Planification pour les véhicules autonomes

David Filliat

Emmanuel Battesti

ENSTA ParisTech

The PAMU project

- An autonomous valet parking system inside Technocentre,
- A transport system using a fleet of electric vehicles used in open access,
- A reservation system via a web service.

- The vehicle can move without human beings on board to the booking point or parking / charging point.
- A human driver can drive the car outside the Technocentre.

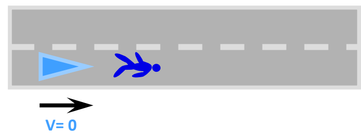
UNS Goals

The goal of the navigation module (UNS), which operates in autonomous mode, is to :

- Compute the shortest route to follow, from the starting point to the destination point.
- Generate local trajectories to follow this route and avoid dynamical obstacles

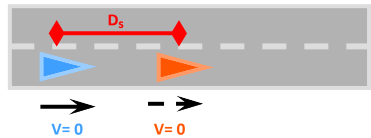
Use Cases

Pedestrian presence



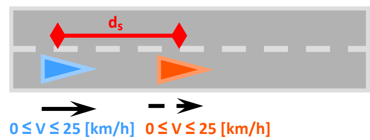
The vehicle stops.
It waits for a given time Tw, then requests for help

A stopped Vehicle in front



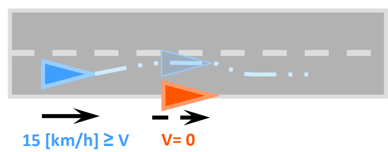
The vehicle stops.
It waits for a given time Tw, then requests for help

A vehicle moves in front



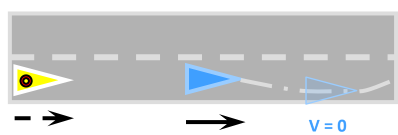
The vehicle adapts its speed.

Parked vehicle



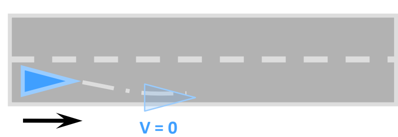
The vehicle pass it slowly.

Emergency vehicle



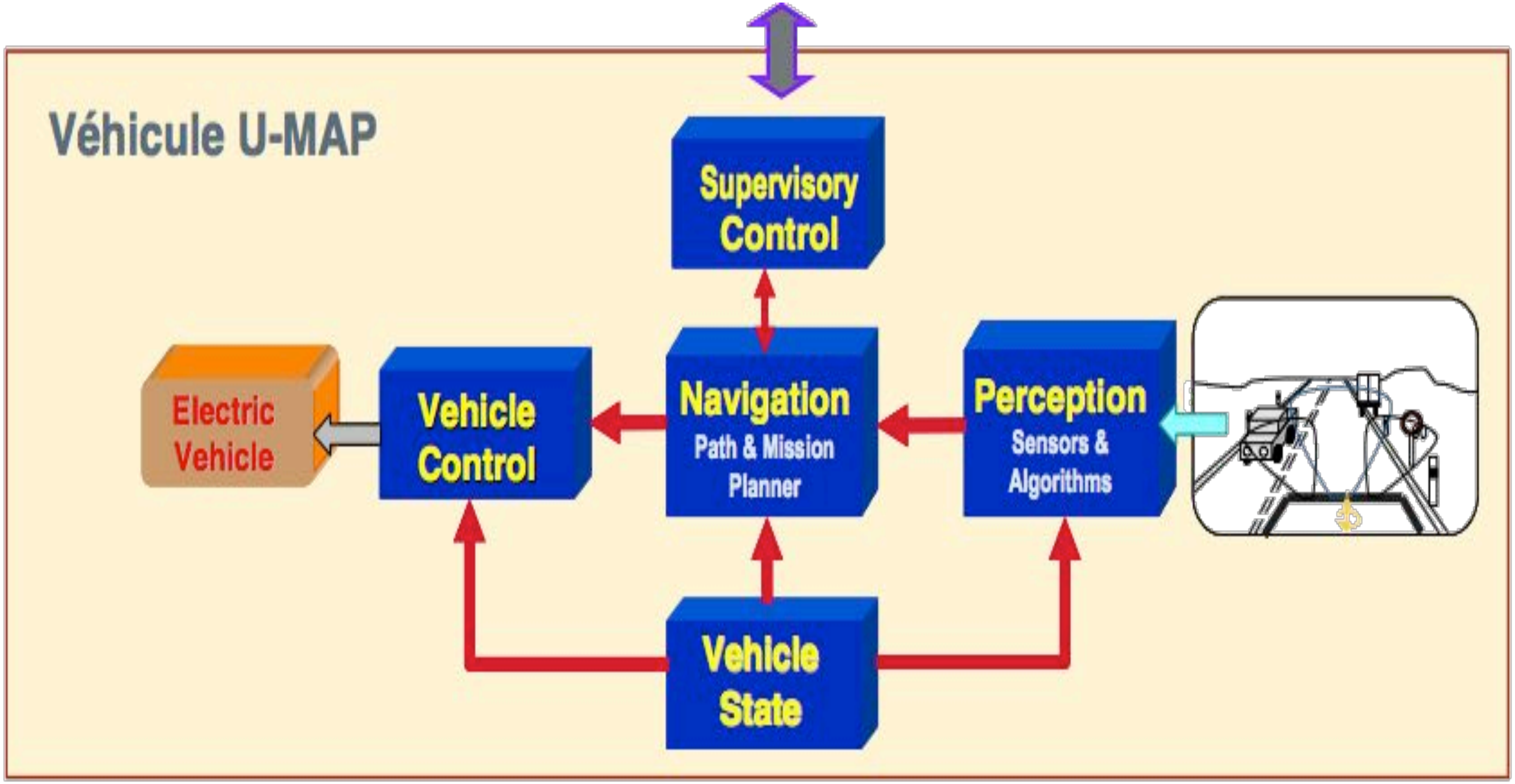
The vehicle moves to one side and wait until the emergency is cleared.

Pickup point



The vehicle and notifies that it is at destination.

Global Architecture



Interfaces : Localisation (ULS)

- ULS is an UNS **input**.
- It provides :
  - Position and orientation in the local frame
  - Speed et longitudinal acceleration
  - Wheel Angle
  - Information about road surface marking
  - Timestamp

Interfaces : Perception (UPS)

- UPS is an UNS **input**.
- It provides a list of obstacles with :
  - Position,
  - Dimensions,
  - Type,
  - Speed,
  - Timestamp

Interfaces : Control (UCS)

- UCS is an UNS **output** via CAN-BUS
- It sends at 10Hz a vector of 30 points which defines a local trajectory in the vehicle frame

For each point :

•X positions values
•Y positions values
•headings angle values
•speed values
•curvature values
•derivatives of the X positions values
•derivatives of the Y positions values
•derivatives of the headings angle values
•acceleration values
•double derivatives of the X positions values
•double derivatives of the Y positions values in vehicle
•double derivatives of the headings angle values
•derivative of the acceleration values

