Intelligent Intersection Coordination and Trajectory Optimization for Autonomous Vehicles

Yixiao Zhang, Gang Chen and Tingting Zhang

School of Electronics and Information Engineering,
Harbin Institute of Technology, Shenzhen

IEEE ICAS 2021

OUTLINE:

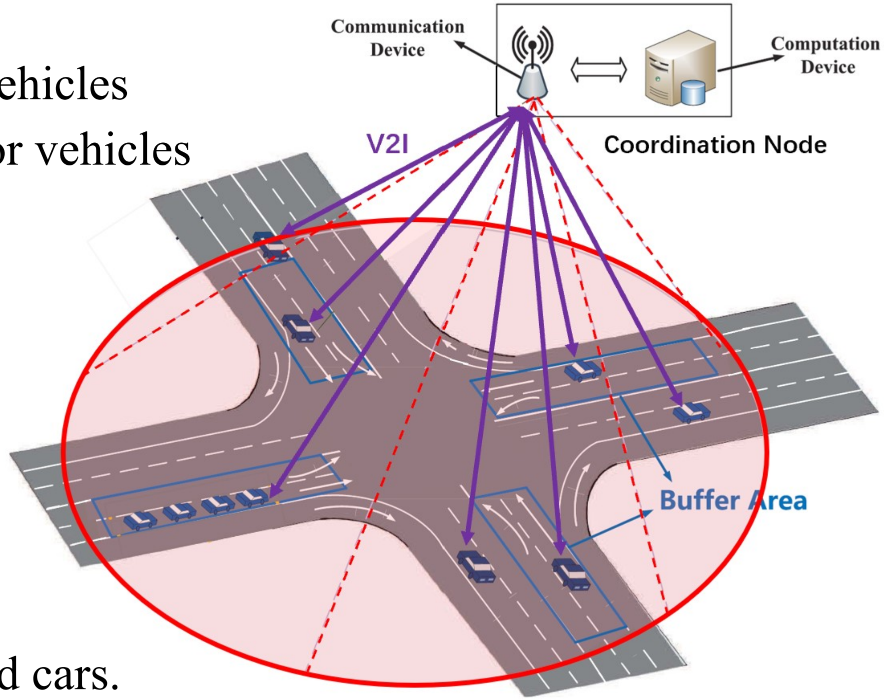
- Introduction
- Model
- High-level Planner (infrastructure-based)
- Low-level Planner (vehicle-based)
- Conclusion

❖ Background

- more and more vehicles are expected to enter the road infrastructure.
- **Intersections**: 40% of traffic accidents, including 20% fatal.
- **Intelligent Intersection Management Systems**:
  - Safety and Efficiency: avoid accidents as well as jams

❖ Infrastructure-based Strategies [1,2]:

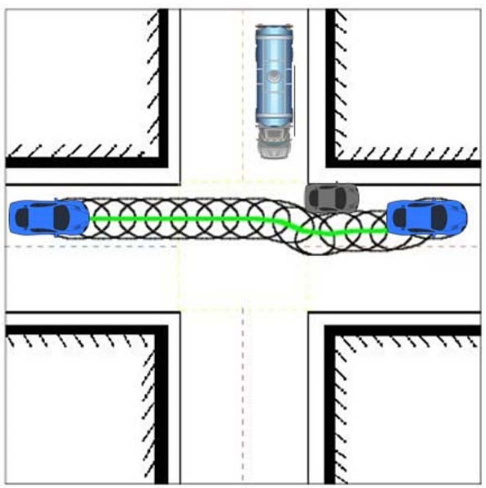
- A coordination node (at RSU):
  - receive the requests sending from vehicles
  - allocate space and time resources for vehicles
- **Merits:**
  - safety among intelligent vehicles
  - fuel efficiency
  - traffic throughput
- **Demerits:**
  lack of flexibility when
  - dealing with unexpected obstacles, such as pedestrians, bicycles, packed cars.
  - communication and perception error



[1] L. Chen and C. Englund, "Cooperative intersection management: A survey," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 570–586, 2016.
[2] E. Namazi, J. Li, and C. Lu, "Intelligent intersection management systems considering autonomous vehicles: A systematic literature review," IEEE Access, vol. 7, pp. 91 946–91 965, 2019.

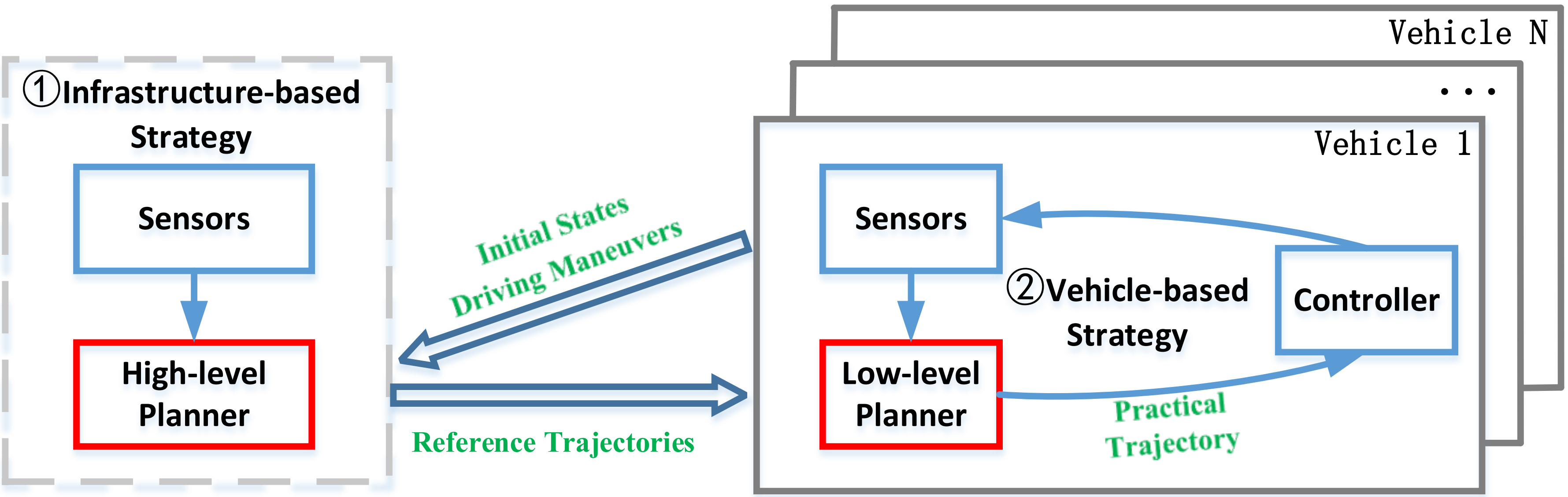
❖ Vehicle-based Strategies [3,4]:

- **Vehicles use their own sensors (cameras, radars and LIDARs) for environment awareness, and then plan trajectories accordingly.**
- **Advantages:** detect possible obstacles, trajectory smoothness
- **Disadvantages:** Deadlocks among vehicles, lacking in traffic throughput

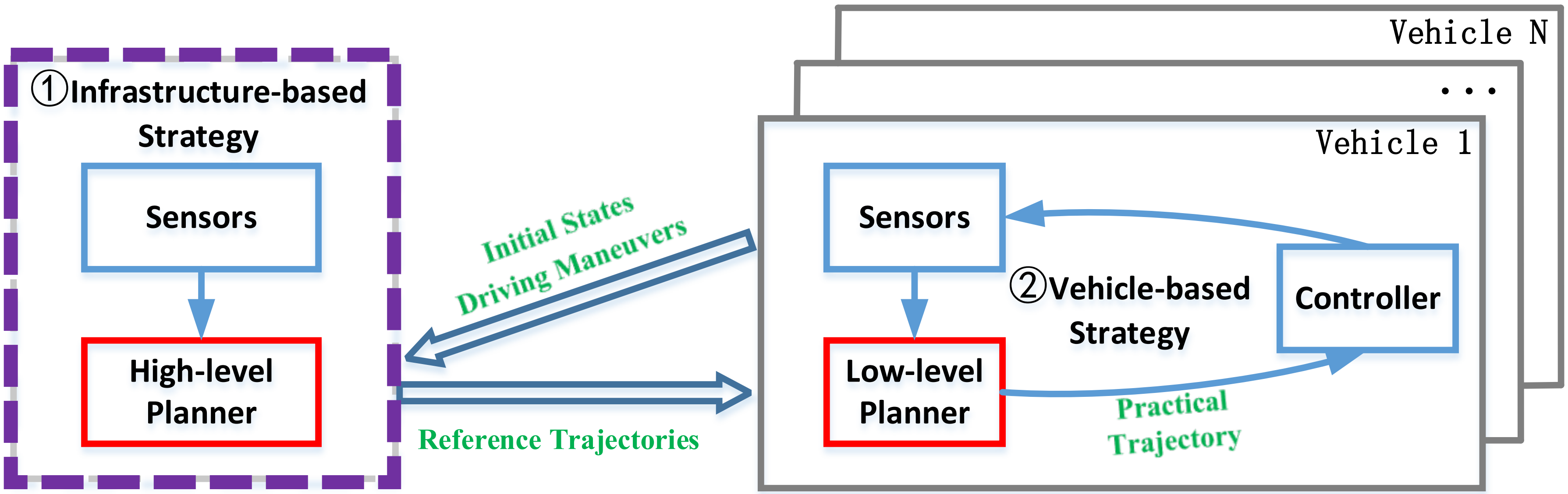


[3] D. González, J. Pérez, V. Milanés and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1135-1145, April 2016, doi: 10.1109/TITS.2015.2498841.
[4] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 5, pp. 1826–1848, 2020.

