4. Trajectory Planning
第4课：轨迹规划

金力 Li Jin
li.jin@sjtu.edu.cn
上海交通大学密西根学院
Shanghai Jiao Tong University UM Joint Institute

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Recap

- Longitudinal control
  - Speed tracking
  - Trajectory tracking
  - Vehicle following
- Lateral control
  - Lane keeping
  - Lane changing

Outline

- Single-vehicle on an empty road
  - 1 dimensional
  - 2 dimensional
- Multi-vehicle planning
  - Policy-free approach
  - Policy-based approach
- Perception
  - Onboard perception
  - Vehicle-to-vehicle connectivity
  - Vehicle-to-infrastructure connectivity

Single vehicle on an empty road

- 1 dimensional
- 2 dimensional

Objective

- We have a vehicle that needs to travel from an origin to a destination.
- For 1D, the orbit is known, but the spatio-temporal trajectory is to be determined.
- For 2D, the orbit is unknown, and the geometry of the orbit as well as the waypoint times are to be determined.

1D: Problem formulation

- Data
  - Vehicle parameters
  - Locations of the origin and the destination
- Decision variables
  - Spatio-temporal trajectory
  - Denoted by $\{x(t); t = 0,1,2, \dots\}$
- Constraints
  - Vehicle dynamics
- Objective
  - Option 1: reach the destination as soon as possible
  - Option 2: consume as little energy as possible
  - Option 3: a weighted sum of the above

1D: Problem formulation

- Data
  - Location of origin: $x = 0$ [m]
  - Location of destination: $x = D$ [m]
  - Maximal speed $\bar{v}$ [m/s]
  - Maximal acceleration (speed increment) $\bar{a}$ [m/s^2]
  - Fuel rate f [L/s]: $f = \alpha v^2$.
  - Rationale: air drag is proportional to v^2. (Let's use this simple model for now.)
  - Assume unit time step $\Delta t = 1$ [s]
- Decision variables
  - $v(t)$ for $t = 0,1,2, \ldots$
  - Speed will determine acceleration and position.

1D: Problem formulation

- We need to first resolve a technical issue: what is the dimension of v? (We cannot deal with an infinite set of decision variables $\{v(t); t = 0,1,2 \dots\}$ in practice.)
- We can address this issue by computing an apparent upper bound on the traverse time.
- Traverse time = how long it takes for the vehicle to move from the origin to the destination.
- This upper bound is typically determined by prior knowledge.
- Let T be the upper bound.
- So, the decision variables are $v(1), v(2), \dots, v(T)$

1D: Problem formulation

- Constraints are as follows.
  - Origin & destination

$$x(0) = 0$$
$$x(T) = D$$

  - Kinematic equation

$$x(t+1) = x(t) + v(t)$$
$$a(t) = v(t) - v(t-1)$$

  - Technological constraint (saturation)

$$-\bar{a} \le a(t) \le \bar{a}$$
$$0 \le v(t) \le \bar{v}$$

- Objective function
  - One-step fuel cost: $\alpha v(t)^2$
  - Cumulative cost: $\sum_{t=0}^{T} \alpha v(t)^2$

1D: Problem formulation

A standard optimization formulation:

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1,\dots,T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2,\dots,T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1,\dots,T.
\end{aligned}$$

Data? Decision variables? Constraints? Objective function?

1D: Optimization

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1, \dots, T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2, \dots, T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1, \dots, T.
\end{aligned}$$

- Mathematical classification
- Feasibility
- Existence of optimal solution(s)
- Computation of optimal solution(s)

1D: Optimization: Mathematical classification

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1,\dots,T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2,\dots,T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1,\dots,T.
\end{aligned}$$

- Linear constraints
- Quadratic objective function
- Classification: quadratic programming with linear constraints（线性约束二次规划）.

1D: Optimization: Mathematical classification

- Consider a **real-valued** optimization problem

$$\min f(x)$$
$$s.t.\ g(x) \leq 0$$
$$x \in \mathbb{R}$$

- If f and g are both linear, we have a linear programming（线性规划）.
- If f or g is quadratic **or linear** (but not both linear), we have a quadratic programming（二次规划）.
- If f or g is nonlinear, we have a nonlinear programming（非线性规划）.
- Linear programming is easy to solve, and nonlinear programming is in general hard (or not that easy).

1D: Optimization: Mathematical classification

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1, \dots, T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2, \dots, T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1, \dots, T.
\end{aligned}$$

- So, the 1D trajectory planning problem is a linearly-constrained quadratic programming.

1D: Optimization: Mathematical classification

- Consider a generic optimization problem

$$\min f(x)$$
$$s.t.\ g(x) \leq 0$$
$$x \in \mathbb{X}$$

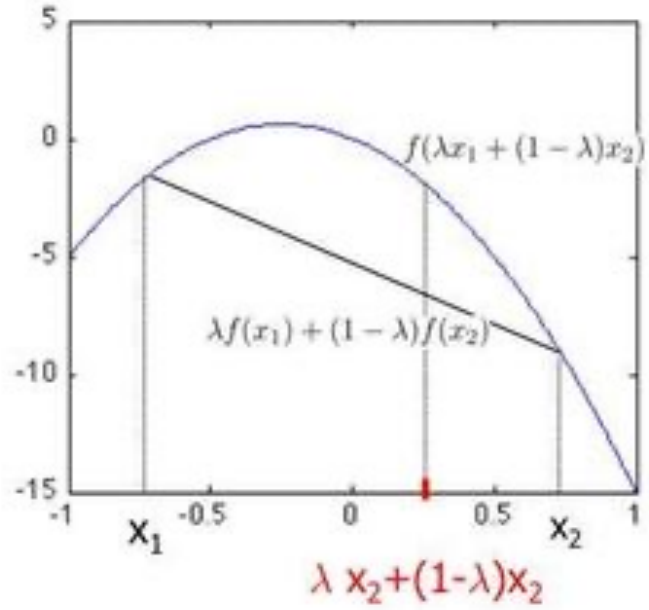
- An optimization problem may have not constraints -> unconstrained optimization.
- If $\mathbb{X}$ (the set of x) is the set of integers -> integer programming（整数规划）.
- If $\mathbb{X} = \mathbb{R}$ and if both f and g are convex in x -> convex optimization（凸优化）.