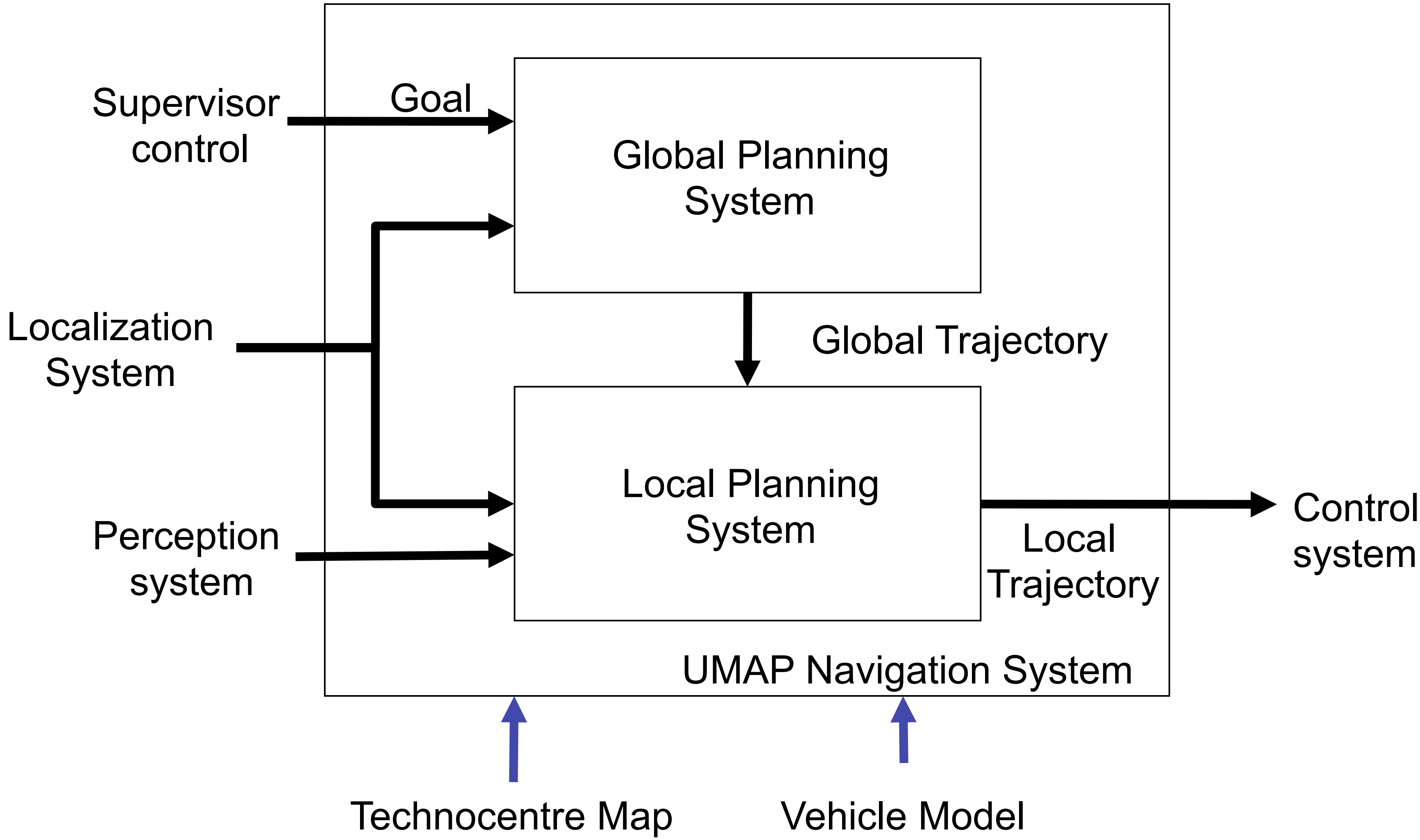
Interfaces : Supervisor (USCM)

- The UNS behavior is controlled by USCM.
- Are transmitted :
  - Message to change of state : standby, ready, working
  - Input command : shutdown, reset
  - Other messages output: PLANNING_DONE, TRAJECTORY_NOT_AVAILABLE, EMERGENCY

Navigation Architecture

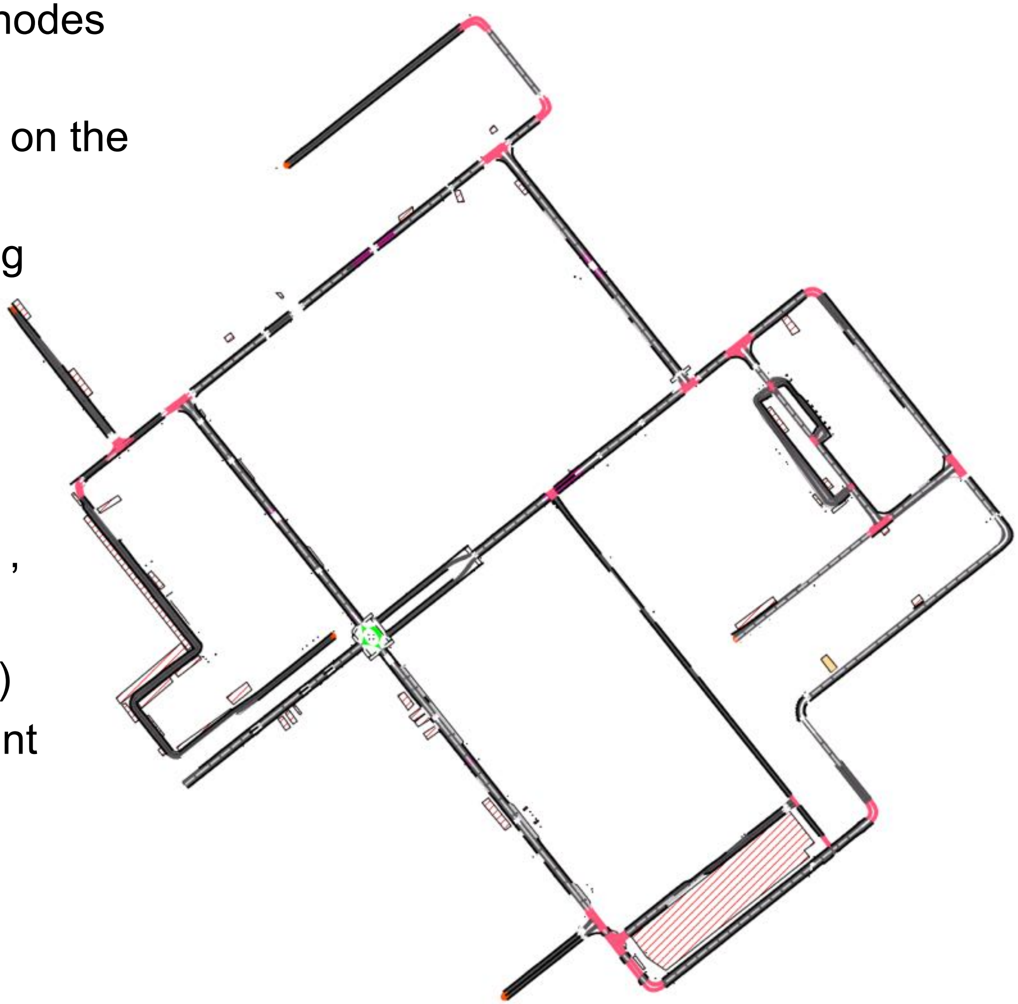
- **Two levels :**
  - Computing a **global path** : the shortest path to go from the actual position of the vehicle U-MAP to its destination,
  - Computing a **local path** then **local trajectory** to follow the global path, given :
    - The vehicle state (speed, wheel angle, position, etc.)
    - The local environment (obstacles, others vehicles, pedestrians, intersection, traffic regulations)

