Navigation and Motion Planning

Where am I going? How do I get there?

Introduction

Let's take a look at the cognitive level of the robot. Cognition is a term that refers to a system's deliberate decision-making and execution in order to attain its highest-order goals.

The special feature of cognition directly linked to robust mobility in the case of a mobile robot is navigation competency. Navigation refers to a robot's ability to act based on its knowledge and sensor values in order to reach its goal positions as efficiently and reliably as possible given partial knowledge about its environment and a goal position or series of positions. This presentation focuses on the navigation and planning challenges.

Two key additional competencies

Path planning
It entails determining a route that, when followed, will lead the robot to the desired location.
Path planning is a strategic problem-solving skill since the robot must decide what to do in the long run to attain its objectives.

Obstacle avoidance
It entails modifying the robot's trajectory in order to avoid collisions. Obstacle avoidance has been proven using a wide range of ways.

Path Planning

How do we go about planning a path?

Path planning for manipulator robots and, indeed, even for most mobile robots, is formally done in a representation called configuration space.

The most common approach is to assume for path planning purposes that the robot is in fact holonomic, simplifying the process tremendously. Furthermore, mobile roboticists will often plan under the further assumption that the robot is simply a point. Thus we can further reduce the configuration space for mobile robot path planning to a two dimensional representation with just x and y axes.

Because we have reduced the robot to a point, we must inflate each obstacle by the size of the robot’s radius to compensate

Path Planning Strategies

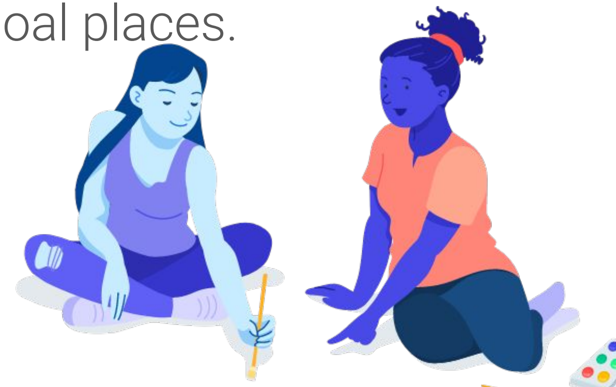
1 Road map

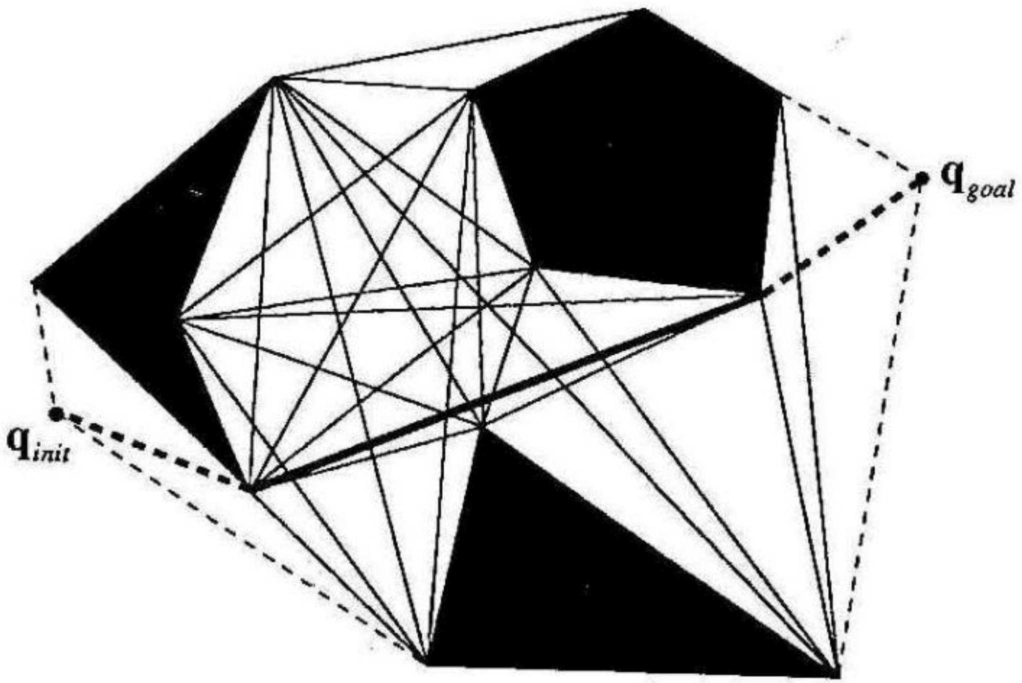
2 Cell decomposition:

3 Potential field

Road-Map Path Planning

The connectivity of the robot's free space is captured in a network of one-dimensional curves or lines called road-maps. Once a road-map has been created, it is utilised to plan robot mobility as a network of road (path) segments. As a result, path planning is reduced to linking the robot's beginning and goal positions to the road network, then searching for a series of roads that connect the robot's initial and goal places.





Visibility graph (adopted from [5] p. 13):
The nodes of the graph are the initial and goal points and the vertices of the configuration space obstacles (polygons). All nodes which are visible from each other are connected by straight line segments, defining the road map. This means there are also edges alonhg each polygon's sides.

Visibility graph is a type of road map

Cell Decomposition Path Planning

The goal of cell decomposition is to distinguish between free geometric spaces, or cells, and those that are occupied by objects. The positioning of cell boundaries is an important aspect of cell breakdown methods.

Exact cell decomposition is a method in which the borders are placed as a function of the environment's structure, resulting in lossless decomposition.

The system is called ***approximate cell decomposition*** if the decomposition produces an approximation of the actual map.

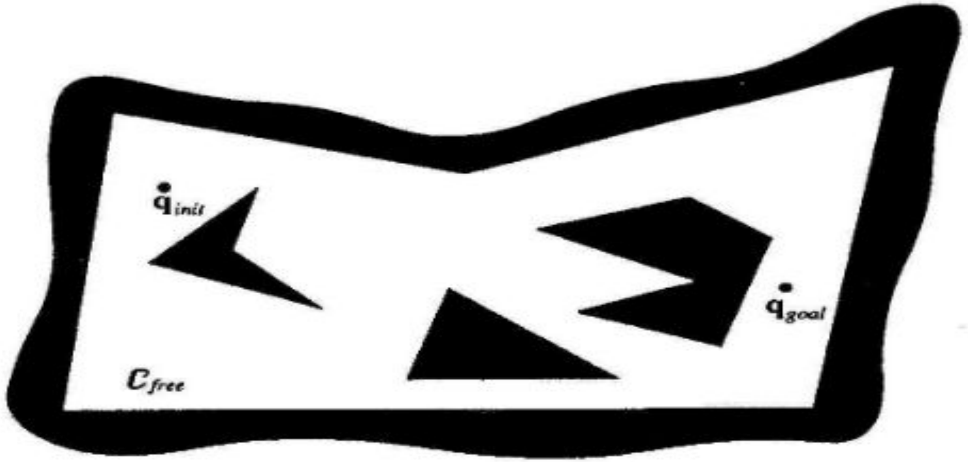
Steps involved:

1. Divide F into simple, connected regions called “cells”

2. Determine which opens cells are adjacent and construct a “connectivity graph”

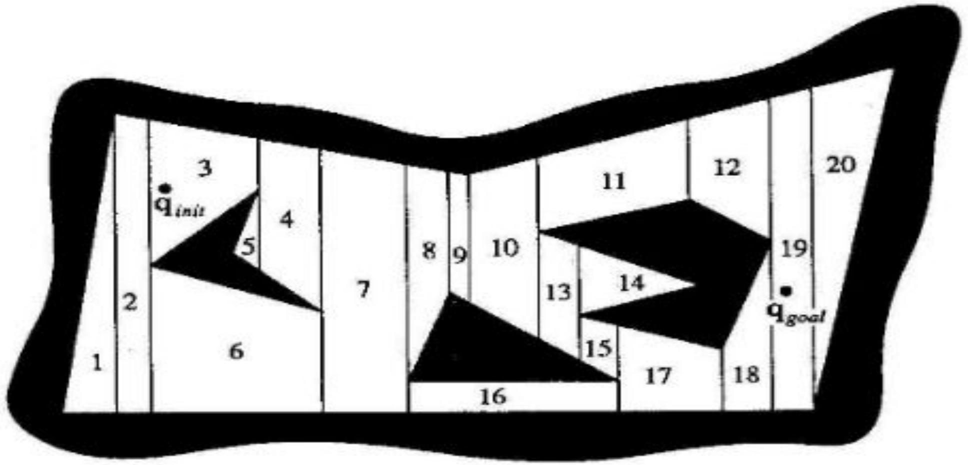
3. Find the cells in which the initial and goal configurations lie and search for a path in the connectivity graph to join the initial and goal cell.

4. Compute a path within each cell using the sequence of cells found with an appropriate searching technique, such as going via the midpoints of the cell boundaries or a series of wall following motions and movements along straight lines.



q_{init}
q_{goal}
$\mathcal{C}_{free}$

(a)



1
2
3
q_{init}
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
q_{goal}
20

(b)

Potential Field Path Planning

Potential field path planning creates a field, or gradient, across the robot's map that directs the robot to the goal position from multiple prior positions.

This approach was originally invented for robot manipulator path planning and is used often and under many variants in the mobile robotics community.

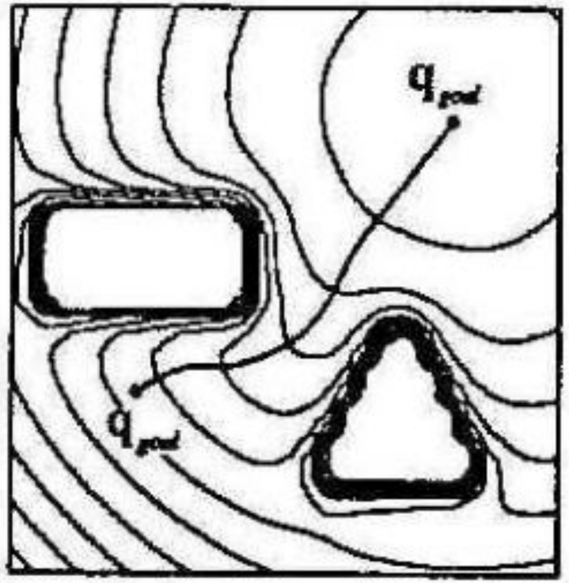
How does it work?

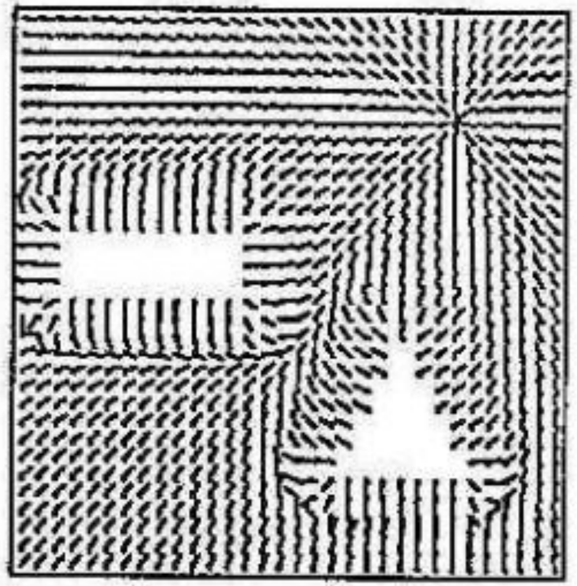
The robot is treated as a point under the influence of an artificial potential field in the potential field approach. Like a ball rolling downhill, the robot moves by following the field.