1D: Optimization: Mathematical classification

- Convex 凸 & concave 凹

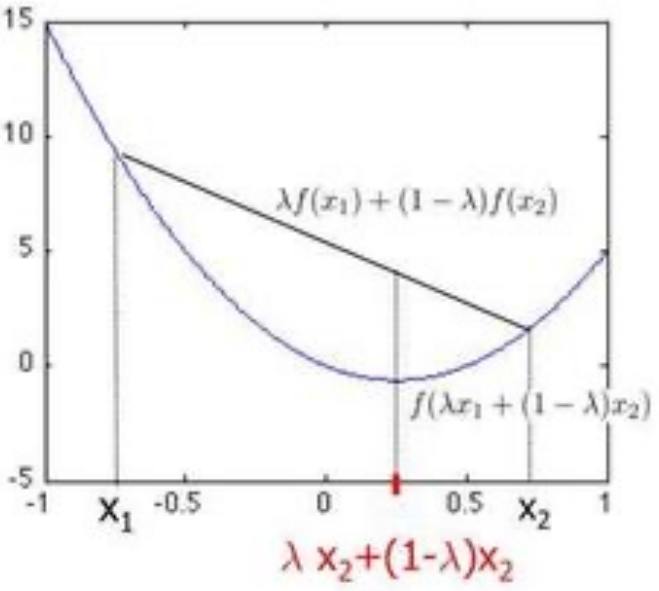
f is ***concave*** if and only

$$\forall x_1, x_2, \quad \forall \lambda \in (0,1),$$
$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda) f(x_2)$$



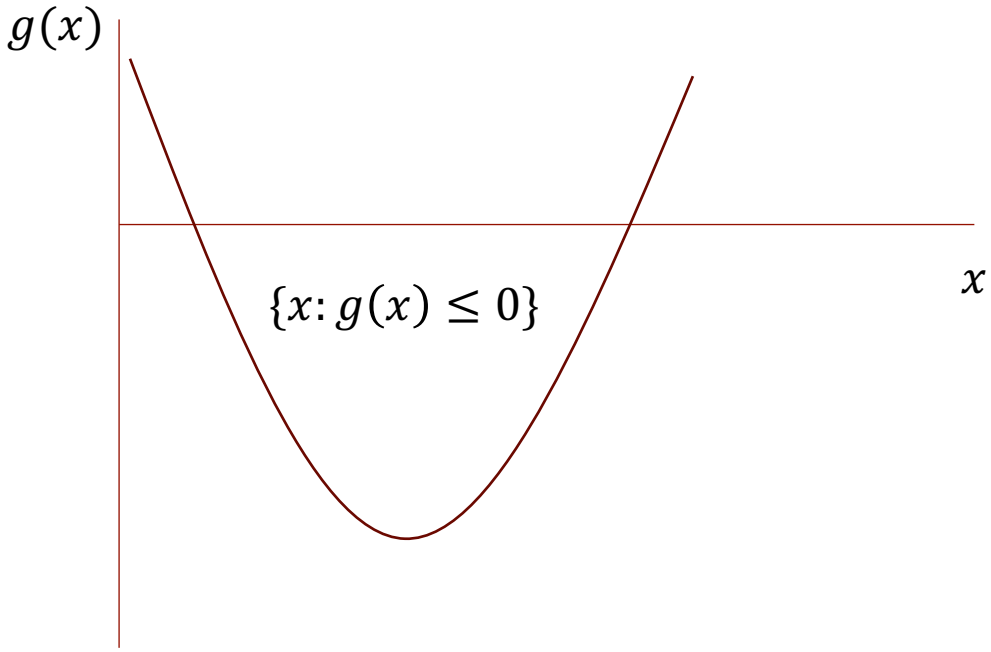
f is ***convex*** if and only

$$\forall x_1, x_2, \quad \forall \lambda \in (0,1),$$
$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$



1D: Optimization: Mathematical classification

- If $g(x)$ is convex in x, then $\{x: g(x) \leq 0\}$ is a convex set.
- Hence, $g(x) \leq 0$ is called the standard form of constraints



1D: Optimization: Mathematical classification

- Why do we care about convexity?
- Because we can solve an unconstrained convex optimization

$$\min f(x)$$

by solving

$$\nabla_x f(x) = 0$$

(if f is differentiable.)

- We can also solve a constrained convex problem in a systematic manner
  - An optimal solution is guaranteed.
  - Note: convex problem requires both convex objective function and convex constraints!

1D: Optimization: Mathematical classification

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1, \dots, T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2, \dots, T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1, \dots, T.
\end{aligned}$$

- The 1D trajectory planning problem is convex.
- So you can solve it using optimization tools in Python/MATLAB…

1D: feasibility

- Recall that the trajectory planning problem is subject to the following constraints

$$x(0) = 0, x(T) = D,$$
$$x(t+1) = x(t) + v(t), t = 0,1, \dots, T-1,$$
$$-\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2, \dots, T,$$
$$0 \le v(t) \le \bar{v}, t = 0,1, \dots, T.$$

- Any set of values for $\{v(t); t = 1,2, \dots, T\}$ satisfying the above are called a feasible solution（可行解）.
- If there exists at least one feasible solution, the problem is said to be feasible.

1D: feasibility

- When would the problem be infeasible?

$$x(0) = 0, x(T) = D,$$
$$x(t+1) = x(t) + v(t), t = 0,1, \dots, T-1,$$
$$-\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2, \dots, T,$$
$$0 \le v(t) \le \bar{v}, t = 0,1, \dots, T.$$

- T too small; D too large
- $\bar{a}$ too small; $\bar{v}$ too small
- Practical interpretation?

1D: existence of optimal solution(s)

- Definition: consider an optimization problem

$$\min f(x)$$
$$s.t.\ g(x) \leq 0$$

A feasible solution x^* is said to be an optimal solution if for any feasible solution x', we have

$$f(x^*) \leq f(x').$$

- Example:

$$\min x^2 \text{ s.t. } x \in \mathbb{R}^2$$

has a unique optimal solution

$$x^* = [0\ 0]^T$$

1D: existence of optimal solution(s)

- A feasible optimization problem does not always have an optimal solution!
- Counter example 1:

$$\min x \qquad \text{s.t. } x \leq 0$$

- Counter example 2:

$$\min \frac{1}{x} \qquad \text{s.t. } x \geq 1$$

- For most engineering problems, if your formulation is consistent with reality, then an optimal solution must exist.
- Otherwise, your formulation is wrong.