Navigation Architecture



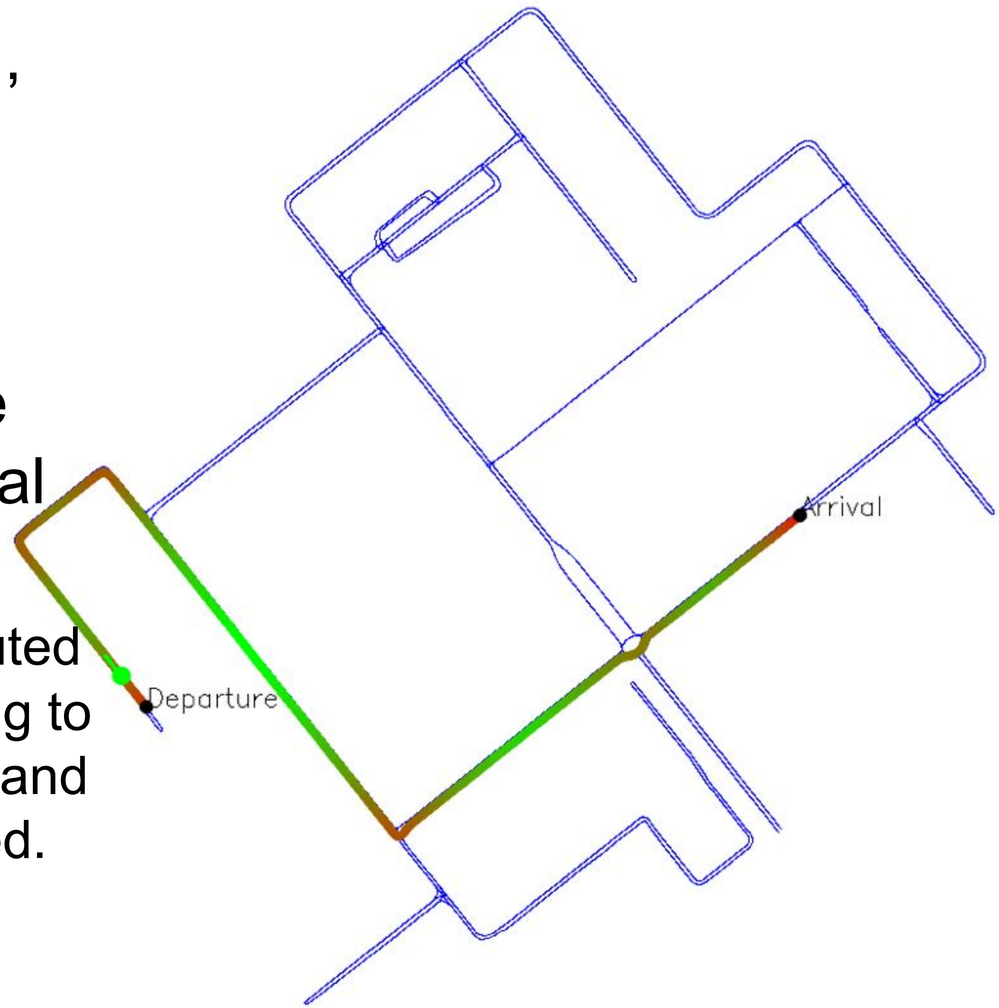
The global map

- A database that contains a graph of nodes and polygons.
- Each node corresponds to a position on the road, plus additional information.
- For each node, there are the following information:
  - Position,
  - List of adjacent nodes,
  - The direction of the lane,
  - Information about the lane width ,
  - Authorized speed ,
  - Road signs (signs and markings)
  - Information about the type of point (road, parking, pick-up point),
  - Layout of intersections.



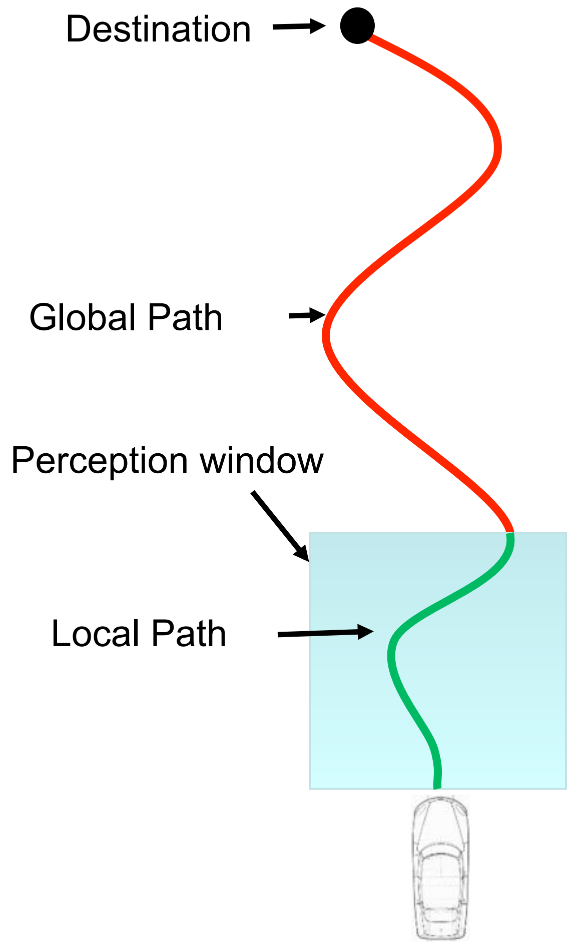
Global path

- From the vehicle position, which route to follow to reach your destination?
- A* algorithm provides the shortest path on the global map.
- Speed constraint are recomputed along the global path according to maximum lateral acceleration and maximum steering angle speed.
- Global path is smoothed to be feasible by a vehicle.