❖ Integrated Coordination Framework

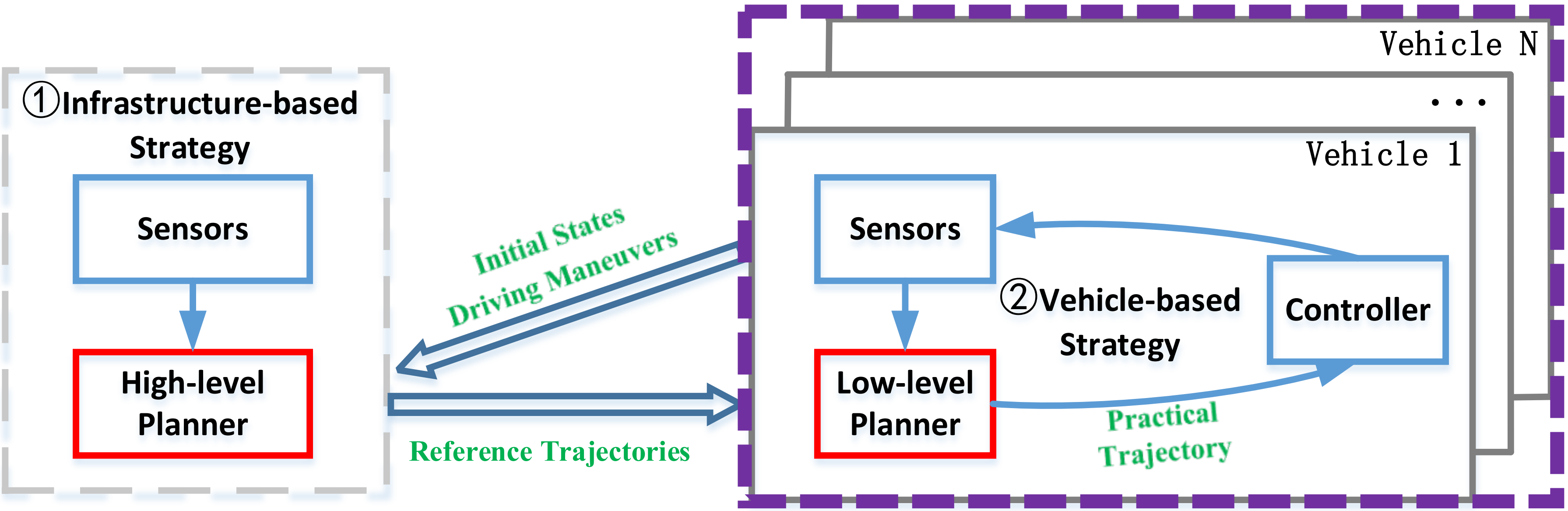


❖ Integrated Coordination Framework



- **High-level Planner (on coordination node):**
  - receive *driving maneuvers* and *initial states* from vehicles via V2I communication.
  - generate reference trajectories in Cartesian Coordinates.

❖ Integrated Coordination Framework

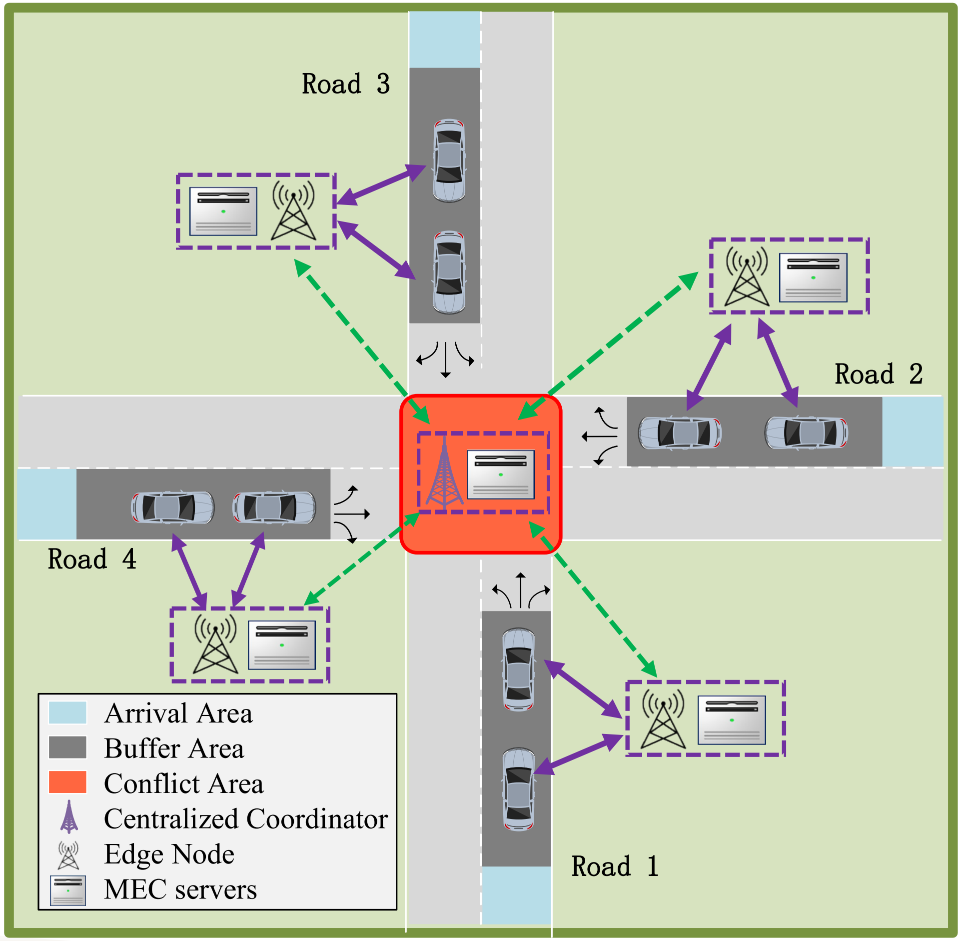


- **High-level Planner (on coordination node):**
  - receive *driving maneuvers* and *initial states* from vehicles via V2I communication.
  - generate reference trajectories in Cartesian Coordinates.
- **Low-level Planner (on each vehicle):**
  - replan its practical trajectory based on the reference trajectory in *Frenét Frame*, using on-board observations.

OUTLINE:

- Introduction
- Model
- High-level Planner (infrastructure-based)
- Low-level Planner (vehicle-based)
- Conclusion

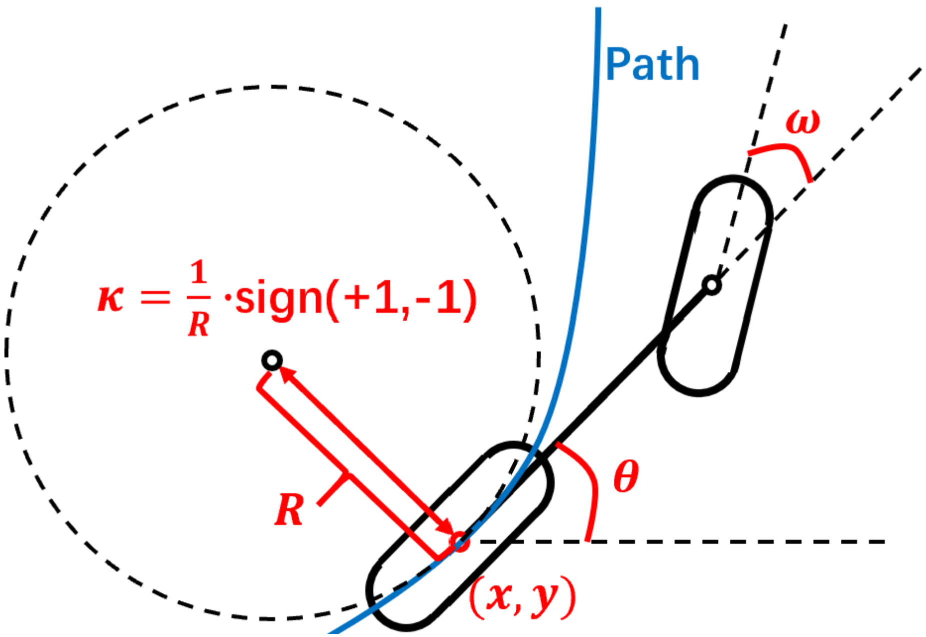
❖ Intersection Model



- **Areas**
  - --- Arrival Area (AA)
  - --- Buffer Area (BA)
  - --- Conflict Area (CA)
- **Computing Units**
  - --- Computing Units on Vehicles
  - --- Centralized Coordinator
  - --- Edge Nodes (Offloading)

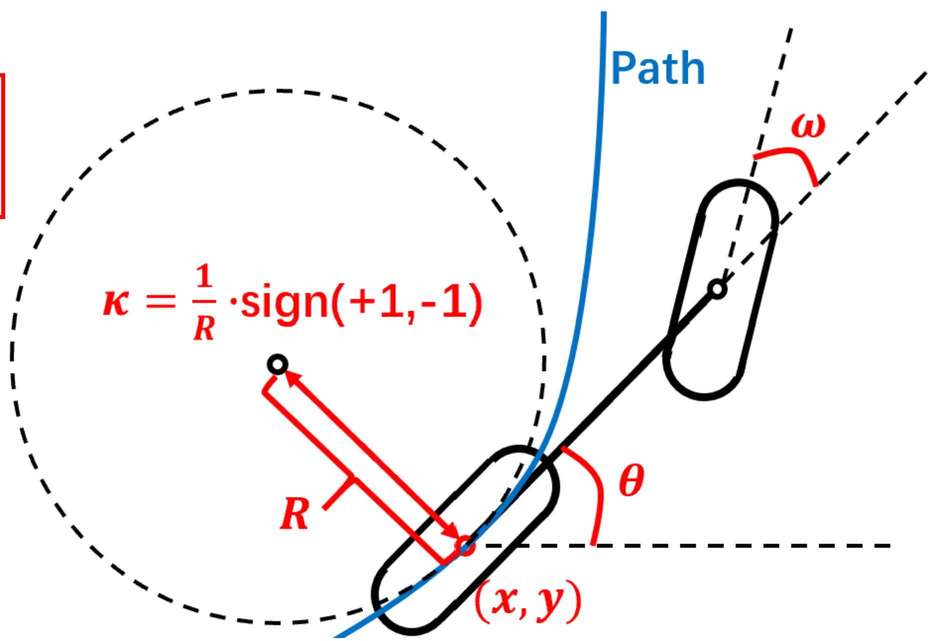
❖ Kinematic Bicycle Model

- **State parameters:** $[x, y, \theta, v]$
- **Inputs:** $[\kappa, a]$



❖ Kinematic Bicycle Model

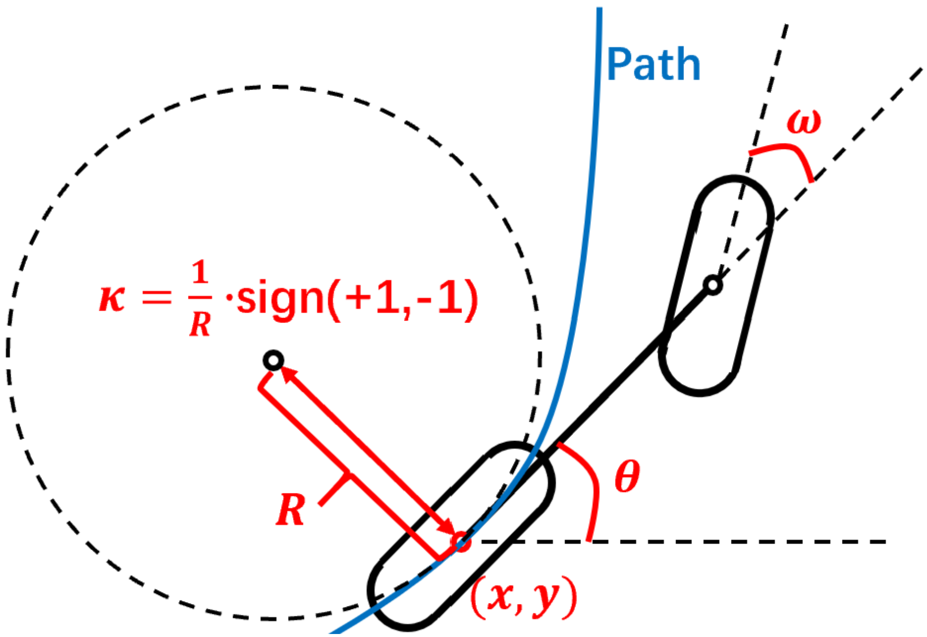
- **State parameters:** $[x, y, \theta, v]$
- **Inputs:** $[\boxed{\kappa}, a]$ $\boxed{\kappa = \dfrac{\tan(\omega)}{L}}$



