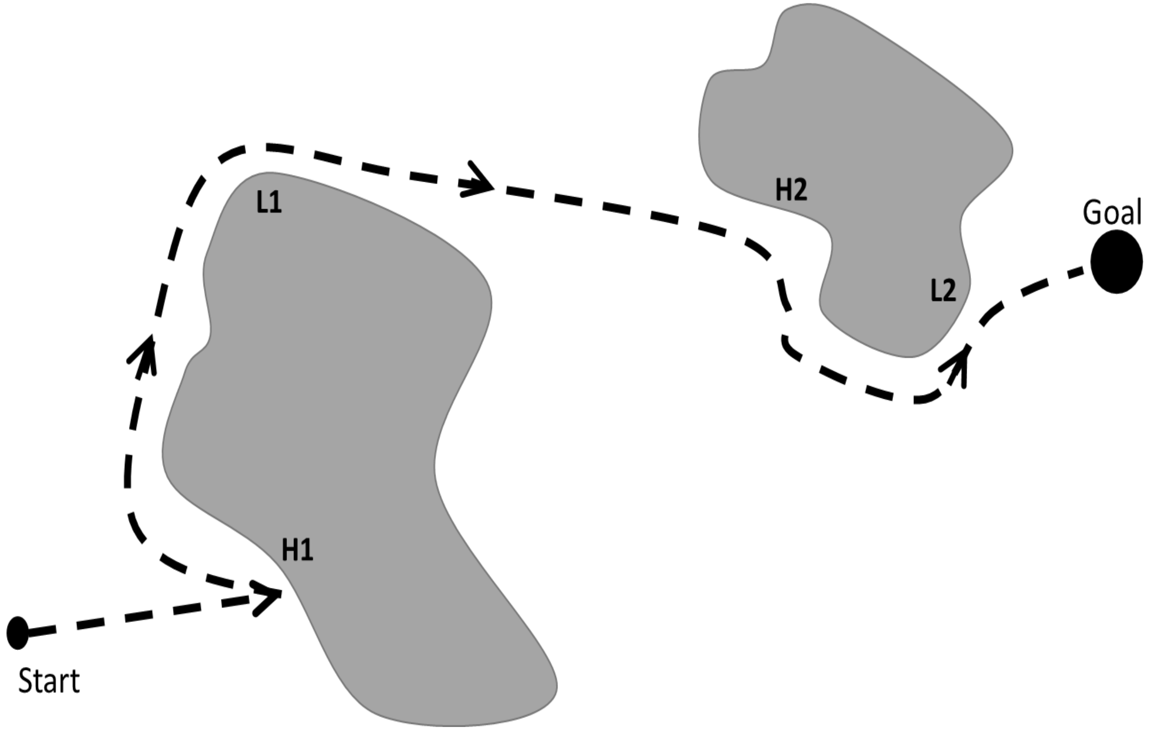
Reactive Navigation (Obstacle Avoidance)



Robot needs to know when it reaches the goal.