The goal (the lowest point in this region) attracts the robot, while the barriers act as peaks, repelling it. All forces are superimposed on the robot, which is believed to be a point in the configuration space in most circumstances.

An artificial potential field like this smoothly directs the robot toward the target while avoiding known impediments.

)


q_{goal}
q_{goal}



Obstacle Avoidance

What is it?

During robot motion, local obstacle avoidance focuses on changing the robot's trajectory as informed by its sensors. The robot's current or recent sensor readings, as well as its goal position and relative location to the goal position, determine the robot's motion.

The obstacle avoidance algorithms that we will see rely on the existence of a global map and the robot's accurate awareness of its location relative to the map to varied degrees.

Bug Algorithm

The Bug algorithm is one of the most basic obstacle avoidance algorithms available.

The primary idea is to circum-navigate each obstacle in the robot's path by following the contour of the obstacle.

With Bug1, the robot first completes a full circle around the object before departing from the place with the shortest distance to the target. Of course, this method is inefficient, but it ensures that the robot will attain every goal that may be reached.

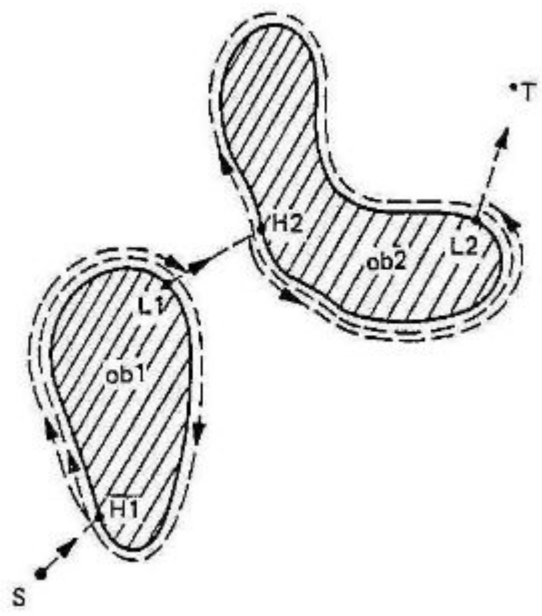
Bug-2 Algorithm

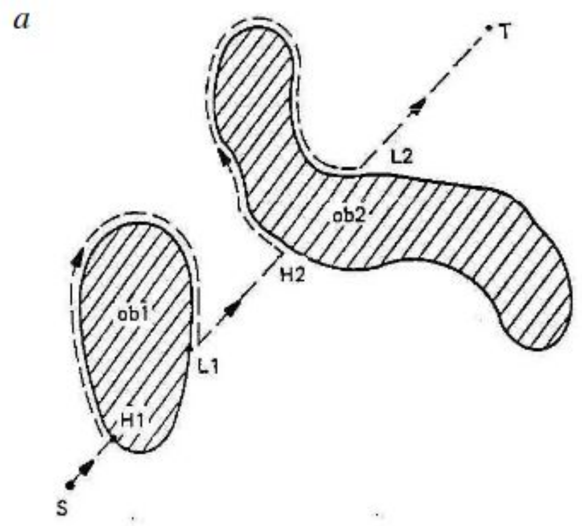
With Bug2 the robot begins to follow the object’s contour, but departs immediately when it

is able to move directly toward the goal. In general this improved Bug algorithm will have significantly shorter total robot travel as shown in path.