❖ Kinematic Bicycle Model

- **State parameters:** $[x, y, \theta, v]$
- **Inputs:** $[\kappa, a]$
- **Kinetic Equation:**

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = v \cdot \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \kappa \\ 0 \end{bmatrix} + a \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



❖ Trajectory Functions

- **Cartesian Coordinates:**

$$\begin{cases} \text{Path:} & \begin{cases} x = f(s) \\ y = g(s) \end{cases} \\ \text{Speed Profile:} & s = u(t) \end{cases}$$

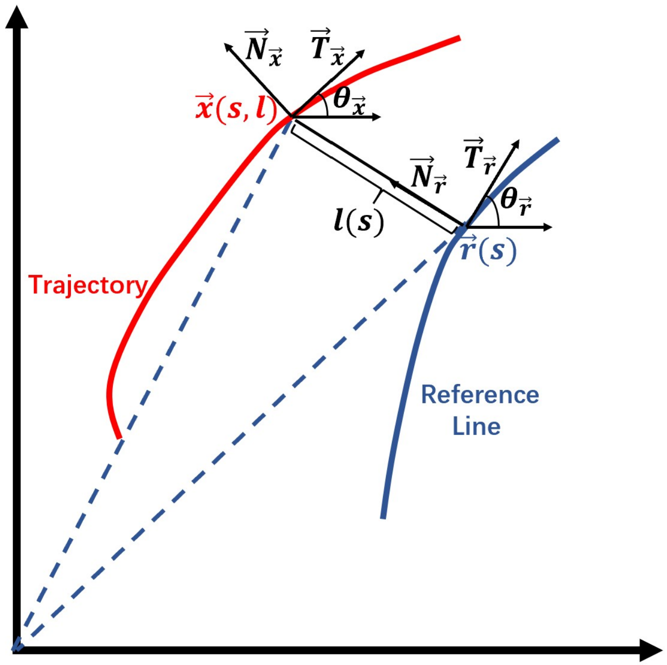
Trajectory Functions

- **Cartesian Coordinates:**

$$\begin{cases} \text{Path:} & \begin{cases} x = f(s) \\ y = g(s) \end{cases} \\ \text{Speed Profile:} & s = u(t) \end{cases}$$

- **Frenét Frame (SLT) [5]:**

$$\begin{cases} \text{Path:} & l = p(s) \\ \text{Speed Profile:} & s = q(t) \end{cases}$$



[5] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a fren´et frame," in 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 987–993.

Trajectory Functions

- **Cartesian Coordinates:**

$$\begin{cases} \text{Path:} & \begin{cases} x = f(s) \\ y = g(s) \end{cases} \\ \text{Speed Profile:} & s = u(t) \end{cases}$$

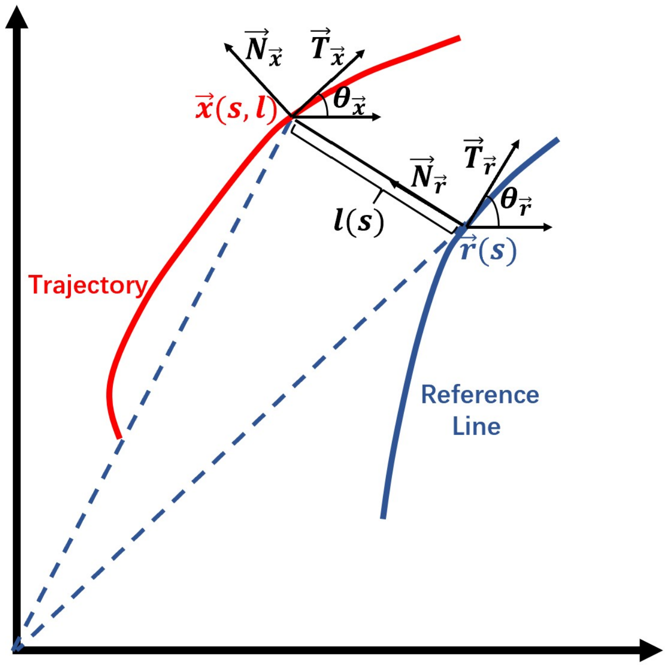
- **Frenét Frame (SLT) [5]:**

$$\begin{cases} \text{Path:} & l = p(s) \\ \text{Speed Profile:} & s = q(t) \end{cases}$$

- **Control:**

$$[f, g, \arctan(\frac{g'}{f'}), \frac{|f'g'' - f''g'|}{(f'^2 + g'^2)^{3/2}}, u', u'']^{\mathrm{T}} = \boxed{[x, y, \theta, \kappa, v, a]^{\mathrm{T}}}$$

$$[p, q, p', p'', q', q''] \mapsto \boxed{[x, y, \theta, \kappa, v, a]}$$



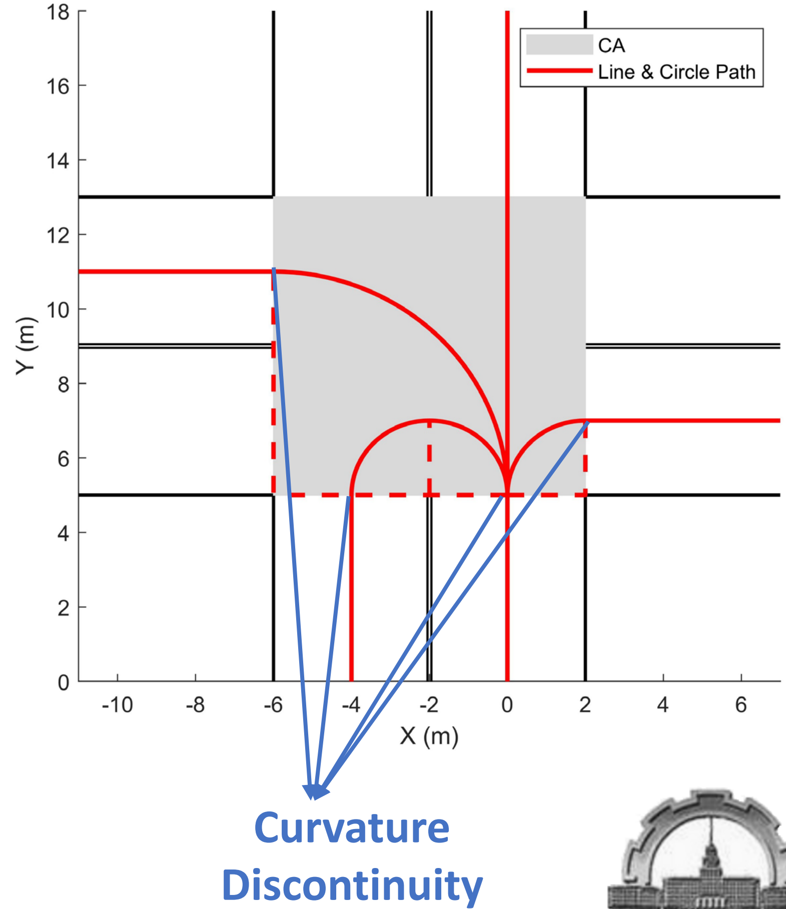
[5] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 987–993.

OUTLINE:

- Introduction
- Model
- High-level Planner (infrastructure-based)
- Low-level Planner (vehicle-based)
- Conclusion

❖ Reference Path Generator

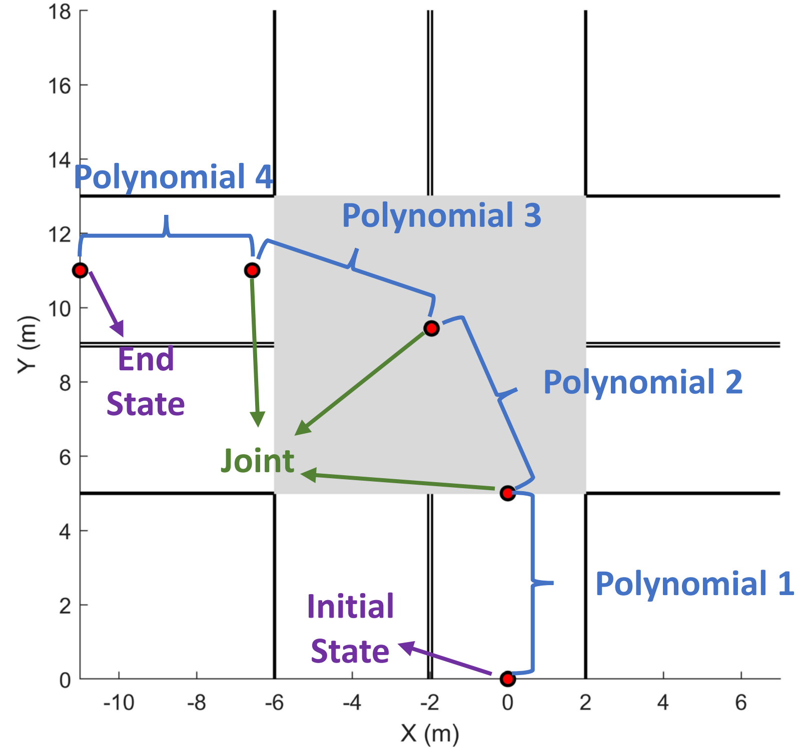
- Find the reference path functions $x = f(s), y = g(s)$ for each vehicle
- Sampling on Line & Circle Path