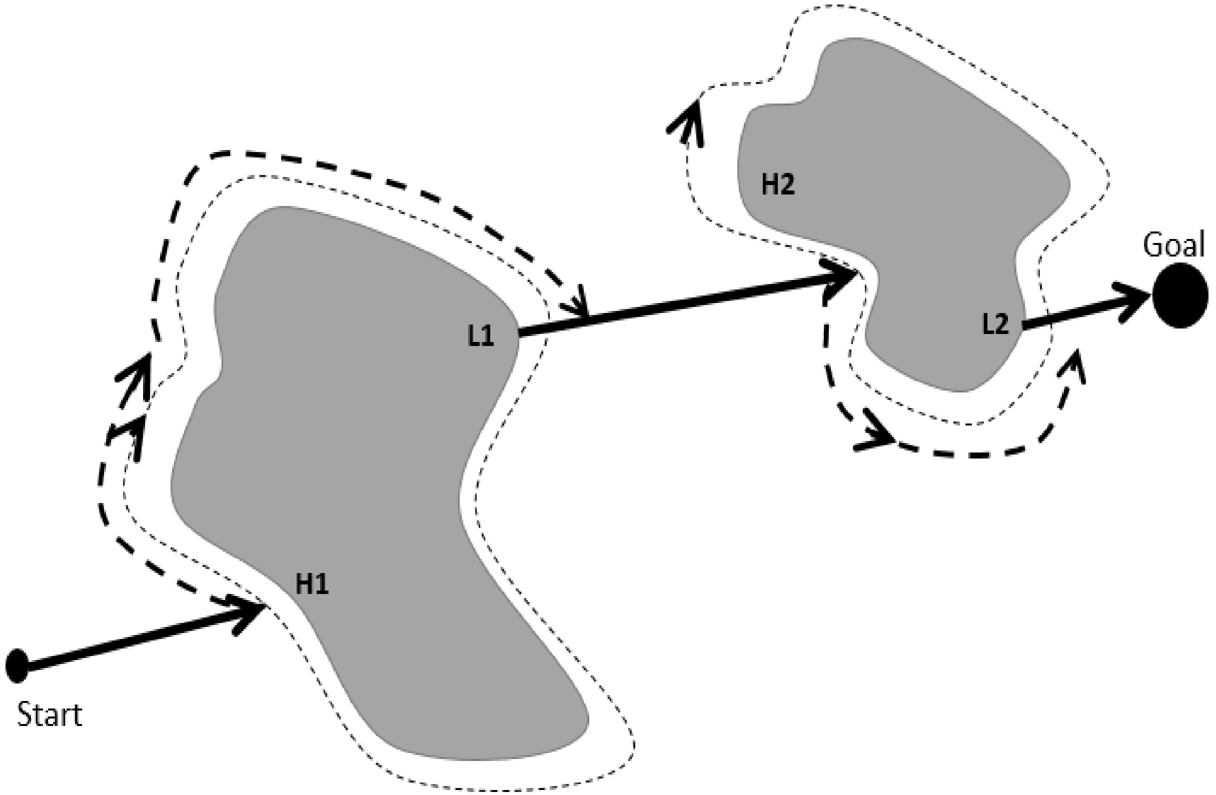
Obstacle Avoidance algorithms

Bug 0 This algorithm consists on the robot circumnavigating the obstacle and departing as soon as the robot can head towards the goal again.