1D: computation of optimal solution(s)

- Within the scope of this course, you can solve the trajectory planning problem

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

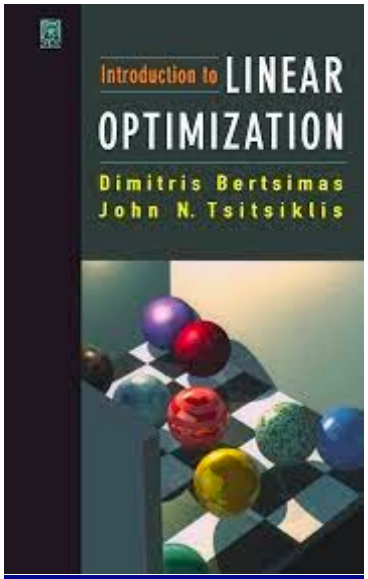
$$\begin{aligned}
\text{s.t.} \quad & x(0) = 0, x(T) = D, \\
& x(t+1) = x(t) + v(t), t = 0,1,\dots,T-1, \\
& -\bar{a} \le v(t) - v(t-1) \le \bar{a}, t = 1,2,\dots,T, \\
& 0 \le v(t) \le \bar{v}, t = 0,1,\dots,T.
\end{aligned}$$

using Python/MATLAB.

- You just need to know how to code the optimization model and how to run the optimization tool.

1D: computation of optimal solution(s)*

- Beyond the scope of this course, there is extensive theory on computing optimal solutions.
- *People spend career on this topic.*
- Analytical approaches: exactly characterize the structure and properties of optimal solutions
- Numerical approaches: use iterative algorithms to asymptotically search the optimal solutions
- Learning-based approaches: observe the optimal solution of similar problems and predict the most likely optimal solution



2D: Problem formulation

- Data
  - Vehicle parameters
  - Locations of the origin and the destination
- Decision variables
  - Spatio-temporal trajectory
  - Denoted by $\{x(t) \in \mathbb{R}^2; t = 0,1,2, \dots\}$
- Constraints
  - Vehicle dynamics
- Objective
  - Fuel + time

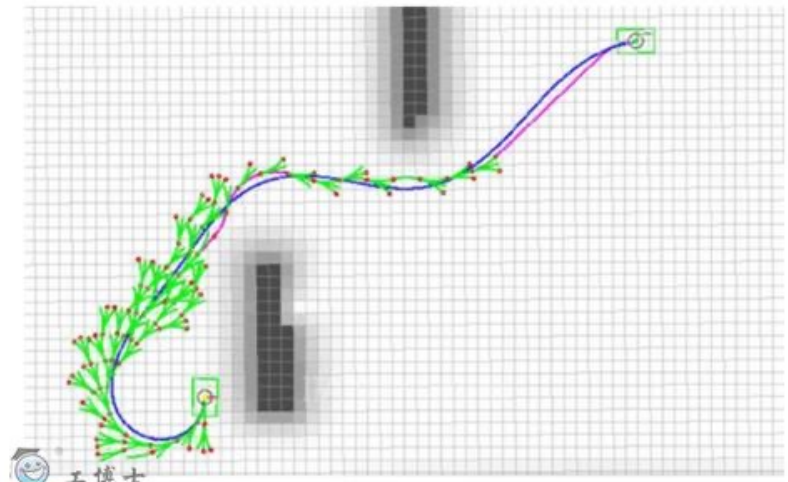
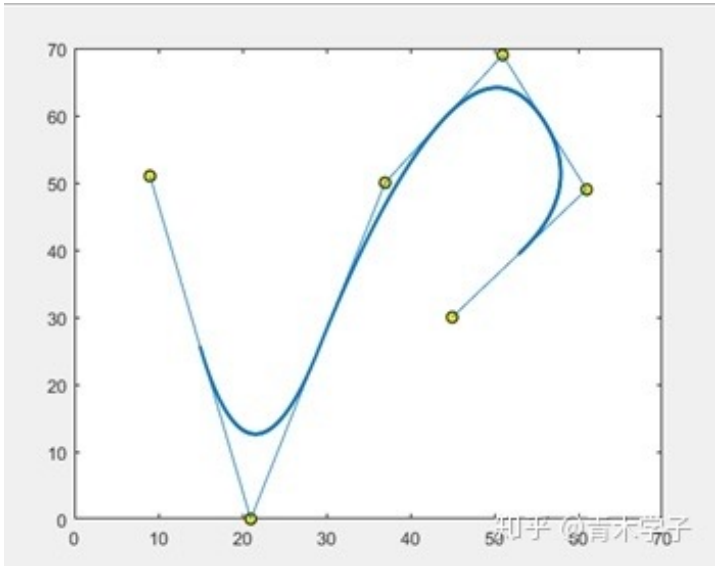


图2 启发式搜索到最优路径



2D: Problem formulation

- Data
  - Location of origin: $x = [0,0]$ [m]
  - Location of destination: $x = [D_1, D_2]$ [m]
  - Road space $\mathcal{X}$
  - Maximal speed $\bar{v}$ [m/s]
  - Maximal acceleration (speed increment) $\bar{a}$ [m/s^2]
  - Maximal angular speed $\bar{\omega}$ [rad/s]
  - Maximal angular acceleration $\bar{\phi}$ [rad/s^2]
  - Fuel rate f [L/s]: $f = \alpha v^2$.
  - Rationale: air drag is proportional to v^2. (Let's use this simple model for now.)
  - Assume unit time step $\Delta t = 1$ [s]

2D: Problem formulation

- Decision variables
  - Speed $v(t)$ for $t = 0,1,2, \ldots$
  - Angular speed $\omega(t)$ for $t = 0,1,2, \ldots$
  - Speed & angular speed will determine acceleration and position.
- Objective function
  - One-step fuel cost: $\alpha v(t)^2$
  - Cumulative cost: $\sum_{t=0}^{T} \alpha v(t)^2$

2D: Problem formulation

- Constraints are as follows.
  - Origin & destination & road space

$$x(0) = [0,0], \qquad \theta(0) = \theta_0$$
$$x(T) = [D_1, D_2]$$
$$x(t) \in \mathcal{X}$$

  - Kinematic equation

$$x(t+1) = x(t) + \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \end{bmatrix}$$
$$\theta(t+1) = \theta(t) + \omega(t)$$
$$a(t) = v(t) - v(t-1)$$
$$\phi(t) = \omega(t) - \omega(t-1)$$

  - Technological constraint (saturation)

$$-\bar{a} \le a(t) \le \bar{a}, \qquad 0 \le v(t) \le \bar{v}$$
$$-\bar{\phi} \le \phi(t) \le \bar{\phi}, \qquad -\bar{\omega} \le \omega(t) \le \bar{\omega}$$