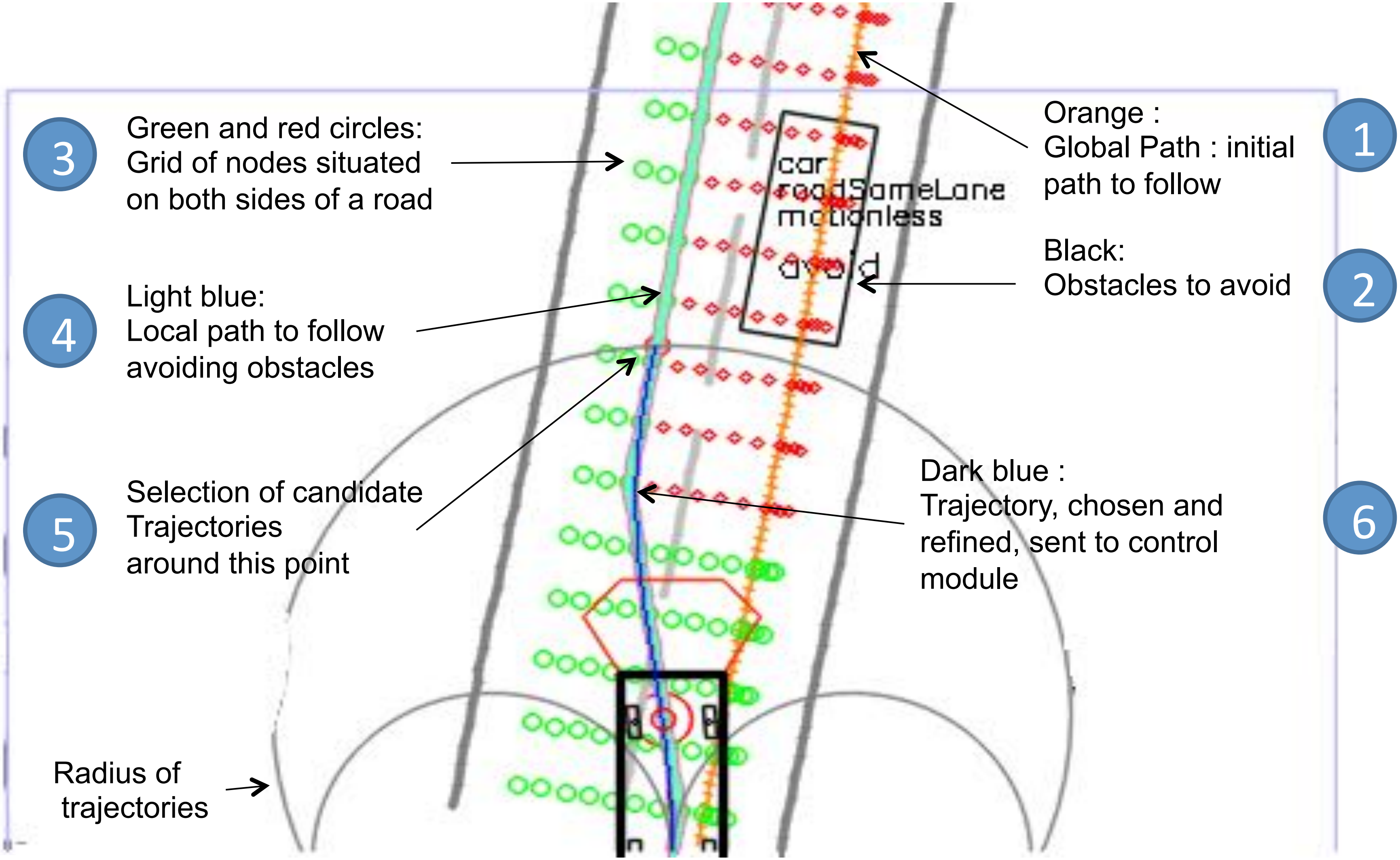


Local path

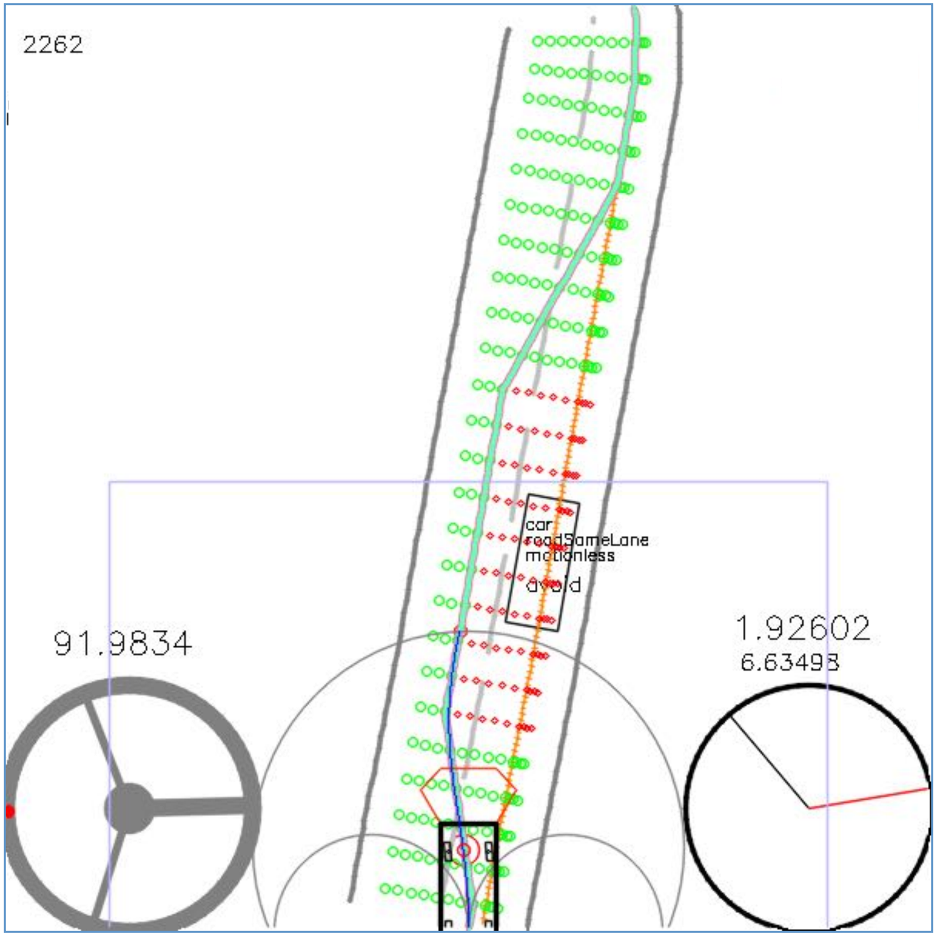
- A local path is extracted from the global path with a perception window
- This local path can be deformed to avoid dynamical obstacles recently perceived by the perception module

