Perception Driven Autonomous Vehicle:

A vehicle is designed to use locally perceived information in preference to potentially inaccurate map data to navigate a road network while obeying the traffic rules and regulations. Vehicles use a powerful and general purpose rapidly exploring Random Trees (RRT's) based planning algorithm to achieve the requirement of driving in lanes, following traffic signals, parking and other such functions.

Architecture of the Vehicle:

1. Road Paint detector: Uses two different image processing techniques for lane marking.
2. Obstacle Detector: Uses SICK and Velodyne LIDAR to identify static as well as moving obstacles.
3. Low lying hazard detector uses downward LIDAR to assess the drivability of the road along with the bumps and obstacles.
4. Fast vehicle detector uses millimeter wave radar to detect fast approaching vehicles.

Based on all these, the motion planner creates a kinodynamically suitable path that the car can follow.

A large number of low cost sensors are used at the time of design instead of a few expensive ones so that they can be attached to a lot of places in the car to give a wider Field of View (FOV). As all the motion planning is done real time, it is absolutely important to not rely on one or two sensors for the drivability data making the use of a large number of sensors all the more trivial.

Minimal reliance on GPS: GPS though a fairly useful technology is avoided in motion planning because GPS cannot be relied upon for high accuracy localization at all times. Though it is used when other techniques for motion planning fail and there's no way forward.

LIDAR based Obstacle Detection:

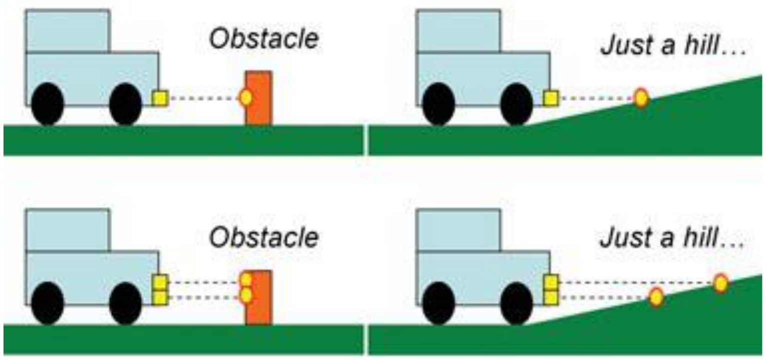
LIDAR is a remote sensing technology that measures distances by illuminating a target with a laser and analyzing the reflected light.

Autonomous vehicles use LIDAR for obstacle detection and avoidance to navigate safely through environments.

An autonomous vehicle featuring 12 SICK LIDAR's and a Velodyne LIDAR.

Fig: Raw data. Left: camera view of an urban scene with oncoming traffic. Middle: corresponding horizontal planar LIDAR data. Right: Velodyne data

A single planar LIDAR cannot reliably differentiate between obstacles and non-flat terrain. However with more than one planar LIDAR, an appreciable change in z can be measured. This strategy requires that any potential obstacle be observable by multiple planar LIDARs and that the LIDARs observe the object at different heights.



Obstacle or hill? With a single planar LIDAR, obstacles cannot be reliably discriminated from traversable (but hilly) terrain. Multiple planar LIDARs allow appreciable changes in *z* to be measured, resolving the ambiguity.

Sensor-Based Motion Planning for Car like Robot:

Autonomous navigation for car like robots is a challenging field because the car like robot must depend on only the local information from the sensor during the motion planning. Thus the navigation algorithm needs to guarantee that the robot approaches the goal location only using the local information of the environment. Another difficulty for navigation occurs because of the motion constraints, like the nonholonomic constraint that prevent the side slip motion in the perpendicular direction, and the minimum turning radius. In order to design a motion planner for the car-like robot, the nonholonomic motion generated from the motion planning algorithm should satisfy these motion constraints.

Compared to the two degree of freedom point robot, the car like robot is modeled with three degree of freedom including not only position but also orientation, and it also has a nonholonomic constraint.

In the paper published “Sensor based motion planning for a car like robot based on bug-family algorithms” by World academy of Science, Engineering and Technology, Vol: 6 2012-11-23, the instantaneous nonholonomic motion is based on the pure pursuit method, so that the motion constraints of the robot are satisfied.