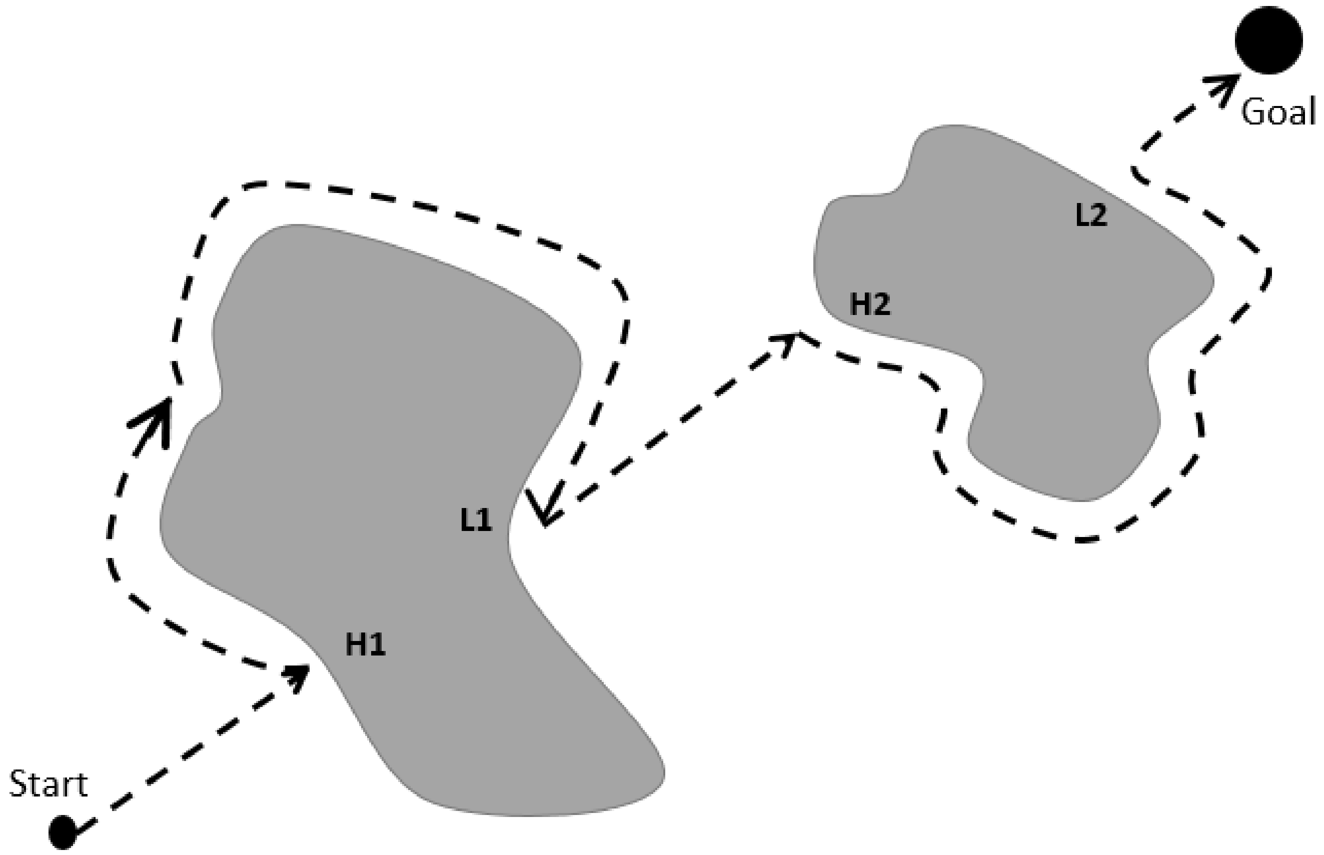
Obstacle Avoidance algorithms

Bug 1 This algorithm consists on the robot to fully circumnavigate the complete obstacle one time, then choosing and departing from the point with the shortest distance towards the goal.



Obstacle Avoidance algorithms

Bug 2 It consists on circumnavigating the obstacle and departing immediately after finding a point that allows it to move towards the goal.



Obstacle Avoidance algorithms

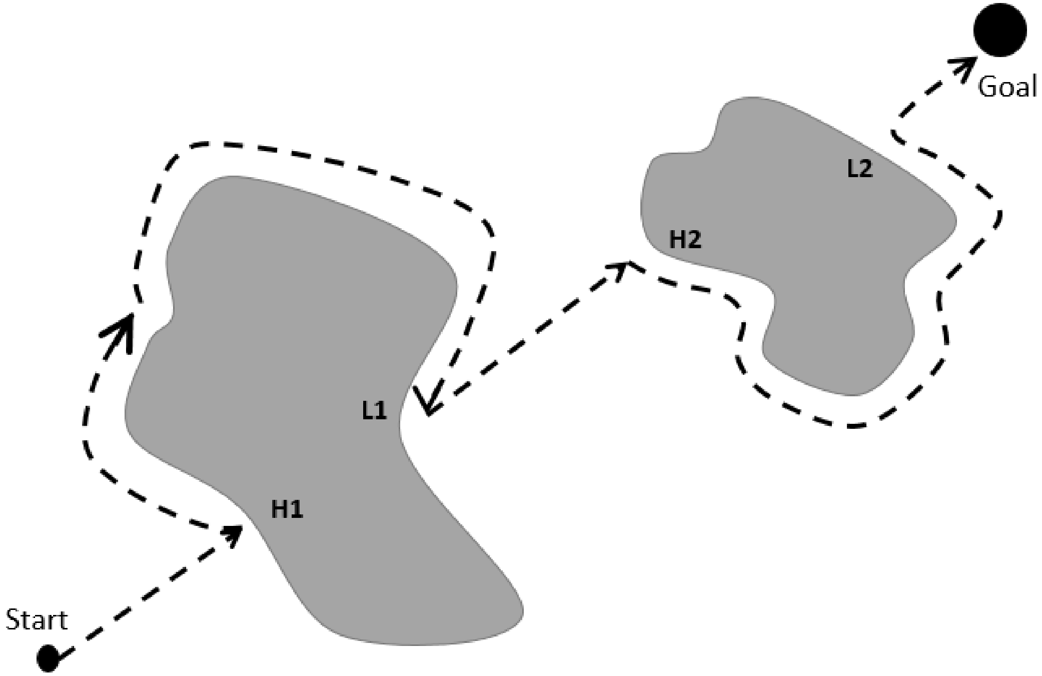
On a high level, the Bug2 algorithm has two main modes:

- **Go to Goal Mode: Move from the current location towards the goal (x,y) coordinate.**

- **Wall Following Mode: Move along a wall.**

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:



Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

1. Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

1. Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.
2. While Not at the Goal

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

1. Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.
2. While Not at the Goal

Move towards the goal along the start-goal line.