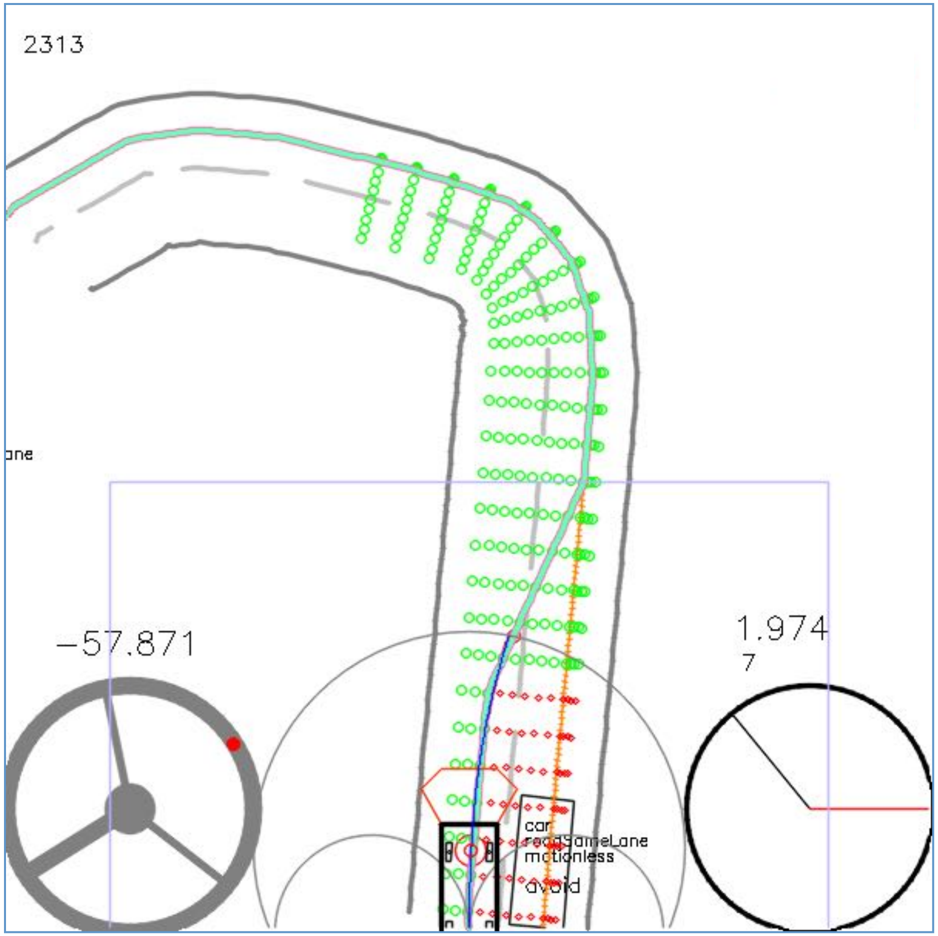


Local path generation



Local path generation





Trajectories database: Parameterization

- Nonholonomic vehicle : all motions are not feasible.
- Solution : using parametric polynomial spirals

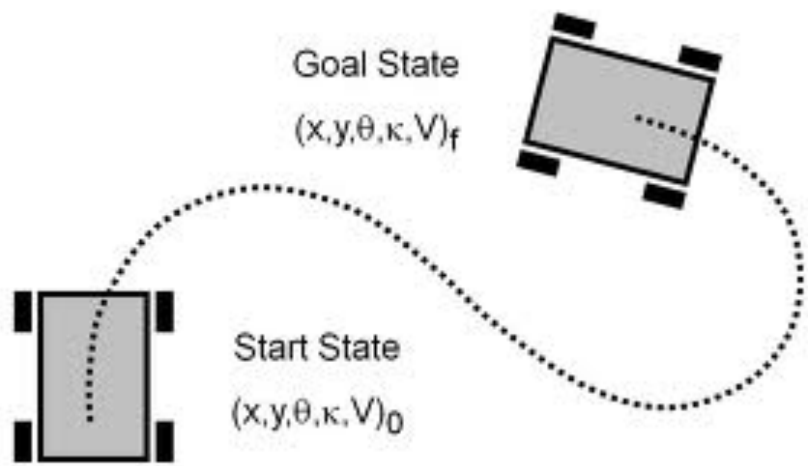
$$\kappa(s) = a + bs + cs^2 + ds^3$$

$$\theta(s) = as + \frac{bs^2}{2} + \frac{cs^3}{3} + \frac{ds^4}{4}$$

$$x(s) = \int_0^s \cos\left[as + \frac{bs^2}{2} + \frac{cs^3}{3} + \frac{ds^4}{4}\right] ds$$

$$y(s) = \int_0^s \sin\left[as + \frac{bs^2}{2} + \frac{cs^3}{3} + \frac{ds^4}{4}\right] ds$$

- From initial and desired final states of the vehicle, we can compute the parameters of the trajectories by numerical analysis
- Advantages :
  - Continuous values
  - Easily executed
  - Compact representation

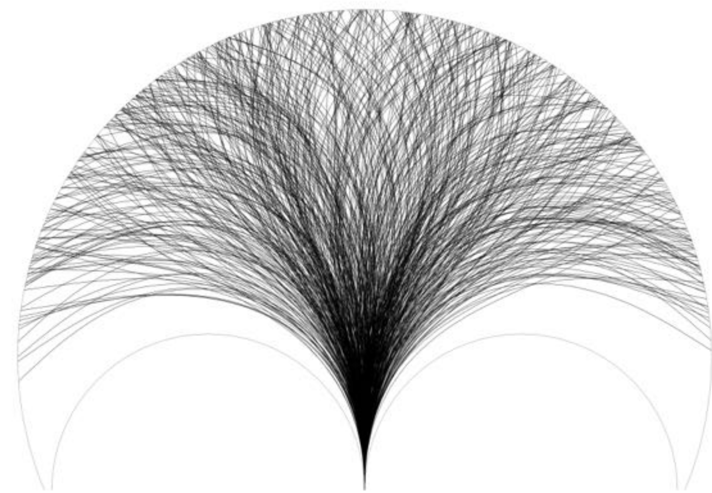


Trajectories database : Constraints

- max. speed: 5.56 m/s
- max. steering angle : 0.61959 rad
- max. steering angle speed
  - in standard driving : 0.698132 rad/s
  - in emergency driving : 6.98132 rad/s
  - in parking driving : 17.4533 rad/s
- max. curvature : 0.17857 1/m
- max. longitudinal acceleration
  - in standard driving: 1.96 m/s²
  - in emergency driving : 3.43 m/s²
- max. longitudinal decel.
  - in standard driving : 1.96 m/s²
  - in standard driving without comfort: 2.94 m/s²
  - in emergency driving (ABS) : 9.8 m/s²
- max. lateral acceleration
  - in standard driving : 1.96 m/s²
  - in emerg. driving : 4.9 m/s²
- max. sum of acceleration in std driving : 1.96 m/s²
- length of the vehicle : 4.749 m
- width of the vehicle : 1.809 m
- wheelbase of the vehicle: 2.701 m
- front overhang: 0.908 m
- rear overhang: 1.010 m
- gap between wheels: 1.160 m

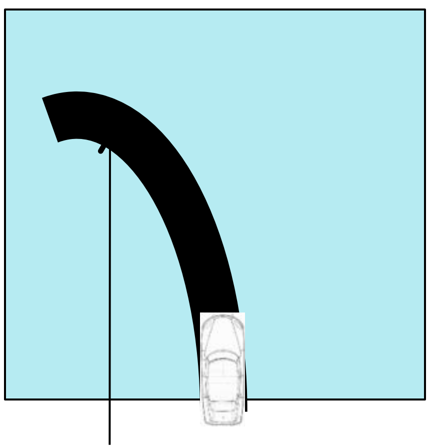
Trajectories database : Lookup table

- We store in a lookup table a set of pre-computed trajectories, whose goals are inside perception area (offline):
  - Decreasing time computation
  - Avoiding numerical instability
  - Easing obstacles avoidance
- Thus, online, we can sample in this lookup table to select desired trajectories



Trajectories database : inverted index

- Offline, for each pixel of obstacle in the occupancy grid, we compute the subset of trajectories from the lookup table subject to collision with this pixel => inverse index