❖ Reference Path Generator

- **Find the reference path functions x $= f(s), y = g(s)$ for each vehicle**
- **Sampling on Line & Circle Path**

  Knots→ parameterization (spline)



❖ Reference Path Generator

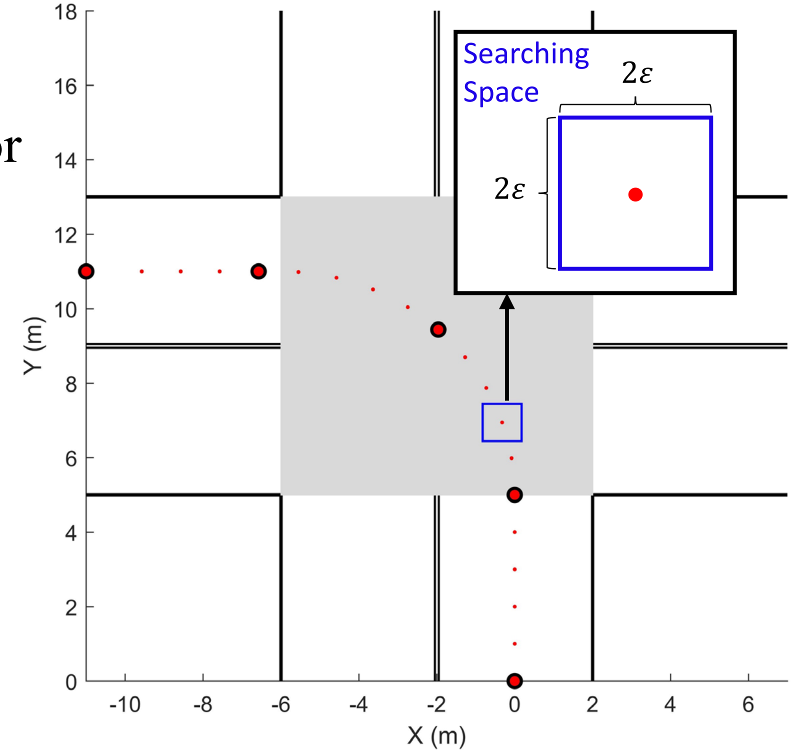
- **Find the reference path functions x = $f(s)$, $y = g(s)$ for each vehicle**
- **Sampling on Line & Circle Path**

Knots→ parameterization (spline)

Anchor Points→ keep the vehicle behavior

$$|f(s_{a,j}) - x_{a,j}| \le \varepsilon$$

$$|g(s_{a,j}) - y_{a,j}| \le \varepsilon$$



Reference Path Generator

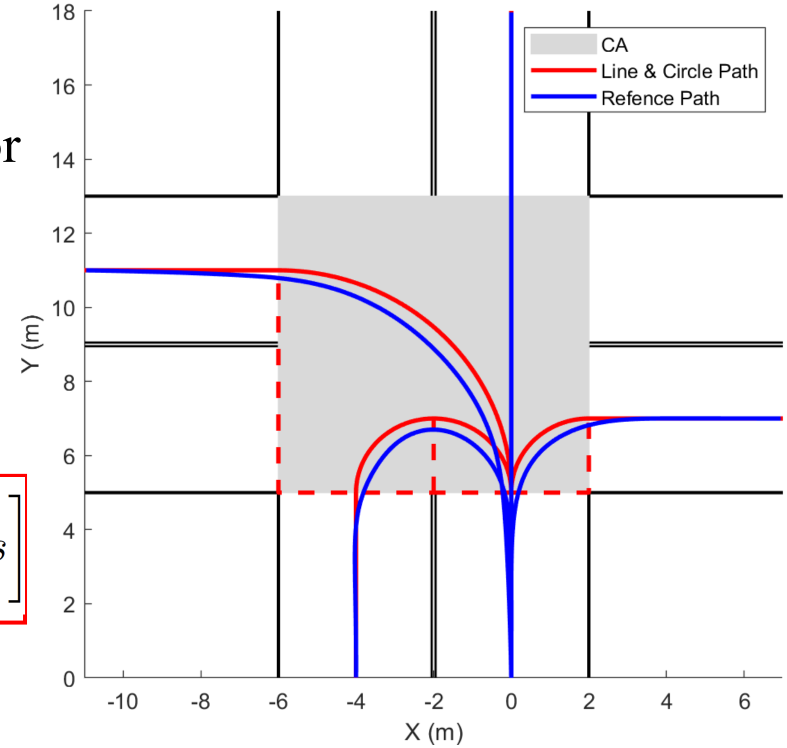
- **Find the reference path functions x = $f(s)$, $y = g(s)$ for each vehicle**
- **Sampling on Line & Circle Path**

  Knots→ parameterization (spline)

  Anchor Points→ keep the vehicle behavior

$$\left|f(s_{\mathrm{a},j}) - x_{\mathrm{a},j}\right| \leq \varepsilon$$

$$\left|g(s_{\mathrm{a},j}) - y_{\mathrm{a},j}\right| \leq \varepsilon$$



- **Smooth Optimization**

$$\text{Min.}\ \boxed{\sum_{z=2}^{3} w_{\mathrm{s},z}^{\mathrm{P}} \left[\int \left(f^{(z)}\right)^{2}(s)\mathrm{d}s + \int \left(g^{(z)}\right)^{2}(s)\mathrm{d}s\right]}$$

Curvature and Derivative of curvature

Reference Path Generator

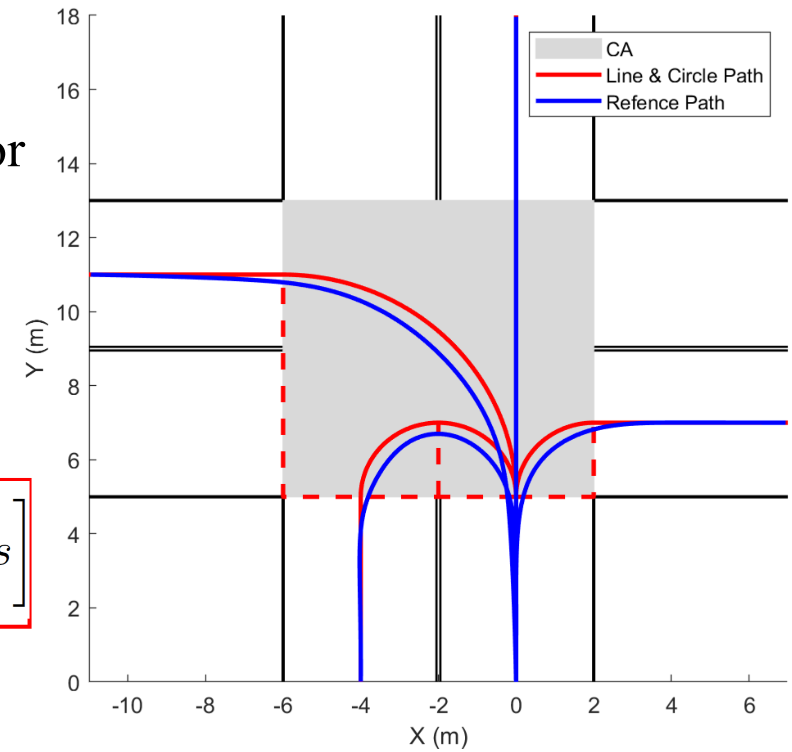
- **Find the reference path functions $x = f(s), y = g(s)$ for each vehicle**
- **Sampling on Line & Circle Path**

  Knots→ parameterization (spline)

  Anchor Points→ keep the vehicle behavior

$$|f(s_{a,j}) - x_{a,j}| \le \varepsilon$$

$$|g(s_{a,j}) - y_{a,j}| \le \varepsilon$$



- **Smooth Optimization**

$$\text{Min.} \boxed{\sum_{z=2}^{3} w_{s,z}^{P} \left[\int \left(f^{(z)}\right)^2 (s)\mathrm{d}s + \int \left(g^{(z)}\right)^2 (s)\mathrm{d}s \right]}$$

Curvature and Derivative of curvature

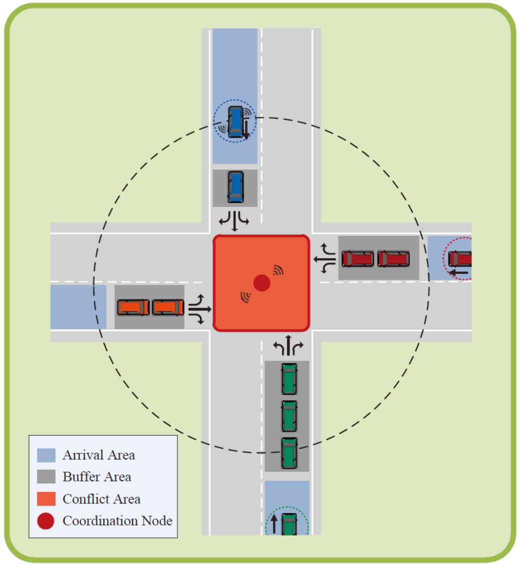
- **Solver:**

  Quadratic Programming (QP)

  → finding the optimal spline coefficients → reference path

❖ Reference Speed Profile Generator

- **Find reference speed functions $u(t)$ for vehicles**



❖ Reference Speed Profile Generator

- **Find reference speed functions $u(t)$ for vehicles**
- **The collision-set coordination**

  →only one vehicle can occupy CA simultaneously

$$\boxed{\boxed{\left[t_1^{\mathrm{In}} \quad t_1^{\mathrm{Out}}\right]} \cap \boxed{\left[t_2^{\mathrm{In}} \quad t_2^{\mathrm{Out}}\right]} = \varnothing}$$

Occupation time of Vehicle 1 — Occupation time of Vehicle 2

- Find reference speed functions $u(t)$ for vehicles
- The collision-set coordination

  →only one vehicle can occupy CA simultaneously

Reference Speed Profile Generator

- **Find reference speed functions $u(t)$ for vehicles**
- **The collision-set coordination**