2D: Problem formulation

A standard optimization formulation:

$$\min \sum_{t=0}^{T} \alpha v(t)^2$$

$$\text{s.t.} \quad x(0) = [0,0],\ \theta(0) = \theta_0, x(T) = [D_1, D_2],\ x(t) \in \mathcal{X},$$

$$x(t+1) = x(t) + \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \end{bmatrix},$$

$$\theta(t+1) = \theta(t) + \omega(t), a(t) = v(t) - v(t-1),$$

$$\phi(t) = \omega(t) - \omega(t-1),$$

$$-\bar{a} \le a(t) \le \bar{a},\ 0 \le v(t) \le \bar{v}$$

$$-\bar{\phi} \le \phi(t) \le \bar{\phi},\ -\bar{\omega} \le \omega(t) \le \bar{\omega},$$

Multiple vehicles

- Policy-free method
- Policy-based method

Policy-free method

- Suppose that a group of n AVs are traveling on a road section.
- Preserving order 1,2,….,n (#1 leads)
- Positions $x(t)$, speeds $v(t)$
- At each time t, a road-side unit (RSU) can observe the positions and speeds of all AVs

$$x(t) = [x_1(t), \dots, x_N(t)]$$
$$v(t) = [v_1(t), \dots, v_N(t)]$$

- One-time decision ("optimization"):
  - Given $x(0), v(0)$
  - Determine $x(t), v(t), t = 1,2, \dots T$

Policy-free method

- Data:
  - Initial and final positions $x(0) = 0, x(T) = D$
  - Vehicle properties
    - Maximal speed $\bar{v}$
    - Maximal acceleration (speed increment) $\bar{a}$
    - Safety distance d
- Decision variables
  - Time series of position $x(1), x(2), \ldots, x(T)$
  - Time series of speed $v(1), v(2), \ldots, v(T)$
- Constraints
  - Maximal speed/acceleration
  - Minimal distance
- Objective: min time + fuel

Policy-free method

$$\min \sum_{t=0}^{T}\sum_{i=1}^{n}\left(c_1 \mathbb{I}_{\{x_i(t)<D\}} + c_2 v_i^2(t)\right)$$

s.t. $x_i(t+1) = x_i(t) + v_i(t),$

$$i = 1, \dots, n, t = 0, \dots, T-1$$

$$0 \le v_i(t) \le \bar{v},\ i = 1, \dots, n, t = 1, \dots, T$$

$$|v_i(t) - v_i(t-1)| \le \bar{a},\ i = 1, \dots, n, t = 1, \dots, T$$

$$x_i(t) - x_{i+1}(t) \ge d,\ i = 1, \dots, n-1, t = 1, \dots, T$$

$$x(T) = [D\ D\ \dots D]^T$$

Nonlinearities:

1. Indicator function $\mathbb{I}_{\{x_i(t)<D\}} = \begin{cases} 1 & x_i(t) < D \\ 0 & otherwise \end{cases}$
2. Quadratic cost $v_i^2(t)$

Policy-free method

- Assumptions for policy-free decision making:
  - Everything is fully predictable and controllable
  - Everything is determined at one time (i.e. at $t = 0$)
  - No adjustment is needed or possible once the decision is made
- However, the above assumptions do not always hold!
  - Vehicles' movement may be perturbed by a variety of factors.
  - A decision seemingly good for now may not be valid later.
  - As error accumulates, the vehicles may no longer be able to implement the instructions.
- *刻舟求剑：notch the boat in search of the sword*
  https://en.wiktionary.org/wiki/%E5%88%BB%E8%88%9F%E6%B1%82%E5%89%91

Policy-based method

- Suppose that a group of n AVs are traveling on a road section.
- Preserving order 1,2,….,n (#1 leads)
- Positions $x(t)$, speeds $v(t)$
- At each time t, a road-side unit (RSU) can observe the positions and speeds of all AVs

$$x(t) = [x_1(t), \dots, x_N(t)]$$
$$v(t) = [v_1(t), \dots, v_N(t)]$$

- Real-time decision ("control"):
  - Given $x(0), v(0)$
  - Determine a policy μ: $(x, v) \mapsto v$ such that $v(t+1) = \mu(x(t), v(t))$

Policy-based method

- Data:
  - Final positions $x(T) = D$
  - Current state $x(t), v(t)$
  - Vehicle properties
    - Maximal speed $\bar{v}$
    - Maximal acceleration (speed increment) $\bar{a}$
    - Safety distance d
- Decision variable
  - Control policy μ such that $v(t+1) = \mu\big(x(t), v(t)\big)$
- Constraints
  - Maximal speed/acceleration
  - Minimal distance
- Objective:
  - Easy one: ensure that every vehicle arrives at destination (feasible)
  - Hard one: minimize the cost-to-go (optimal)

Policy-based method

$$\min \sum_{t=0}^{T} \sum_{i=1}^{n} \left(c_1 \mathbb{I}_{\{x_i(t)<D\}} + c_2 v_i^2(t) \right)$$

$$\text{s.t.} \quad x_i(t+1) = x_i(t) + v_i(t), \quad i = 1, \dots, n, t = 0, \dots, T-1$$

$$v(t+1) = \mu\big(x(t), v(t)\big), \ t = 1, \dots, T-1$$

$$0 \le v_i(t) \le \bar{v}, \ i = 1, \dots, n, t = 1, \dots, T$$

$$|v_i(t) - v_i(t-1)| \le \bar{a}, \ i = 1, \dots, n, t = 1, \dots, T$$

$$x_i(t) - x_{i+1}(t) \ge d, \ i = 1, \dots, n-1, t = 1, \dots, T$$

$$x(T) = \begin{bmatrix} D \\ \vdots \\ D \end{bmatrix}$$

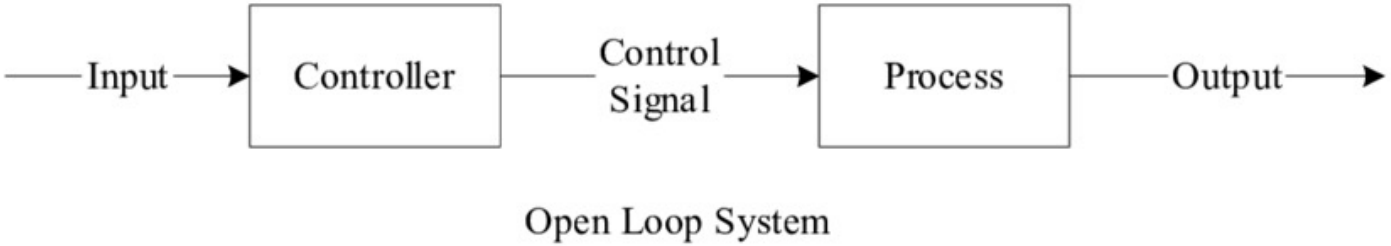
Policy design

- By "finding a policy", we mean two things
  1. Determine the form of the policy, i.e., linear/nonlinear/threshold-based…
  2. Determine the parameters of the policy, e.g. coefficients in a linear/nonlinear/threshold-based function…
- Optimizing the form of the policy is usually hard.
  - You cannot quantify the form of a function, so there is no way that you can formulate the form as a decision variable.
  - A sometimes feasible method is to conjecture or construct an optimal form via intuition, and then prove its optimality.
- Optimizing the parameters is also usually hard…
  - You first need to somehow restrict the form…
  - Leads to nonlinearity and even non-convexity
  - Heuristics, approximation, etc.