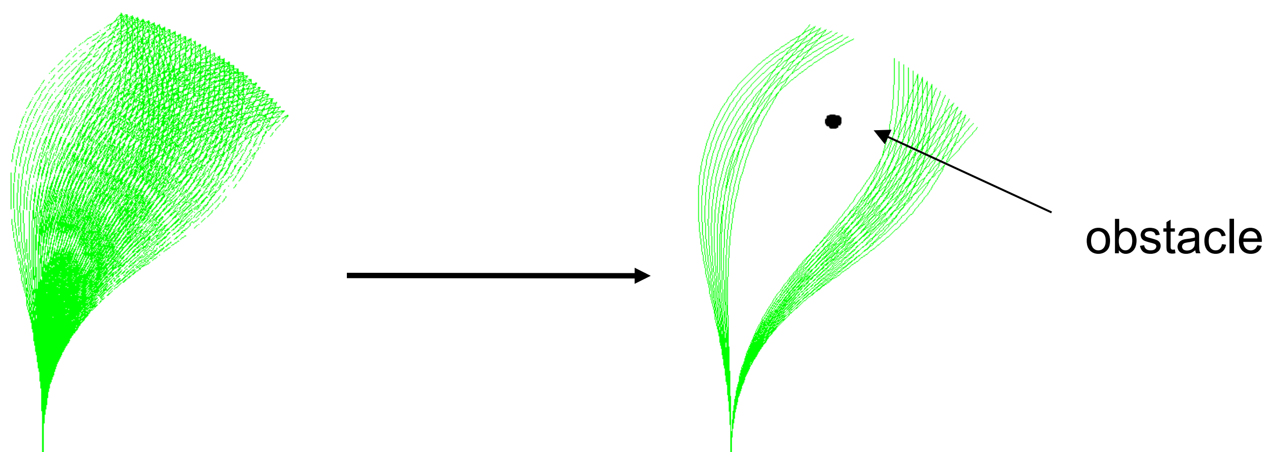
Footprint of the vehicle for trajectory 2578

Pixel 1
→ traj. : 10, 15, 29, 1026
Pixel 2
→ traj. : 10, 15, 30, 1024, 2578
Pixel 3
→ traj. : 11, 16, 30, 1024, 2578
…

For example, these trajectories are passing the vehicle by the pixel 2

Selection of the best trajectory

- Online, from all possible trajectories, we retain a subset which is close to the local path.
- Trajectories that pass through obstacles that must be overcome are removed via inverted index



Subset of possible trajectories

Subset of possible trajectories except those that cross obstacles

Selection of the best trajectory

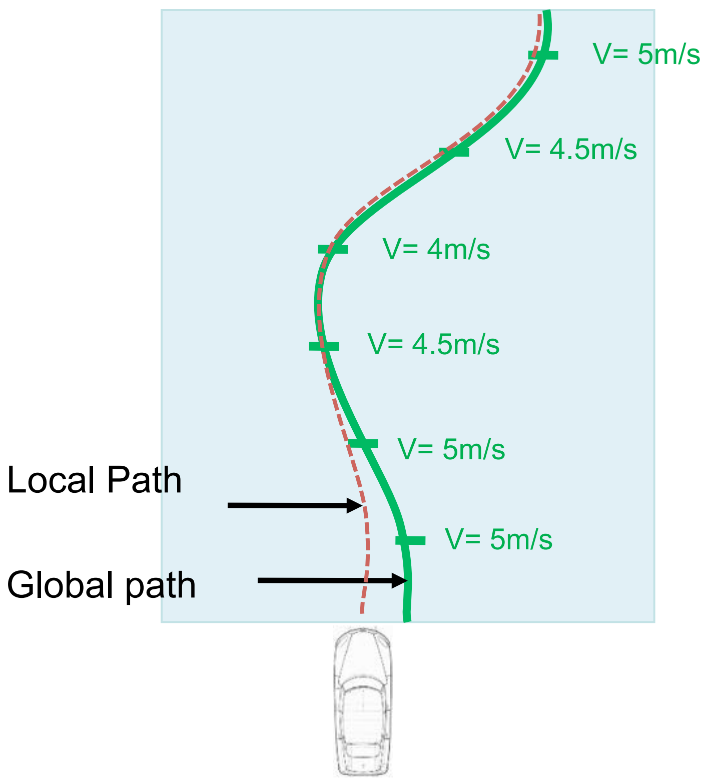
- Trajectories whose intrinsic maximum speed is less than the vehicle speed are penalized.
- The chosen trajectory is the one with the best score, computed according to:
  - The proximity of points of the trajectory with the local path
  - The proximity of points of the trajectory with the previous trajectory

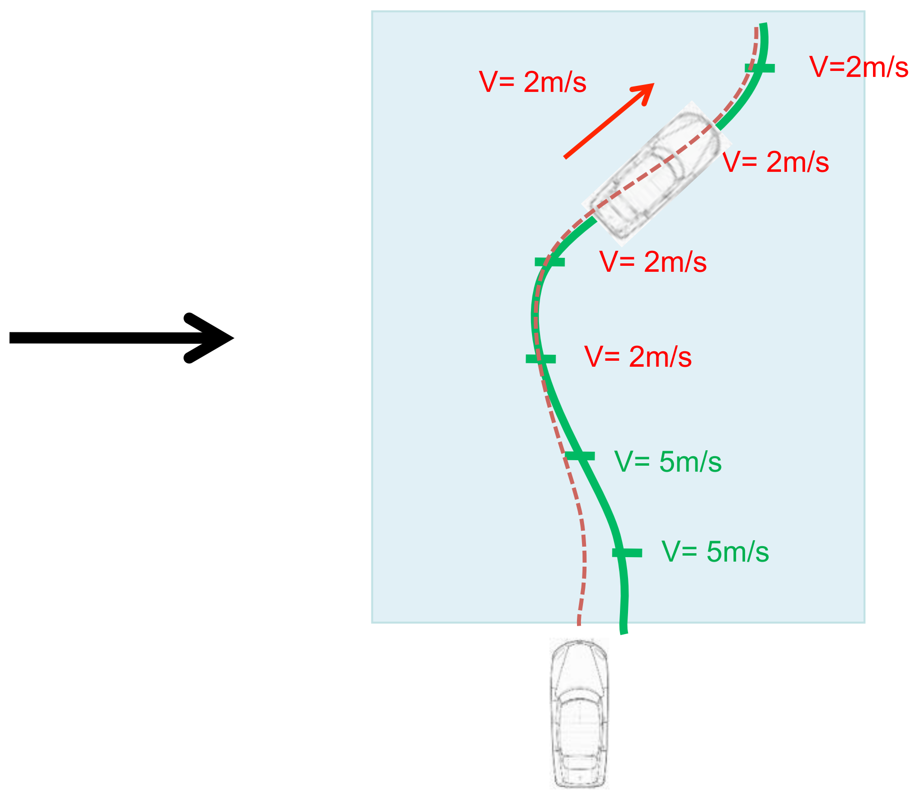
Obstacles management (1/3)

- Obstacles are divided into **two subgroups** of obstacles :
  - Obstacles that require to **change the speed** of U-MAP vehicle (pedestrians, stopped vehicle or slower vehicle) but should not deviate the vehicle.
  - Obstacles qui should **deviate the vehicle** (for example, parked car that is in the navigable space)

Obstacles management (2/3)