However, one can still construct situations in which Bug2 is arbitrarily inefficient

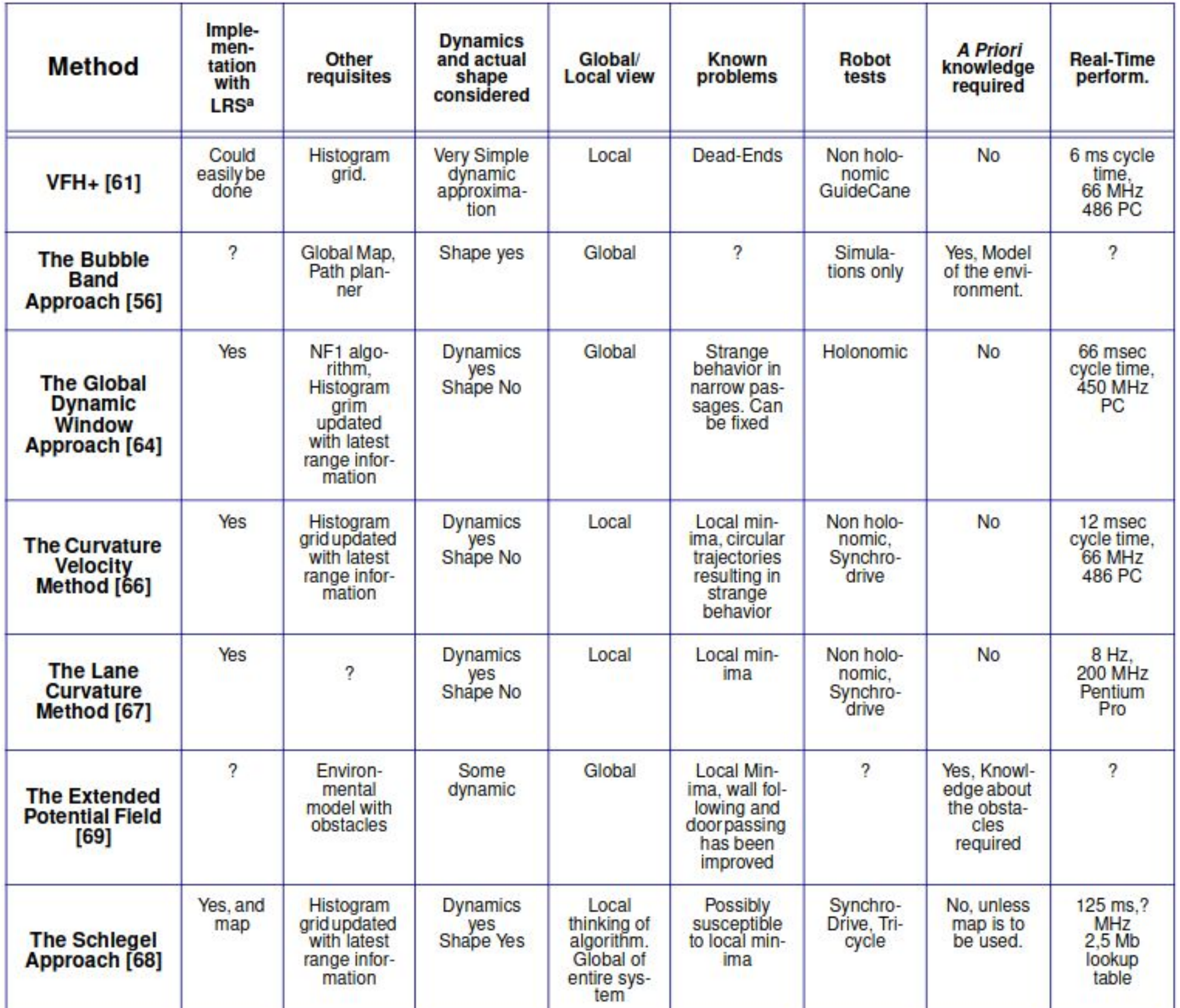
Bug-1 vs Bug-2 Algorithm





Some other well known algos

1. Vector Field Histogram
2. The Bubble Band Technique
3. Curvature Velocity Techniques
4. Dynamic Window Approaches
5. The Schlegel Approach to Obstacle Avoidance