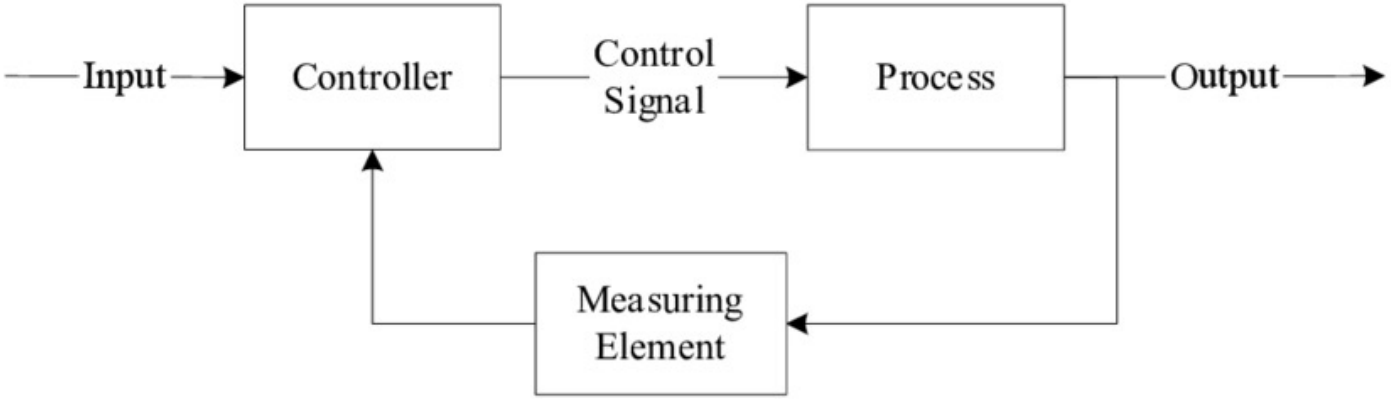
Policy-free vs. policy-based

- If actual $x_a(t), v_a(t)$ deviate from nominal values $x(t), v(t)$,
  - A policy-free approach will ignore such deviation and continue implementing $v(t+1)$ as pre-specified.
  - A policy-based approach will select the speed input according to $v(t+1) = \mu\big(x_a(t), v_a(t)\big)$, i.e. in response to the actual state.
- We can also say that
  - policy-free -> open-loop or non-feedback
  - policy-based -> closed-loop or feedback

Policy-free vs. policy-based



Open Loop System



Closed Loop System

Policy-free vs. policy-based

Policy-free

+ Requires less hardware

+ More flexible

+ Immune to corrupted or falsified data

- Deteriorates as error accumulates

- Not adaptive to unexpected disruptions

- Less intuitive

Policy-based

- Requires more hardware (observation & computation)

- Less flexible (restricted by policy)

- Vulnerable to corrupted or falsified data

+ Keeps suppressing error

+ Adaptive to unexpected disruptions

+ More intuitive

Perception 感知

- Onboard perception
- Vehicle-to-vehicle connectivity
- Vehicle-to-infrastructure connectivity

Perception 感知

Motivation

- Autonomous navigation has been a difficult problem for traditional vision and robotic techniques, primarily because of the noise and variability associated with real world scenes.
- Autonomous navigation systems based on traditional image processing and pattern recognition techniques often perform well under certain conditions but have problems with others.
- Part of the difficulty stems from the fact that the processing performed by these systems remains fixed across various driving situations.

Autonomous Land Vehicle In a Neural Network

- Pomerleau, D. A. (1989). ALVINN: An autonomous land vehicle in a neural network. *Carnegie-Mellon Univ Pittsburgh PA Artificial Intelligence and Psychology Project*.

ALVINN:
AN AUTONOMOUS LAND VEHICLE IN A NEURAL NETWORK

Dean A. Pomerleau
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

ALVINN (Autonomous Land Vehicle In a Neural Network) is a 3-layer back-propagation network designed for the task of road following. Currently ALVINN takes images from a camera and a laser range finder as input and produces as output the direction the vehicle should travel in order to follow the road. Training has been conducted using simulated road images. Successful tests on the Carnegie Mellon autonomous navigation test vehicle indicate that the network can effectively follow real roads under certain field conditions. The representation developed to perform the task differs dramatically when the network is trained under various conditions, suggesting the possibility of a novel adaptive autonomous navigation system capable of tailoring its processing to the conditions at hand.

Why NN

- Artificial neural networks have displayed promising performance and flexibility in other domains characterized by high degrees of noise and variability
- ALVINN (Autonomous Land Vehicle In a Neural Network) is a connectionist approach to the navigational task of road following.
- Specifically, ALVINN is an artificial neural network designed to control the NAVLAB, the Carnegie Mellon autonomous navigation test vehicle.