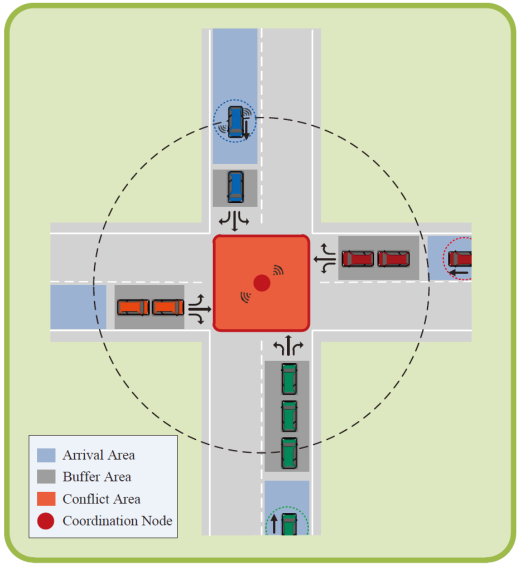
  →only one vehicle can occupy CA simultaneously

$$\left[t_1^{\mathrm{In}} \quad t_1^{\mathrm{Out}}\right] \cap \left[t_2^{\mathrm{In}} \quad t_2^{\mathrm{Out}}\right] = \varnothing$$

Two integer $\alpha, \beta \in \{0,1\}$

$$s_i^{\mathrm{Out}} - \boxed{\alpha_i} \times M \le u_i(t) \le s_i^{\mathrm{In}} + \boxed{\beta_i} \times M$$

$$\boxed{\alpha_{i_1} + \alpha_{i_2}} + \boxed{\beta_{i_1} + \beta_{i_2}} \le 3$$



❖ Reference Speed Profile Generator

- **Find reference speed functions $u(t)$ for vehicles**
- **The collision-set coordination**

  →only one vehicle can occupy CA simultaneously

$$\left[t_1^{\mathrm{In}} \quad t_1^{\mathrm{Out}}\right] \cap \left[t_2^{\mathrm{In}} \quad t_2^{\mathrm{Out}}\right] = \varnothing$$

Two integer $\alpha, \beta \in \{0,1\}$

$$s_i^{\mathrm{Out}} - \boxed{\alpha_i} \times M \le u_i(t) \le s_i^{\mathrm{In}} + \boxed{\beta_i} \times M$$

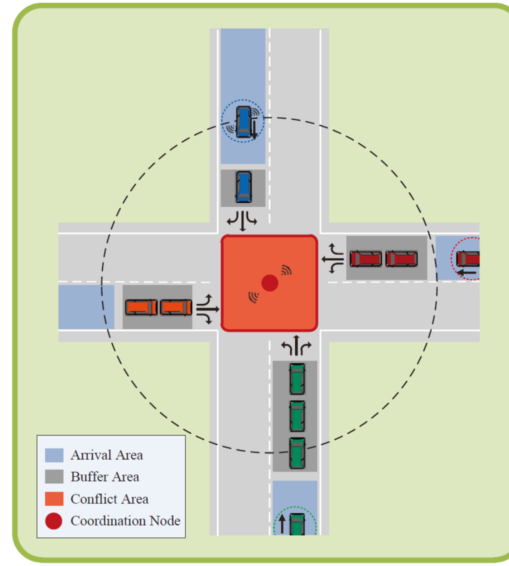
$$\boxed{\alpha_{i_1} + \alpha_{i_2}} + \boxed{\beta_{i_1} + \beta_{i_2}} \le 3$$

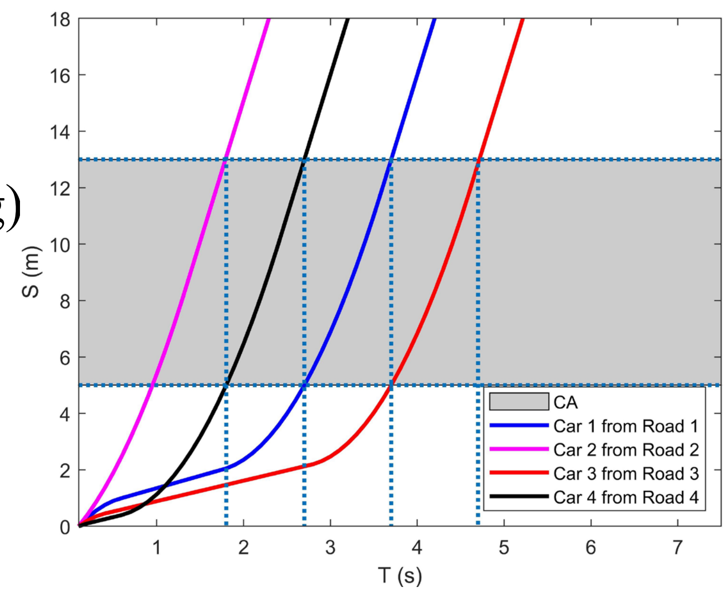
- **Optimization** (mixed integer programming)

Throughput $$C_{\mathrm{p,ref}}^{\mathrm{S}}(u) = w_{\mathrm{p}}^{\mathrm{S}} \int \left(u'(t) - \boxed{v_{\mathrm{eff}}}\right)^2 \mathrm{d}t$$

Smoothness $$C_{\mathrm{s,ref}}^{\mathrm{S}}(u) = \sum_{z=2}^{3} \left[w_{\mathrm{s},z}^{\mathrm{S}} \int \left(u^{(z)}\right)^2 (t)\mathrm{d}t \right]$$

Acceleration, jerk



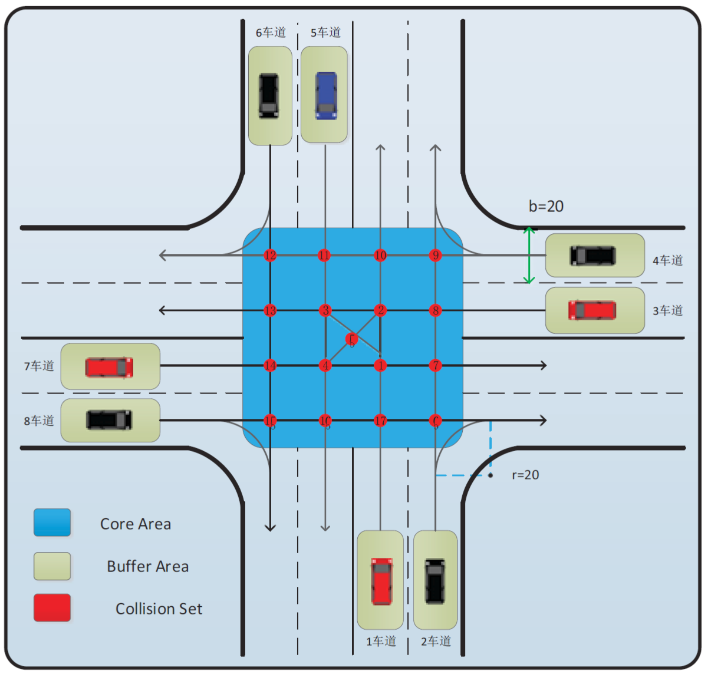


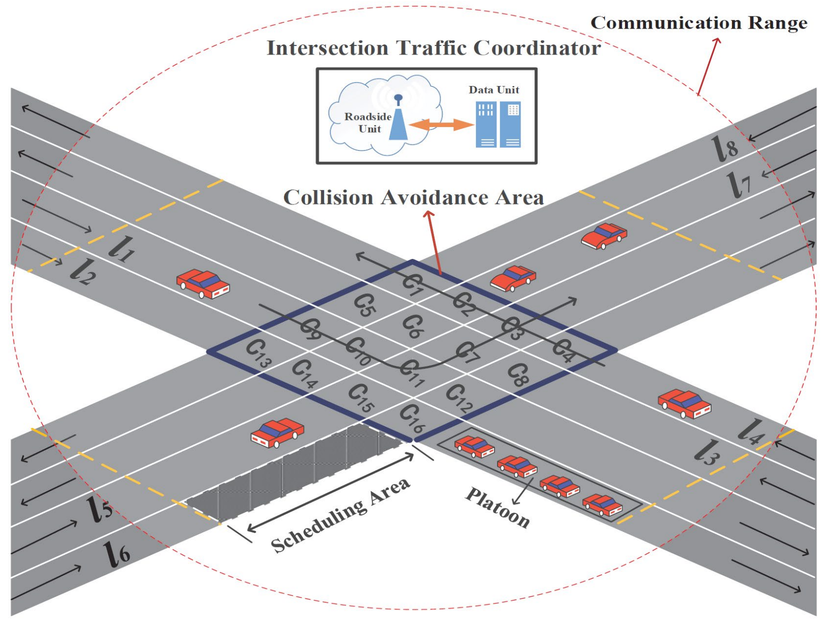
Reference Speed Profile Generator

- **Single-collision-set strategy**

❖ Reference Speed Profile Generator

- **Multi-collision-set strategy [6,7]**
  - based on the road structure
  - based on the number of roads and lanes





[6] C. Liu, Y. Mo, B. Gao and T. Zhang, "Low Complexity Coordination Strategies at Multi-Lane Intersections," 2019 IEEE Globecom Workshops (GC Wkshps), 2019, pp. 1-6.
[7] B. Qian, H. Zhou, F. Lyu, J. Li, T. Ma and F. Hou, "Toward Collision-Free and Efficient Coordination for Automated Vehicles at Unsignalized Intersection," in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10408-10420, Dec. 2019.