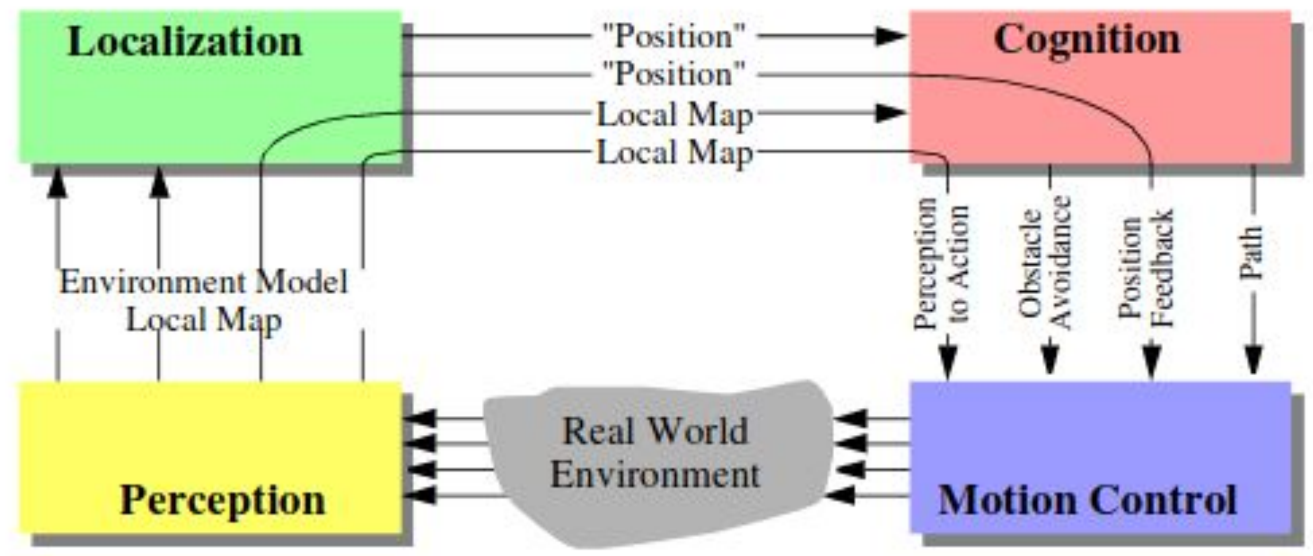


Method	Imple-men-tation with LRS[a]	Other requisites	Dynamics and actual shape considered	Global/ Local view	Known problems	Robot tests	*A Priori* knowledge required	Real-Time perform.
VFH+ [61]	Could easily be done	Histogram grid.	Very Simple dynamic approxima-tion	Local	Dead-Ends	Non holo-nomic GuideCane	No	6 ms cycle time, 66 MHz 486 PC
The Bubble Band Approach [56]	?	Global Map, Path plan-ner	Shape yes	Global	?	Simula-tions only	Yes, Model of the envi-ronment.	?
The Global Dynamic Window Approach [64]	Yes	NF1 algo-rithm, Histogram grim updated with latest range infor-mation	Dynamics yes Shape No	Global	Strange behavior in narrow pas-sages. Can be fixed	Holonomic	No	66 msec cycle time, 450 MHz PC
The Curvature Velocity Method [66]	Yes	Histogram grid updated with latest range infor-mation	Dynamics yes Shape No	Local	Local min-ima, circular trajectories resulting in strange behavior	Non holo-nomic, Synchro-drive	No	12 msec cycle time, 66 MHz 486 PC
The Lane Curvature Method [67]	Yes	?	Dynamics yes Shape No	Local	Local min-ima	Non holo-nomic, Synchro-drive	No	8 Hz, 200 MHz Pentium Pro
The Extended Potential Field [69]	?	Environ-mental model with obstacles	Some dynamic	Global	Local Min-ima, wall fol-lowing and door passing has been improved	?	Yes, Knowl-edge about the obsta-cles required	?
The Schlegel Approach [68]	Yes, and map	Histogram grid updated with latest range infor-mation	Dynamics yes Shape Yes	Local thinking of algorithm. Global of entire sys-tem	Possibly susceptible to local min-ima	Synchro-Drive, Tri-cycle	No, unless map is to be used.	125 ms,? MHz 2,5 Mb lookup table