Architecture for NN

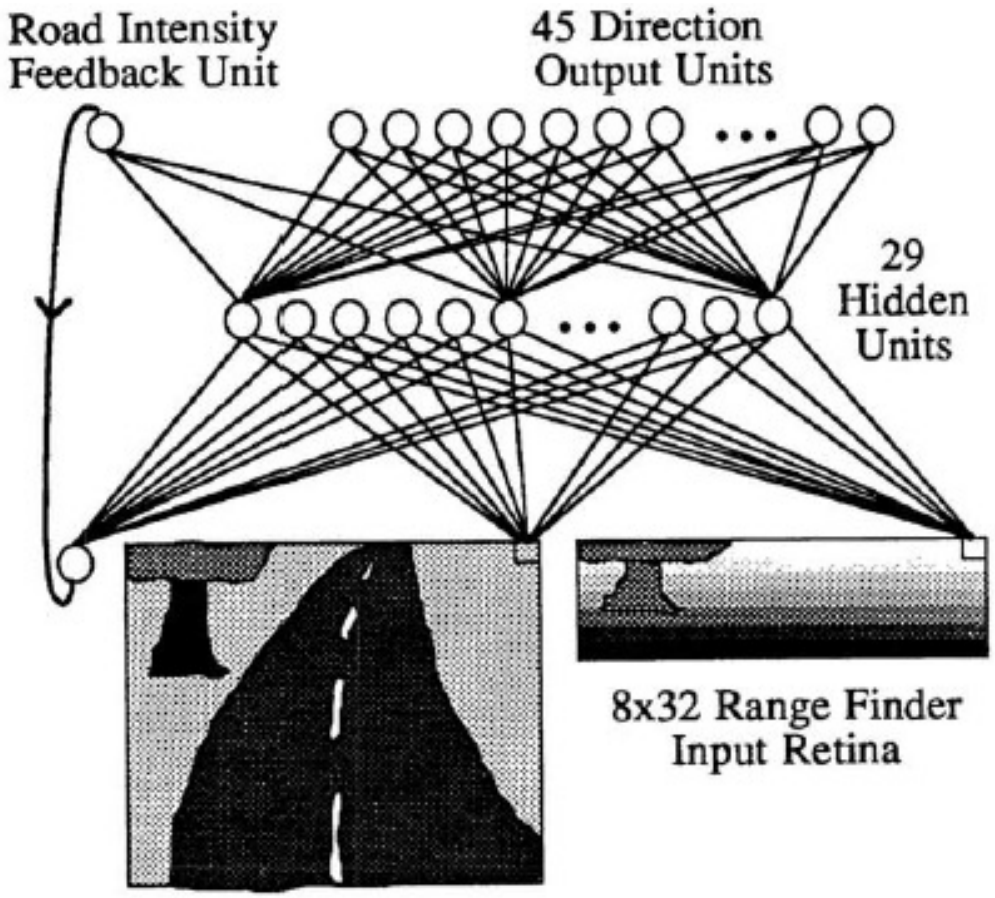


Figure 1: ALVINN Architecture

Input: video

- Input: sensory input available on the autonomous vehicle; video and range information.
- Camera sensor receives video camera input from a road scene.
- The activation level of each is proportional to the intensity in the blue color band of the corresponding patch of the image.
- The blue band of the color image is used because it provides the highest contrast between the road and the non-road

Input: range

- Also receives input from a laser range finder.
- The activation level of each unit is proportional to the proximity of the corresponding area in the image.
- The road intensity feedback unit indicates whether the road is lighter or darker than the non-road in the previous image.
- 1217 input units fully connected to the hidden layer of 29 units, which is in tum fully connected to the output layer

Output

- The output layer consists of 46 units, divided into two groups.
- The first set of 45 units is a linear representation of the turn curvature along which the vehicle should travel in order to head towards the road center.
- The middle unit represents the "travel straight ahead" condition while units to the left and right of the center represent successively sharper left and right turns.
- The final output unit is a road intensity feedback unit which indicates whether the road is lighter or darker than the non-road in the current image
- Parameters to determine: the activation levels in the hidden units

Test

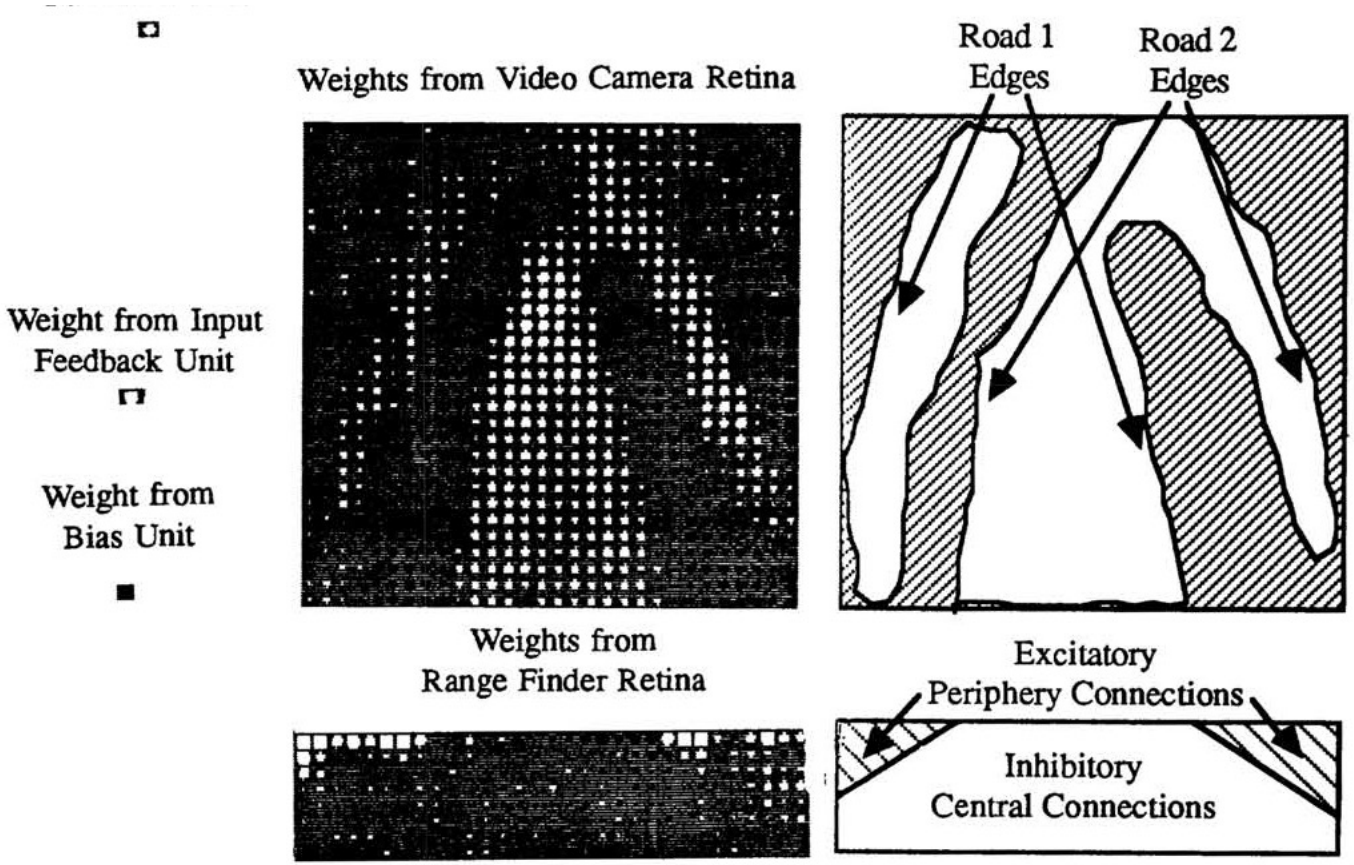


Figure 4: Diagram of weights projecting to and from a typical hidden unit in a network trained on roads with a fixed width. The schematics on the right are aids for interpretation.