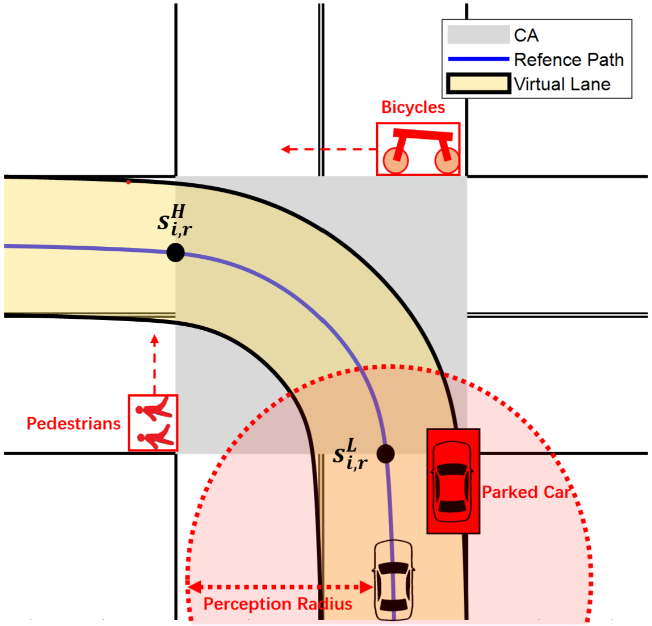
OUTLINE:

- Introduction
- Model
- High-level Planner (infrastructure-based)
- Low-level Planner (vehicle-based)
- Conclusion

❖ Low-level Planner

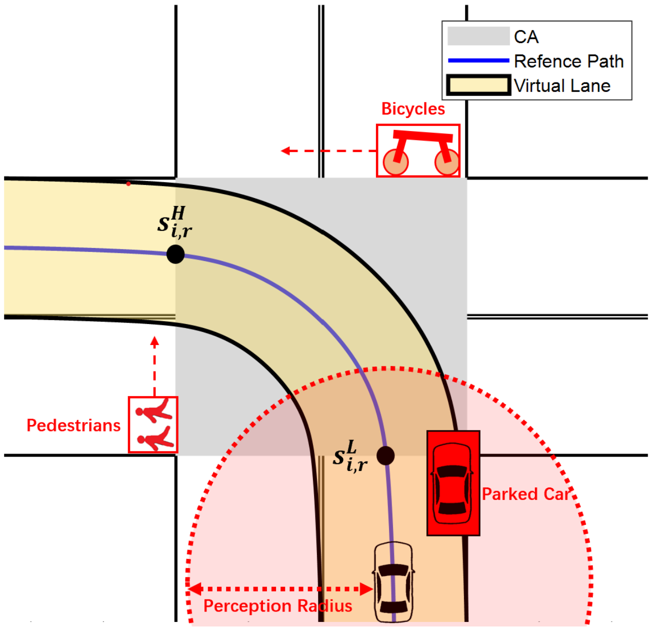
- **Consider various obstacles**

  -- pedestrians, bicycles, parked cars.



❖ Low-level Planner

- **Consider various obstacles**
  -- pedestrians, bicycles, parked cars.
- **Limited perception field**
  -- replanning iteratively
  (deal with the oncoming obstacles)



❖ Low-level Planner

- **Consider various obstacles**

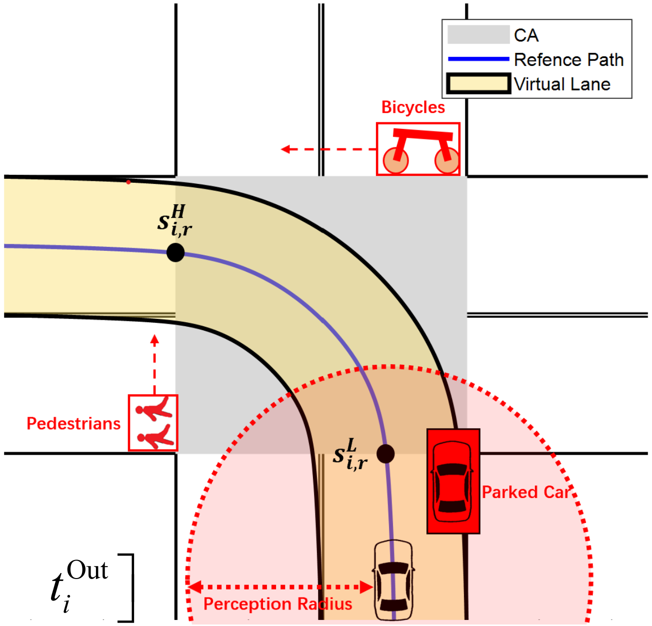
  -- pedestrians, bicycles, parked cars.
- **Limited perception field**

  -- replanning iteratively

  (deal with the oncoming obstacles)
- **Follow instructions from high-level planner**

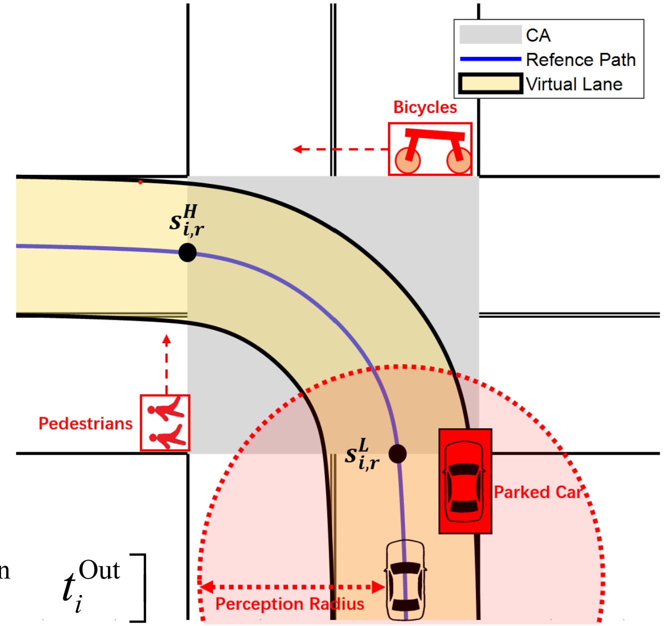
  -- path → within the virtual lane

  -- speed profile → within the time limitation $\left[t_i^{\text{In}} \quad t_i^{\text{Out}}\right]$



❖ Low-level Planner

- **Consider various obstacles**
  -- pedestrians, bicycles, parked cars.
- **Limited perception field**
  -- replanning iteratively
  (deal with the oncoming obstacles)
- **Follow instructions from high-level planner**
  -- path → within the virtual lane
  -- speed profile → within the time limitation $\left[t_i^{\mathrm{In}} \quad t_i^{\mathrm{Out}} \right]$

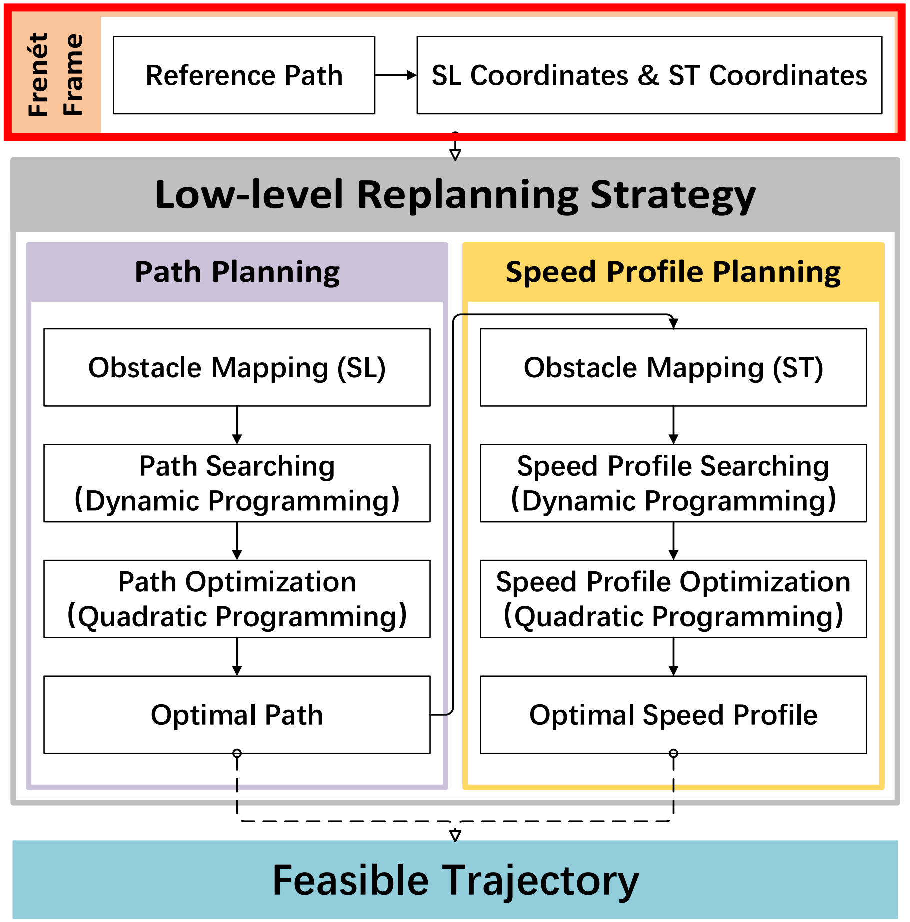


Impact	Path planning	Speed Profile Planning
smoothness	$C_{\mathrm{s}}^{\mathrm{P}} = \sum_{z=2}^{3} \left[w_z^{\mathrm{P}} \int \left(p^{(z)}\right)^2 (s) \mathrm{d}s \right]$	$C_{\mathrm{s}}^{\mathrm{S}} = \sum_{z=2}^{3} \left[w_z^{\mathrm{S}} \int \left(q^{(z)}\right)^2 (t) \mathrm{d}t \right]$
vehicle behavior	$C_{\mathrm{b}}^{\mathrm{P}} = w_4^{\mathrm{P}} \int p^2(s) \mathrm{d}s$	$C_{\mathrm{b}}^{\mathrm{S}} = w_4^{\mathrm{S}} \int \left(q(t) - u_{i,r}(t)\right)^2 \mathrm{d}t$
collision avoidance	$C_{\mathrm{c}}^{\mathrm{P}} = \sum_{j=1}^{N_{\mathrm{o}}} \begin{cases} \mathrm{Inf} & d_j^{\mathrm{s}} < d_{j,\min}^{\mathrm{s}} \ \& \ d_j^{\mathrm{l}} < d_{j,\min}^{\mathrm{l}} \\ 0 & else \end{cases}$	$C_{\mathrm{c}}^{\mathrm{S}} = \sum_{j=1}^{N_{\mathrm{o}}} \begin{cases} \mathrm{Inf} & d_j^{\mathrm{s}} < d_{j,\min}^{\mathrm{s}} \\ 0 & else \end{cases}$