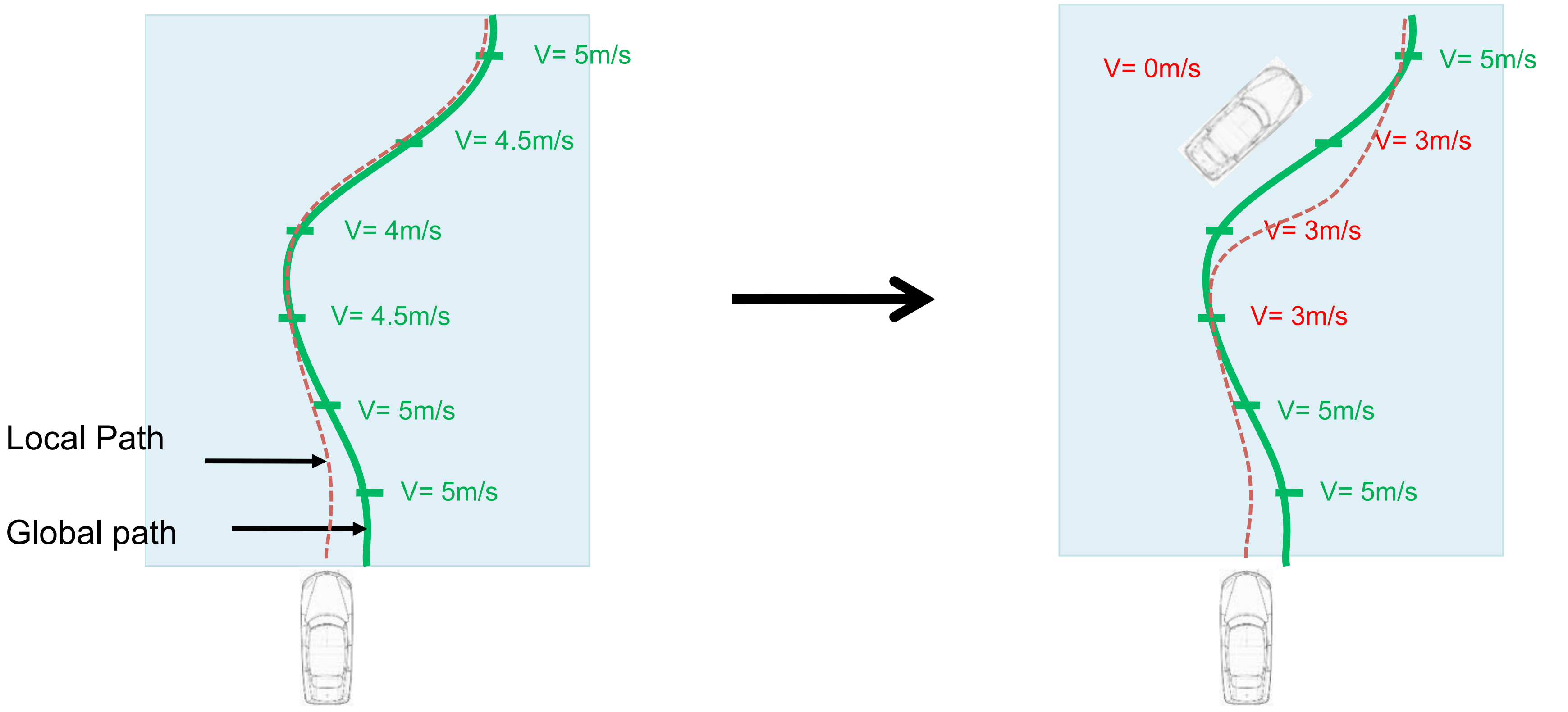
Obstacles changing the speed limit of the local path





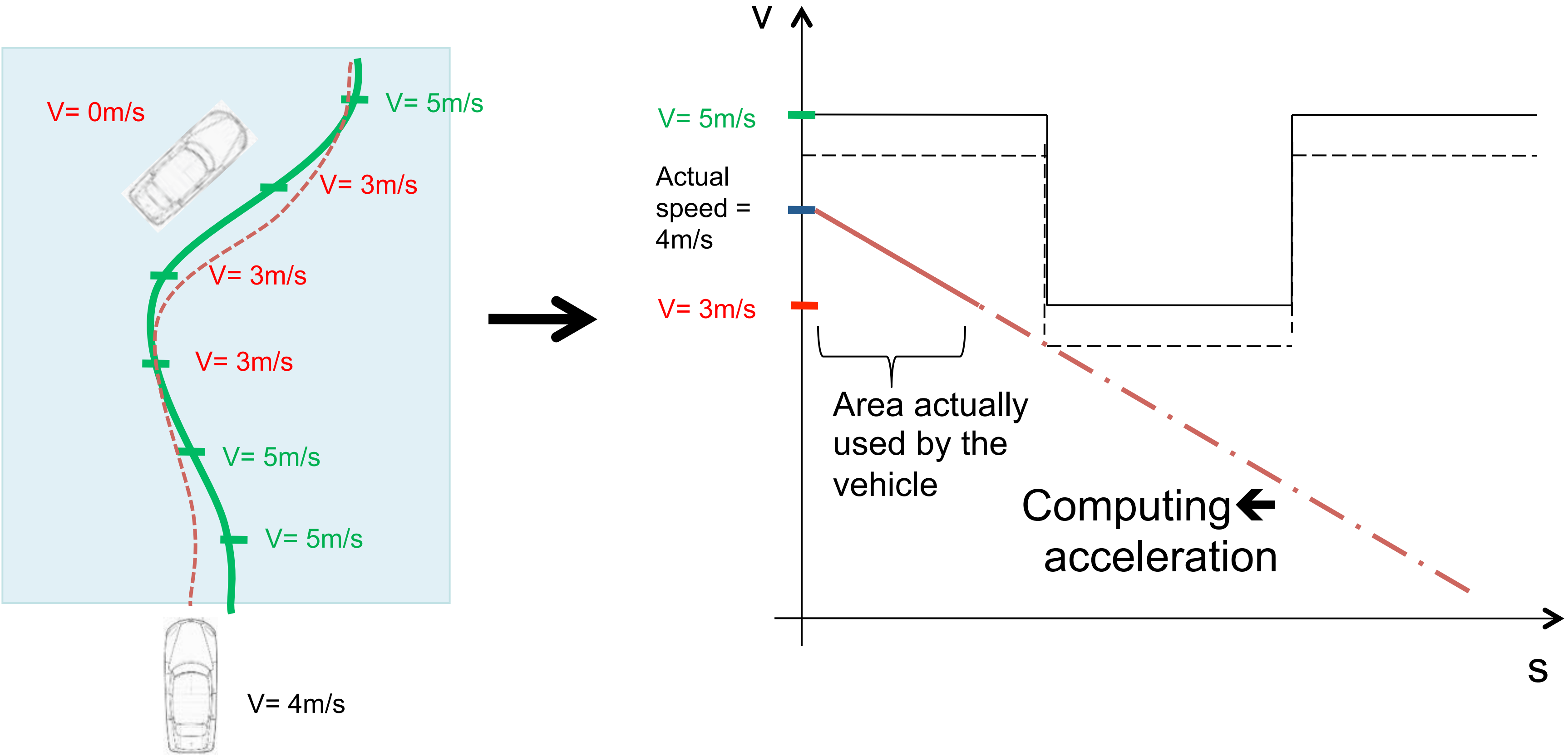
Obstacles management (3/3)