State of the art

- Gao, H., Cheng, B., Wang, J., Li, K., Zhao, J., & Li, D. (2018). Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, *14*(9), 4224-4231.

Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment

Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li

***Abstract*—This paper presents an object classification method for vision and light detection and ranging (LIDAR) fusion of autonomous vehicles in the environment. This method is based on convolutional neural network (CNN) and image upsampling theory. By creating a point cloud of LIDAR data upsampling and converting into pixel-level depth information, depth information is connected with Red Green Blue data and fed into a deep CNN. The proposed method can obtain informative feature representation for object classification in autonomous vehicle environment using the integrated vision and LIDAR data. This method is also adopted to guarantee both object classification accuracy and minimal loss. Experimental results are presented and show the effectiveness and efficiency of object classification strategies.**

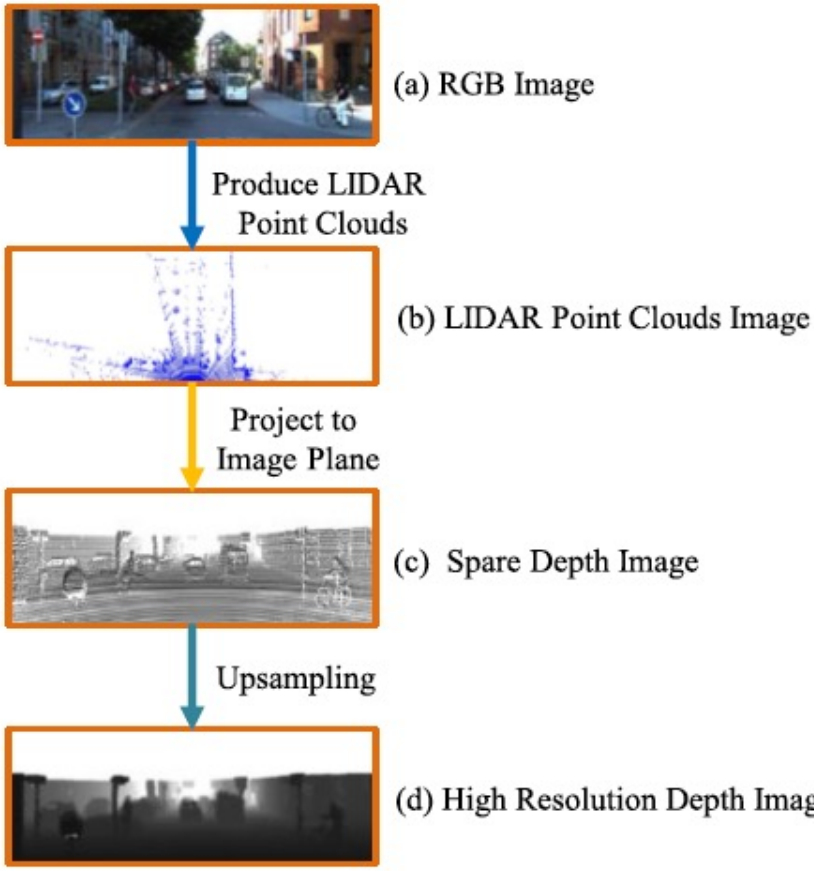
***Index Terms*—Autonomous vehicle, convolutional neural network (CNN), object classification, sensor fusion.**

EUREKA PROMETHEUS project [4], DARPA Grand Challenge [5], Google's autonomous vehicle [6], the annual "Intelligent Vehicle Future Challenge" organized by National Natural Science Foundation of China since 2009 [7]. Hundreds of teams from all over the world participate to compete and demonstrate technological achievements on autonomous vehicles, and to maximize car-following fuel economy and fulfill requirements of intervehicle safety. Especially, Hu *et al.* proposed an optimal look-ahead control method that is based on a model predictive fuel-optimal controller, which uses state trajectories of the leading vehicle from V2V/V2I communication [2]. Autonomous vehicles should be instantaneous, accurate, stable, and efficient in computations to produce safe and acceptable traveling trajectories in numerous urban to suburb scenarios and from high-density traffic flow to high-speed