compare with the reference trajectory

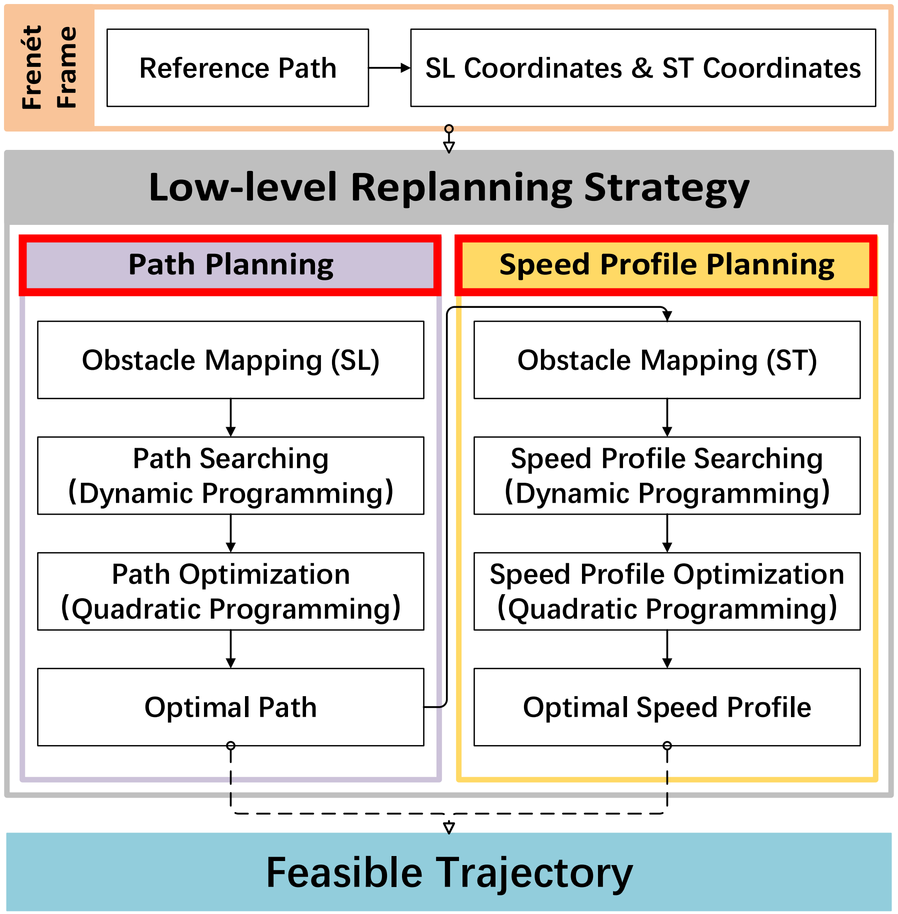
compare the distances and safe distances with obstacles

Low-level Planner [4][5]



- **Frenét Frame**
  - based on reference path

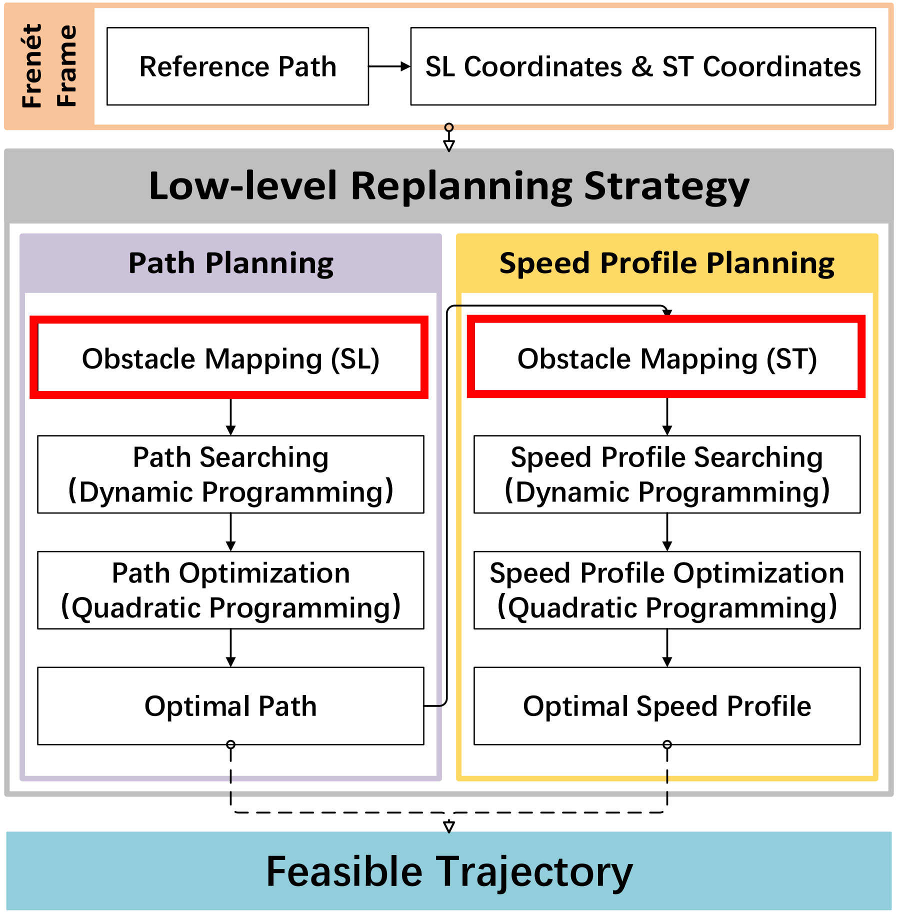
❖ Low-level Planner [4][5]



- **Frenét Frame**
  - based on reference path
- **Planning Strategy:**
  - Path-speed iterative algorithm [8]

[8] Wenda Xu, Junqing Wei, J. M. Dolan, Huijing Zhao, and Hongbin Zha, “A real-time motion planner with trajectory optimization for autonomous vehicles,” in 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 2061–2067.

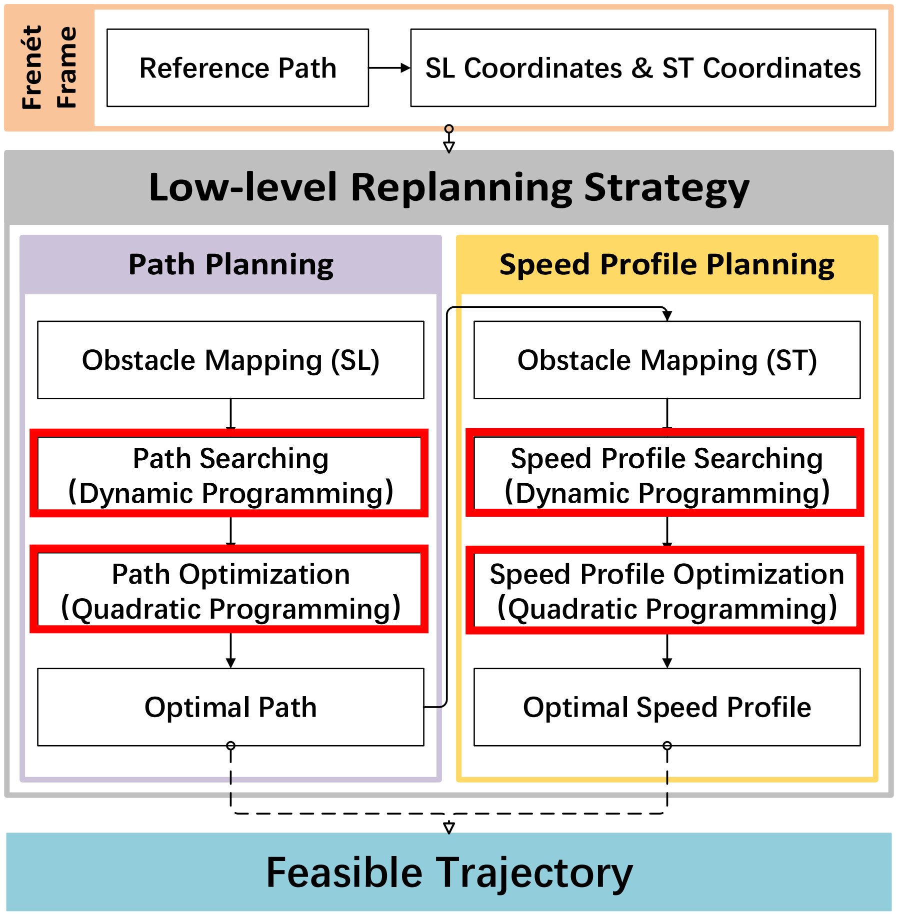
❖ Low-level Planner [4][5]



- **Frenét Frame**
  - based on reference path
- **Planning Strategy:**
  - Path-speed iterative algorithm [8]
- **Obstacle Mapping:**
  - Expectation Maximum (EM) [9]

[8] Wenda Xu, Junqing Wei, J. M. Dolan, Huijing Zhao, and Hongbin Zha, “A real-time motion planner with trajectory optimization for autonomous vehicles,” in 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 2061–2067.
[9] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, “Baidu apollo em motion planner,” arXiv:1807.08048, 2018.

❖ Low-level Planner [4][5]

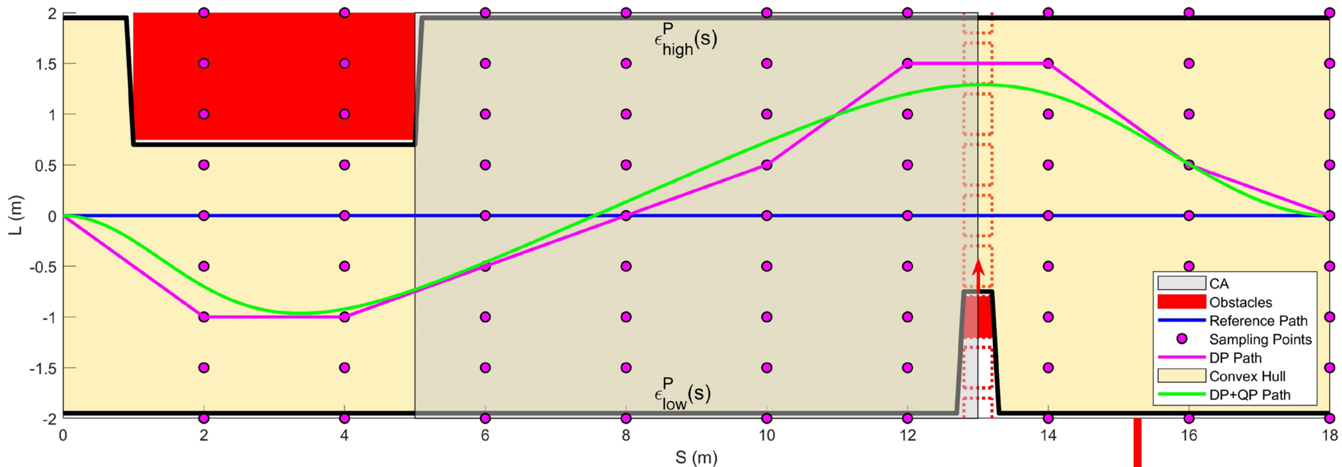


- **Frenét Frame**
  - based on reference path
- **Planning Strategy:**
  - Path-speed iterative algorithm [8]
- **Obstacle Mapping:**
  - Expectation Maximum (EM) [9]
- **Solver:**
  - Dynamic Programming (DP) + Quadratic Programming (QP)