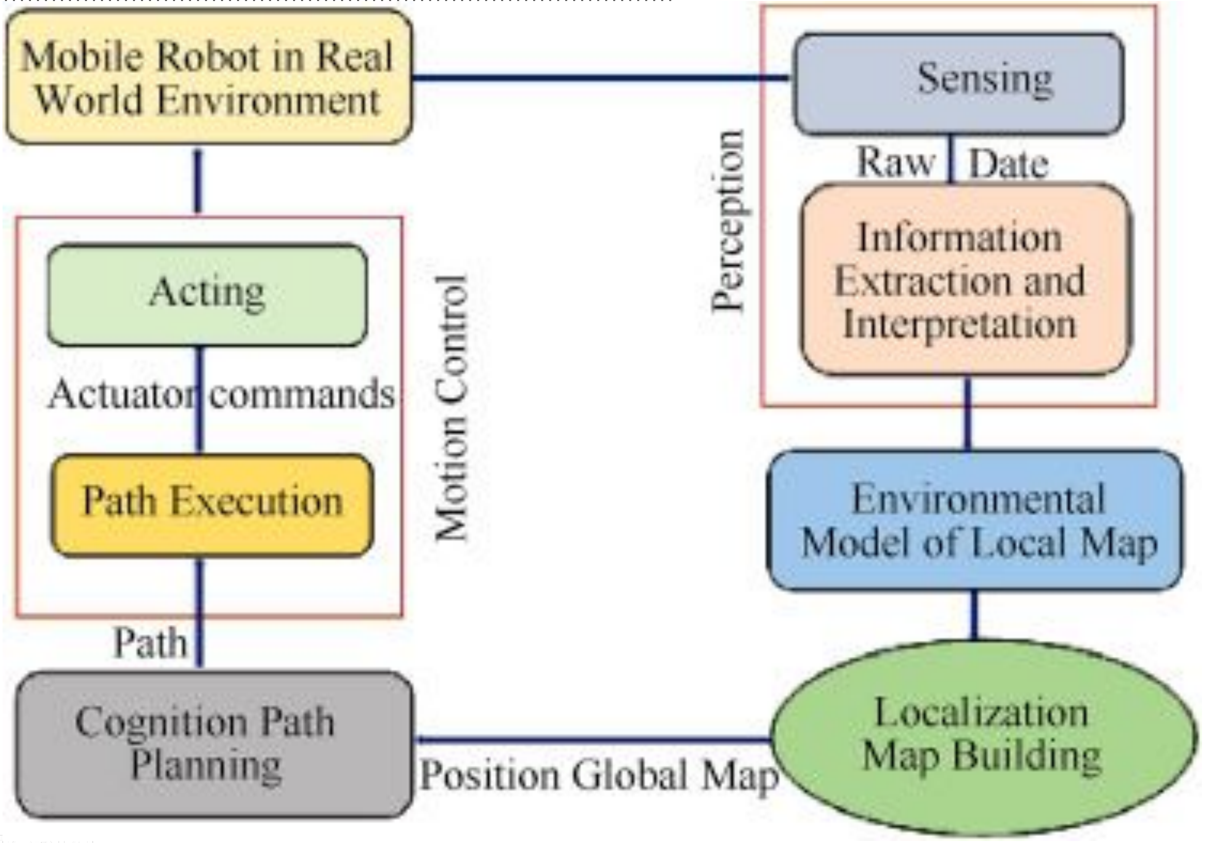
There is no right or wrong method to use. It depends on your use-case and how complex you want your robot's locomotion to be.

Architecture

A Basic Architecture



Slightly-Modified Complex Archi



Mobile Robot in Real World Environment
Sensing
Raw Date
Perception
Information Extraction and Interpretation
Acting
Actuator commands
Motion Control
Path Execution
Environmental Model of Local Map
Path
Cognition Path Planning
Position Global Map
Localization Map Building

Thanks

If you have doubts, google it.