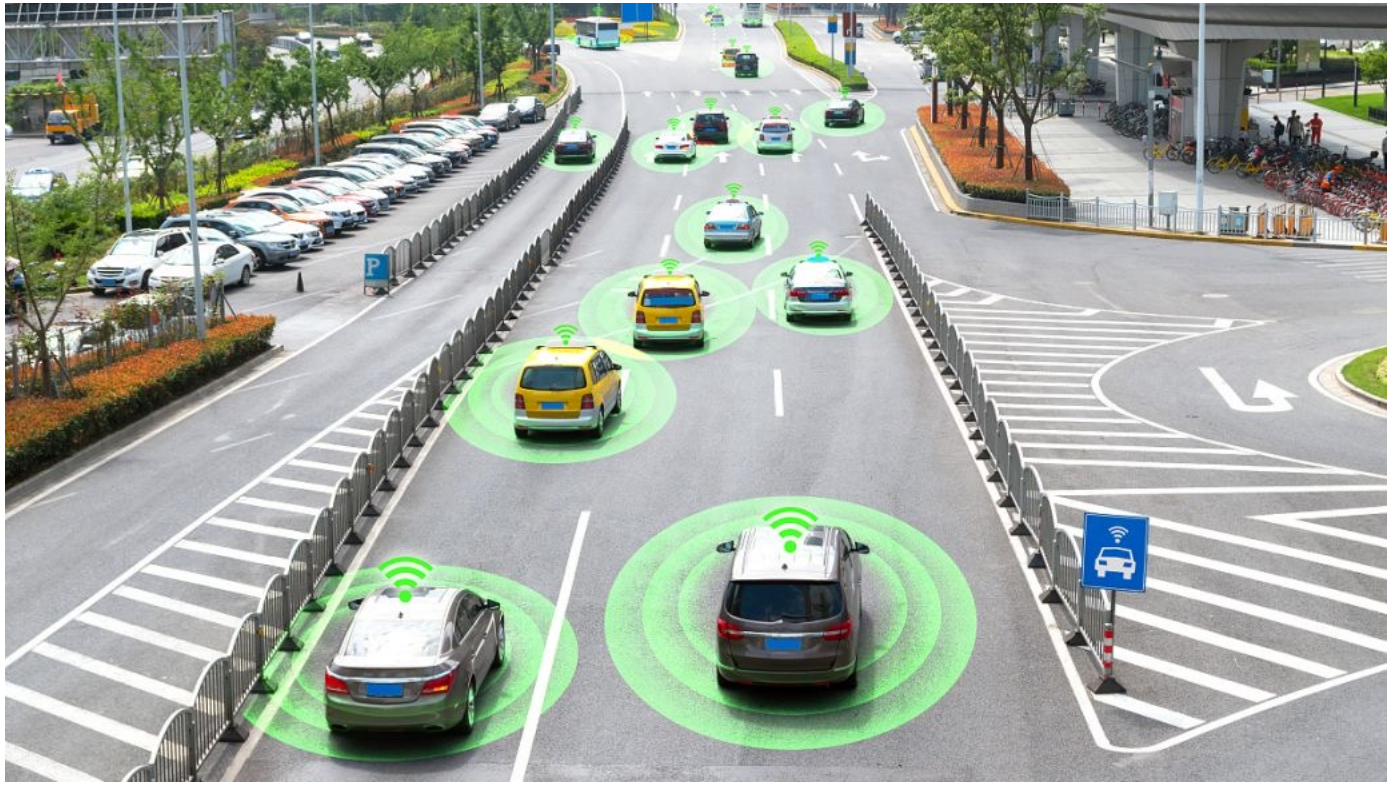
State of the art

Fig. 1. Mengshi autonomous vehicle.

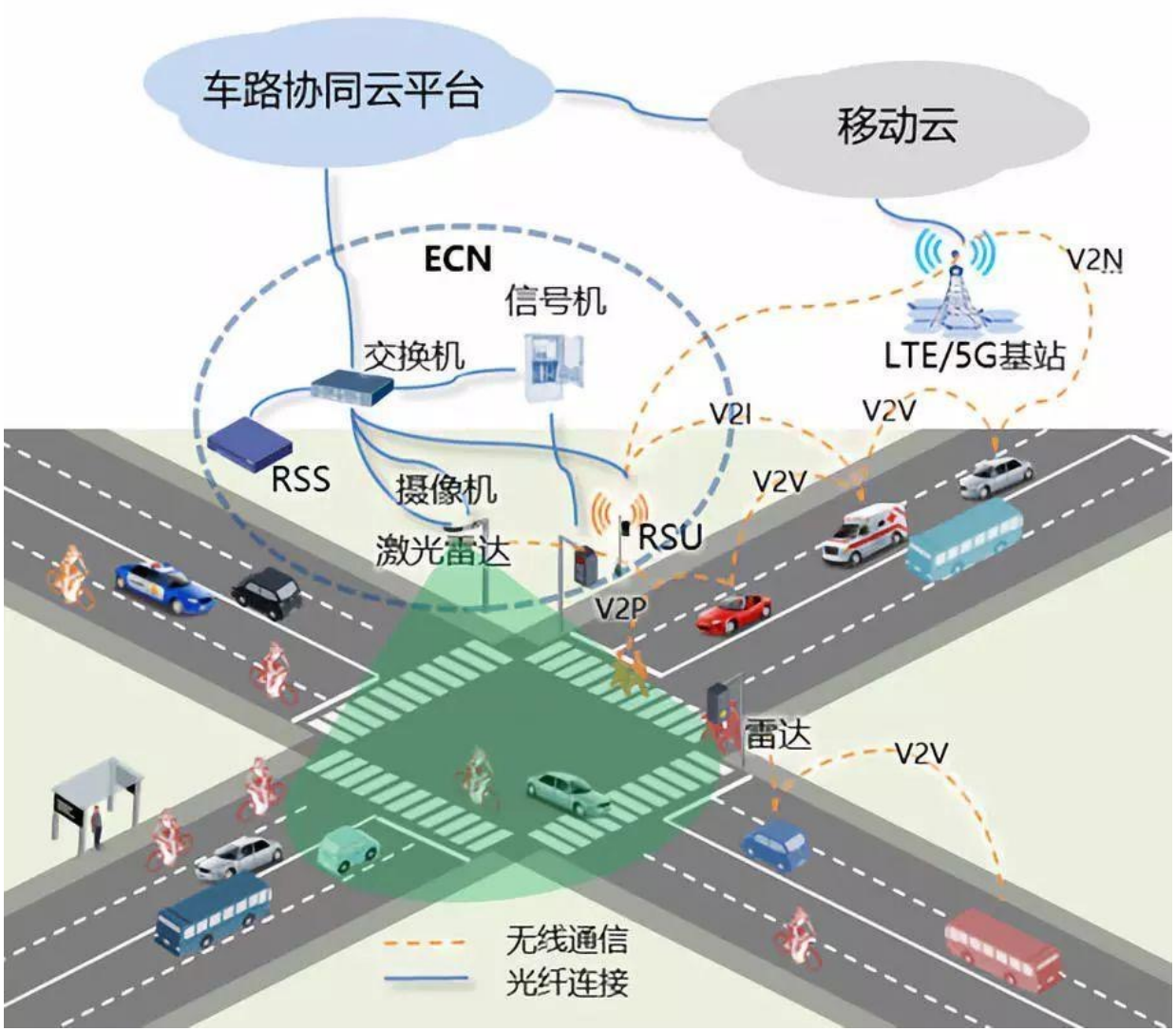
Fig. 2. Sensor deployment of Mengshi.



Vehicle-to-vehicle connectivity



Vehicle-to-infrastructure connectivity



Summary

- Single-vehicle on an empty road
  - 1 dimensional
  - 2 dimensional
- Multi-vehicle planning
  - Policy-free approach
  - Policy-based approach
- Perception
  - Onboard perception
  - Vehicle-to-vehicle connectivity
  - Vehicle-to-infrastructure connectivity

Next time

- Guest lecture by Prof. Chen (5/21)
- Vehicle platooning (5/24)
  - Technological basis
    - Autonomous driving
    - Vehicle-to-vehicle coordination
  - Formulation
    - Modeling
    - Decision making
  - Cooperative adaptive cruise control
    - Objective
    - Design