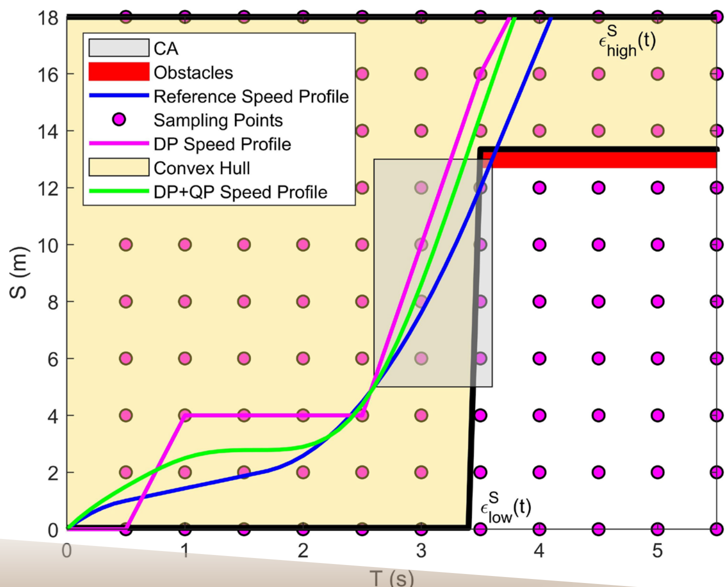
[8] Wenda Xu, Junqing Wei, J. M. Dolan, Huijing Zhao, and Hongbin Zha, “A real-time motion planner with trajectory optimization for autonomous vehicles,” in 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 2061–2067.
[9] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, “Baidu apollo em motion planner,” arXiv:1807.08048, 2018.

❖ Low-level Planner

Path Planning



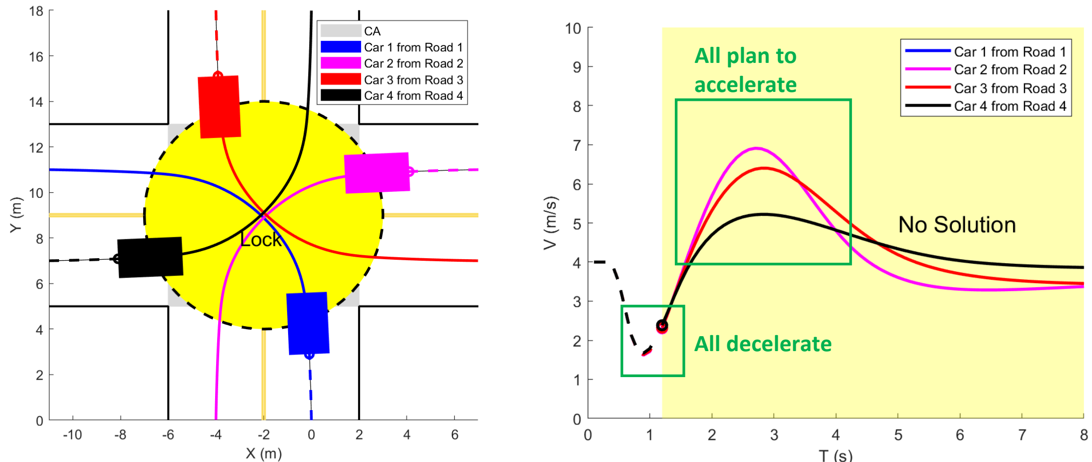
Speed Profile Planning



Within Feasible Regions

❖ Numerical Results for integrated framework

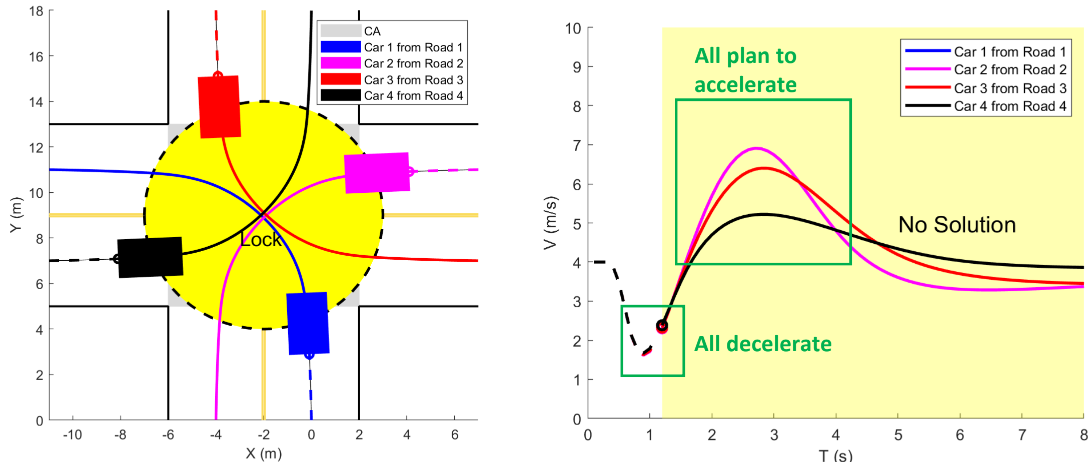
- **Avoid deadlocks**



Only use low-level planner

❖ Numerical Results for integrated framework

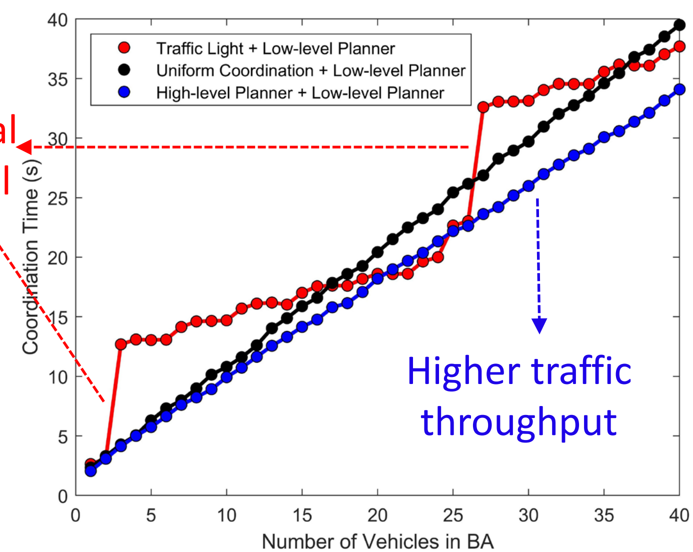
- **Avoid deadlocks**



Only use low-level planner

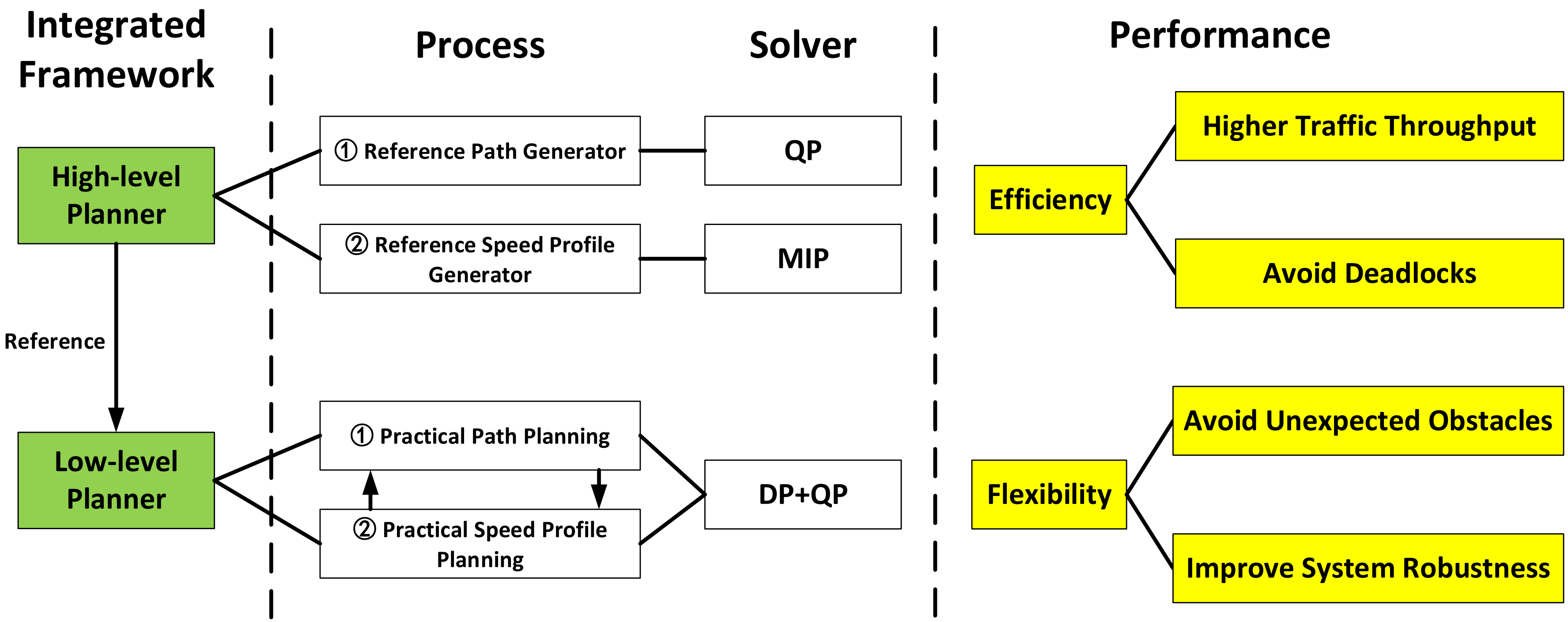
- **Traffic Throughput**
  - traffic lights + low-level planner
  - uniform coordination + low-level planner
  - high-level planner + low-level planner (proposed)

Change interval of traffic signal



OUTLINE:

Conclusion


Integrated Framework
Process
Solver
Performance
High-level Planner
① Reference Path Generator
QP
② Reference Speed Profile Generator
MIP
Reference
Low-level Planner
① Practical Path Planning
② Practical Speed Profile Planning
DP+QP
Efficiency
Higher Traffic Throughput
Avoid Deadlocks
Flexibility
Avoid Unexpected Obstacles
Improve System Robustness

Thanks for Listening