During the motion-to-goal behavior, the robot moves toward the goal point by following the nonholonomic path. When the distance to the goal point begins to increase during the motion-to-goal behavior, the local minima is determined and the behavior switches to the wall-following behavior. The robot circumnavigates the obstacle boundary until the current distance to the goal becomes smaller than the recorded minimal distance to the goal point. Then the robot leaves the obstacle boundary by switching the behavior to the motion-to-goal.

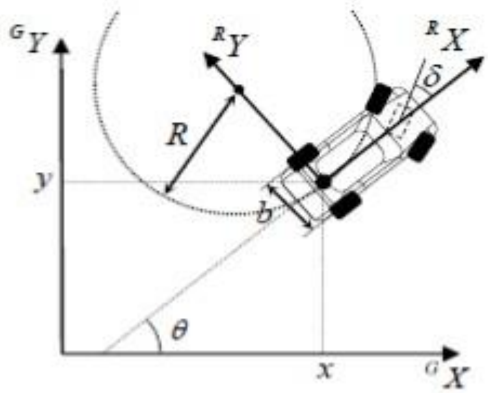


Fig. 1 Kinematic model of a car-like robot

aligned with the heading of the robot. If the position and the orientation of the car-like robot are expressed by (x, y) and θ, respectively, the configuration of the car-like robot can be defined as $\mathbf{q}=[x\ y\ \theta]^T$. Therefore, the kinematic model of the car-like robot is expressed by

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \tag{1}$$

where $\mathbf{u}=[v\ \omega]^T$ is the control input vector that contains the linear velocity v and the angular velocity ω. Here, the backward motion of the car-like robot is not considered in this work: that is, v is positive.

In this kinematic model, the motion of the car-like robot constrained by the following nonholonomic constraint is describes as:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{2}$$

That is, it is assumed that the car-like robot is reasonably slow such that the longitudinal traction and lateral force exerted on the tires do not exceed the maximum static friction between the tires and the floor. This is called a no-slip condition. At the low speed of the car-like robot, ***the kinematic steering is determined from the Ackerman turning geometry as δ=b/R=bκ where δ is the equivalent steering angle for the front wheels, b***

is the wheelbase, and R is the turning radius. Thus with the maximum steering angle for the front wheel δmax, the lower limit on the turning radius is represented by Rlim which satisfies δmax=b/Rlim.

From the laser rangefinder or LIDAR we can get the set of the end points of the obstacle boundary***. Let the set of the end points of the obstacle boundary be the set E. For example, in Fig E= {e1, e2, e3, e4}.***

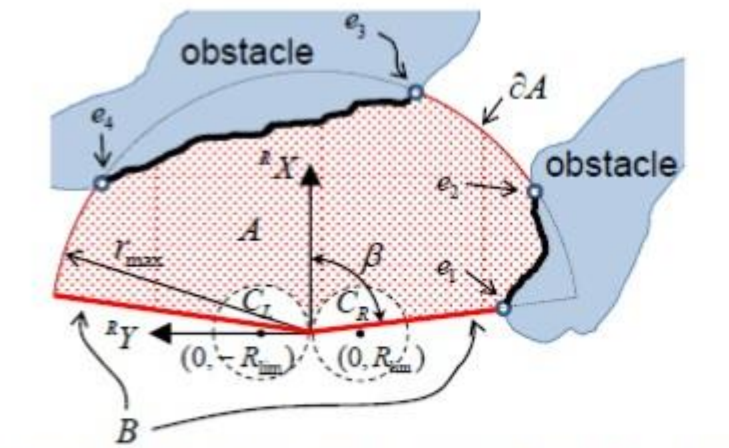


Fig. 2 Local view from 2D laser rangefinder of the robot

Geometry of Pure Pursuit Method:

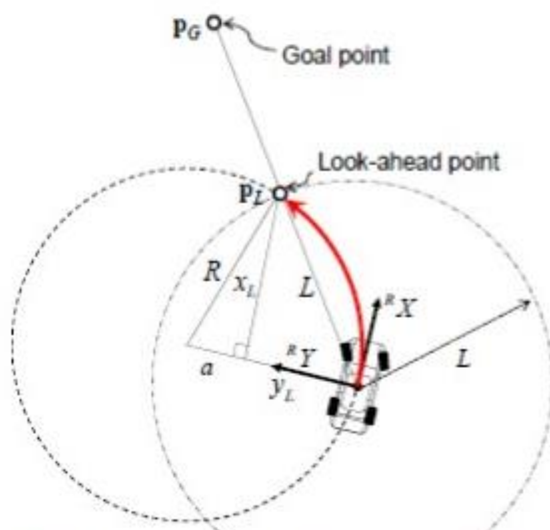


Fig. 3 Geometry of Pure pursuit method

Final results from the Pure Pursuit method:

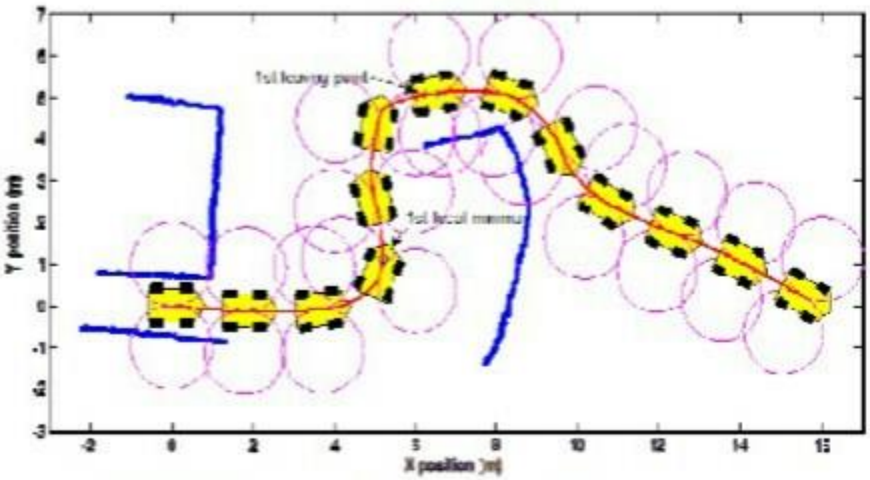


Fig. 11 Experimental results in simple environment

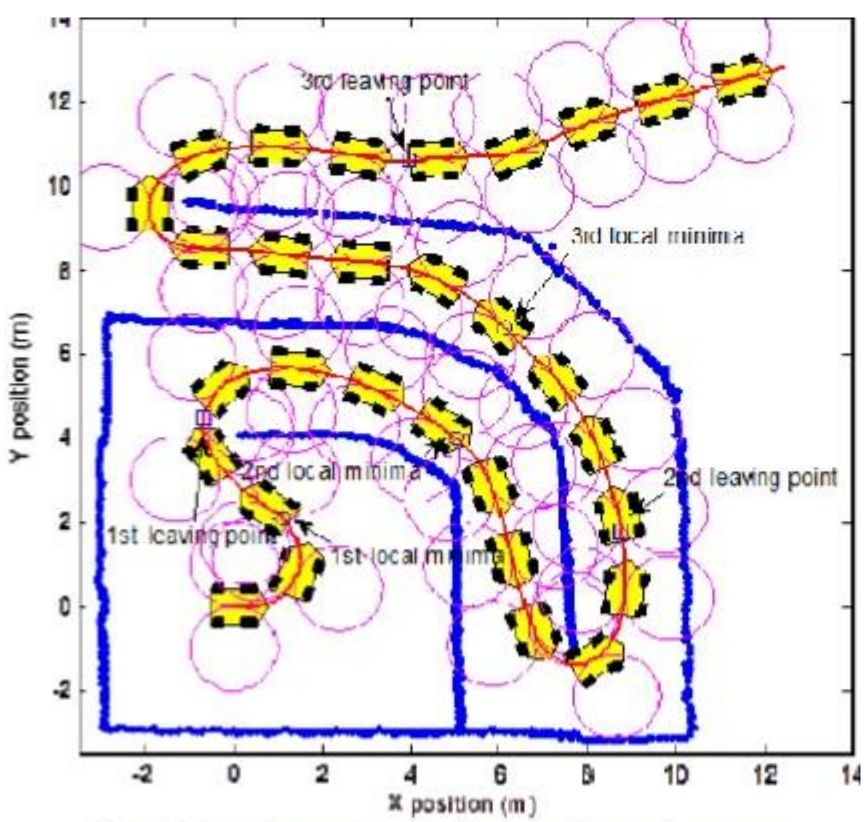


Fig. 13 Experimental results in maze-like environment