Obstacles changing the local path, the vehicle trajectory and the speed limit → use of the inverted index



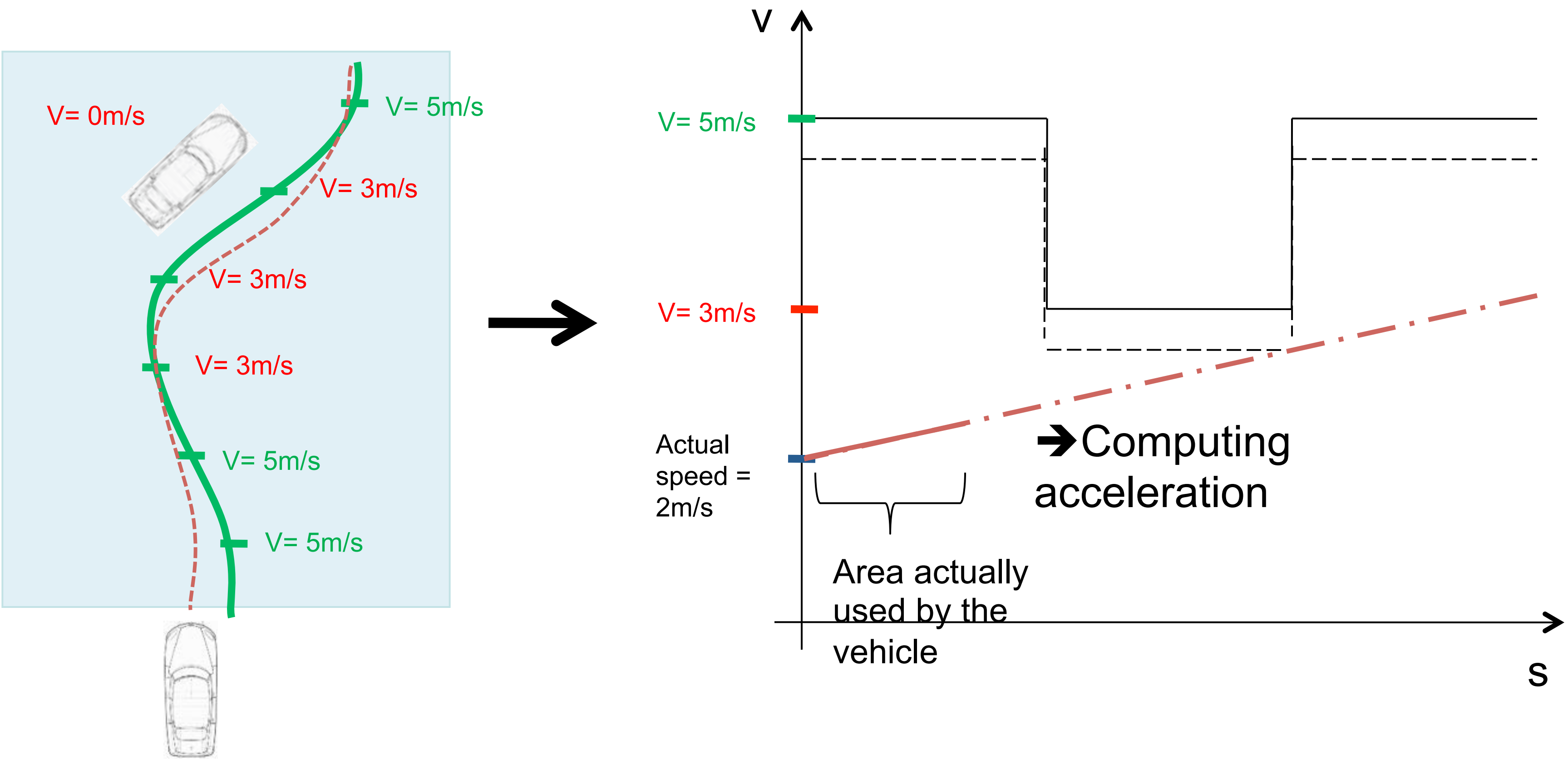
Speed profile

- Constant acceleration profile



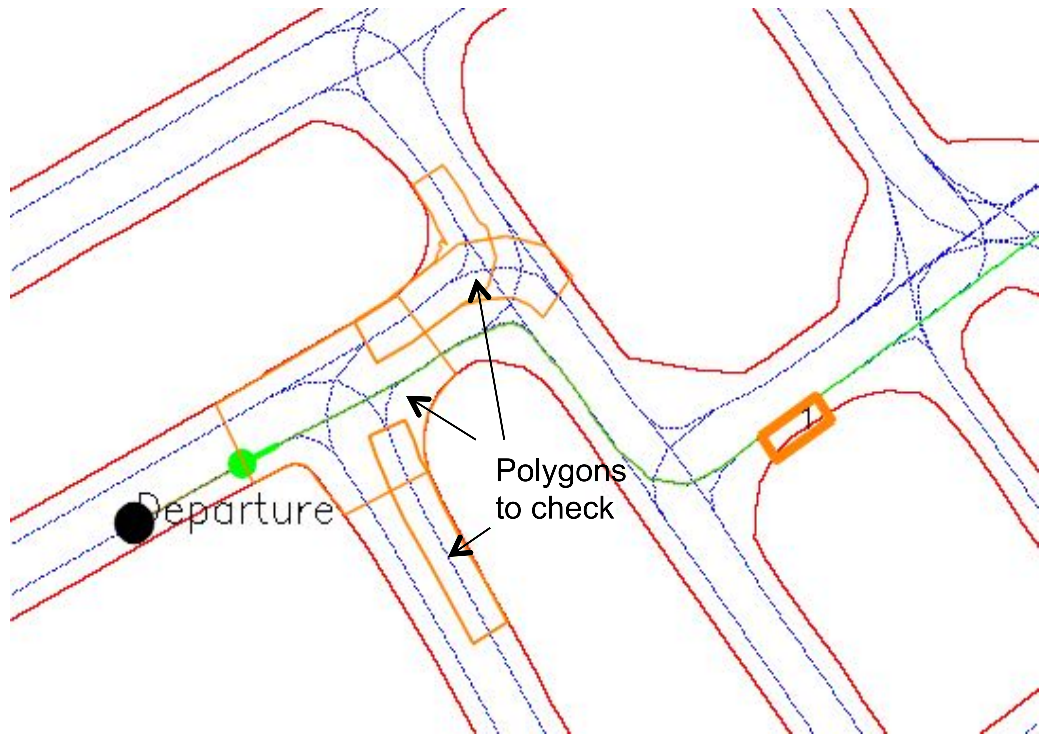
Speed profile

- Constant acceleration profile



Intersection management

- At each crossroad, we know the priority rules included in the global map.
- UNS computes polygons defining areas inside which it will check clearance
- The vehicle waits that there are no more dangerous obstacles inside polygons before proceeding.



Thank you for your attention.

Questions ?