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

```
1.    Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.
2.    While Not at the Goal
Move towards the goal along the start-goal line.
      If a wall is encountered:
            Remember the location where the wall was first encountered. This is the "hit point."
```

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

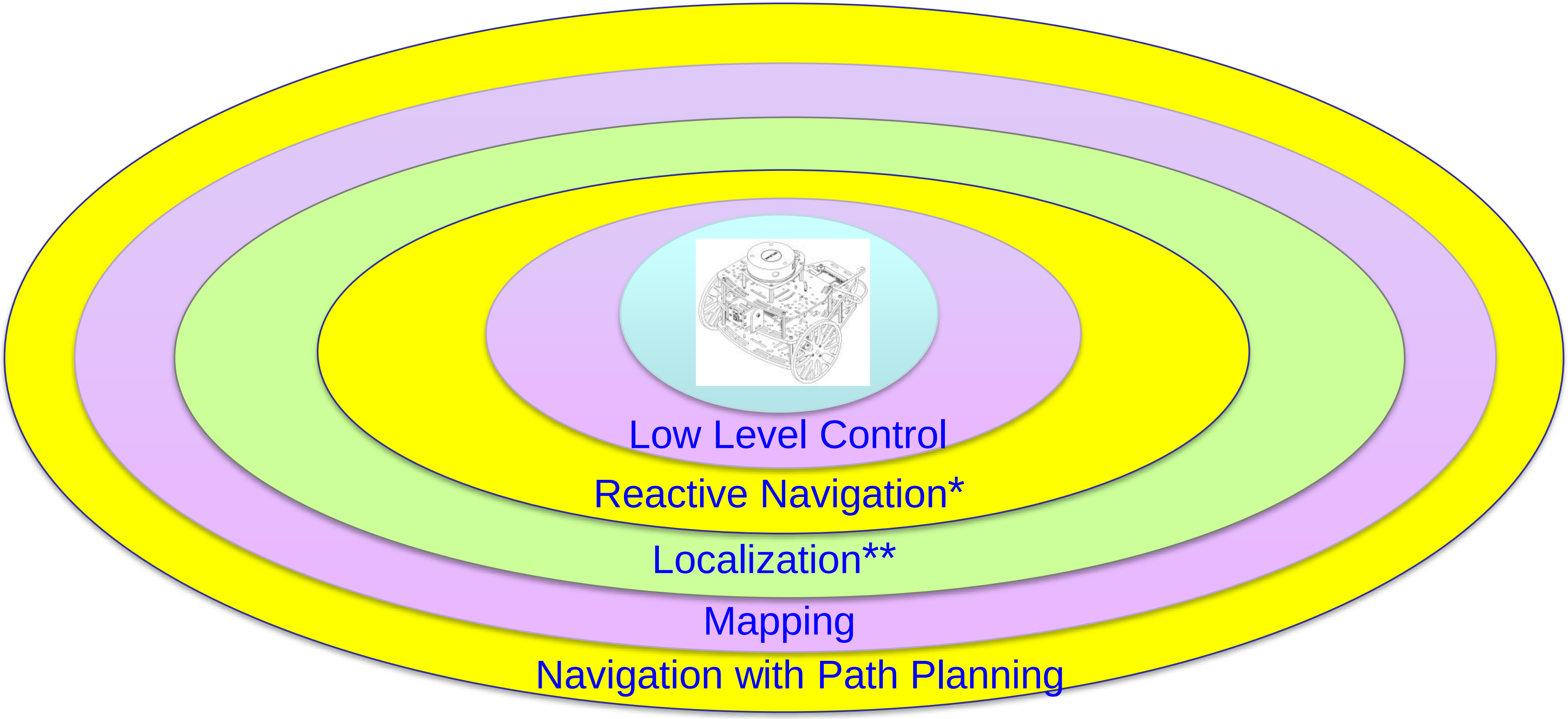
```
1.    Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.
2.    While Not at the Goal
Move towards the goal along the start-goal line.
    If a wall is encountered:
        Remember the location where the wall was first encountered. This is the “hit point.”
        Follow the wall until you encounter the start-goal line. This point is known as the “leave point.”
```

Obstacle Avoidance algorithms

The pseudocode for Bug2 algorithm:

```
1.   Calculate a start-goal line. The start-goal line is an imaginary line that connects the starting position to the goal position.
2.   While Not at the Goal
Move towards the goal along the start-goal line.
     If a wall is encountered:
          Remember the location where the wall was first encountered. This is the “hit point.”
          Follow the wall until you encounter the start-goal line. This point is known as the “leave point.”
               If the leave point is closer to the goal than the hit point, leave the wall, and move towards the goal again.
               Otherwise, continue following the wall.
```

The hierarchy of the autonomy



Low Level Control

Reactive Navigation*

Localization**

Mapping

Navigation with Path Planning

Next theoretical lecture
- The 3rd level of autonomy -

- Localisation using exteroceptive sensors