Convex Optimization Theory and Applications

Topic 10 - Time Series Problems

Li Li

Department of Automation
Tsinghua University

Fall, 2009-2021.

10.0. Outline

10.1. Kalman Filtering

10.2. l_1 Filtering and Total Variation

10.3. 轨迹规划

10.1. Kalman Filtering

Rudolf E. Kalman

May 19, 1930, Budapest - July 2, 2016, Gainesville

10.1. Kalman Filtering

The Kalman filtering method gives an efficient recursive estimateor for the state of a linear discrete-time dynamical system with noise. It was first introduced by Kalman in 1960 [1]. Usually, the system is represented as [1]-[8]

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state to be estiamted, $\boldsymbol{u} \in \mathbb{R}^m$ denotes the control input known in advance. $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ is the control matrix.

We can obtain a measure $\boldsymbol{z} \in \mathbb{R}^l$ that is

$$\boldsymbol{z}(k) = H\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{1.2}$$

where $H \in \mathbb{R}^{l\times n}$ is the observe matrix.

$\boldsymbol{w} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^l$ denote the multivariate normal distributed disturbances that satisfy

$$\boldsymbol{w}(k) \sim N(\mathbf{0}, Q), \;\; \boldsymbol{v}(k) \sim N(\mathbf{0}, R) \tag{1.3}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{l\times l}$ are the corresponding covariance matrices. Moreover, we assume that $\boldsymbol{x}(0)$, $\boldsymbol{w}$ and $\boldsymbol{v}$ are mutually independent.

Let $Z_k = (\boldsymbol{z}(1), \ldots, \boldsymbol{z}(k))$ denote the historical observations of $\boldsymbol{z}$ at time k, $X_k = (\boldsymbol{x}(1), \ldots, \boldsymbol{x}(k))$ denote the historical values of $\boldsymbol{x}$ at time k. Our goal is to obtain the posterior distribution of X_k given Z_k. Using Bayes' rule and the assumption of independency, we have

$$\Pr(X_k \mid Z_k) = \frac{\Pr(Z_k \mid X_k)\Pr(X_k)}{\Pr(Z_k)} \tag{1.4}$$

$$= \frac{\Pr(\boldsymbol{x}(0)) \prod_{t=1}^{k} \Pr(\boldsymbol{z}(t) \mid \boldsymbol{x}(t)) \prod_{t=1}^{k-1} \Pr(\boldsymbol{x}(t+1) \mid \boldsymbol{x}(t))}{\Pr(Z_k)} \tag{1.5}$$

$$\propto \Pr(\boldsymbol{x}(0)) \prod_{t=1}^{k} \Pr[\boldsymbol{z}(t) - H\boldsymbol{x}(t)] \prod_{t=1}^{k-1} \Pr[\boldsymbol{x}(t+1) - A\boldsymbol{x}(t) - B\boldsymbol{u}(t)] \tag{1.6}$$

$$= \Pr(\boldsymbol{x}(0)) \prod_{t=1}^{k} \Pr(\boldsymbol{v}(t)) \prod_{t=1}^{k-1} \Pr(\boldsymbol{w}(t)) \tag{1.7}$$

10.1. Kalman Filtering

We can view the Kalman filtering as the MAP estimator

$$\min_{\boldsymbol{x}(t)} \prod_{t=1}^{k} \exp^{-\frac{1}{2}[\boldsymbol{z}(t)-H\boldsymbol{x}(t)]^T Q^{-1}[\boldsymbol{z}(t)-H\boldsymbol{x}(t)]} + \prod_{t=1}^{k-1} \exp^{-\frac{1}{2}[\boldsymbol{x}(t+1)-A\boldsymbol{x}(t)-B\boldsymbol{u}(t)]^T R^{-1}[\boldsymbol{x}(t+1)-A\boldsymbol{x}(t)-B\boldsymbol{u}(t)]} \tag{10.8}$$

Take the natural logarithm of this likelihood function, we can finally transform the Kalman filtering as a least square problem

$$\min_{\boldsymbol{x}(t)} J_k = \sum_{t=1}^{k} [\boldsymbol{z}(t) - H\boldsymbol{x}(t)]^T Q^{-1} [\boldsymbol{z}(t) - H\boldsymbol{x}(t)] + \sum_{t=1}^{k-1} [\boldsymbol{x}(t+1) - A\boldsymbol{x}(t) - B\boldsymbol{u}(t)]^T R^{-1} [\boldsymbol{x}(t+1) - A\boldsymbol{x}(t) - B\boldsymbol{u}(t)] \tag{10.9}$$

Notice that the new objective function satisfies

$$\begin{aligned} \min_{\boldsymbol{x}(t)} J_k = J_{k-1} &+ [\boldsymbol{z}(k) - H\boldsymbol{x}(k)]^T Q^{-1} [\boldsymbol{z}(k) - H\boldsymbol{x}(k)] \\ &+ [\boldsymbol{x}(k) - A\boldsymbol{x}(k-1) - B\boldsymbol{u}(k-1)]^T R^{-1} [\boldsymbol{x}(k) - A\boldsymbol{x}(k-1) - B\boldsymbol{u}(k-1)] \end{aligned} \tag{10.10}$$

we can obtain the standard recursive form of Kalman filter [5] as

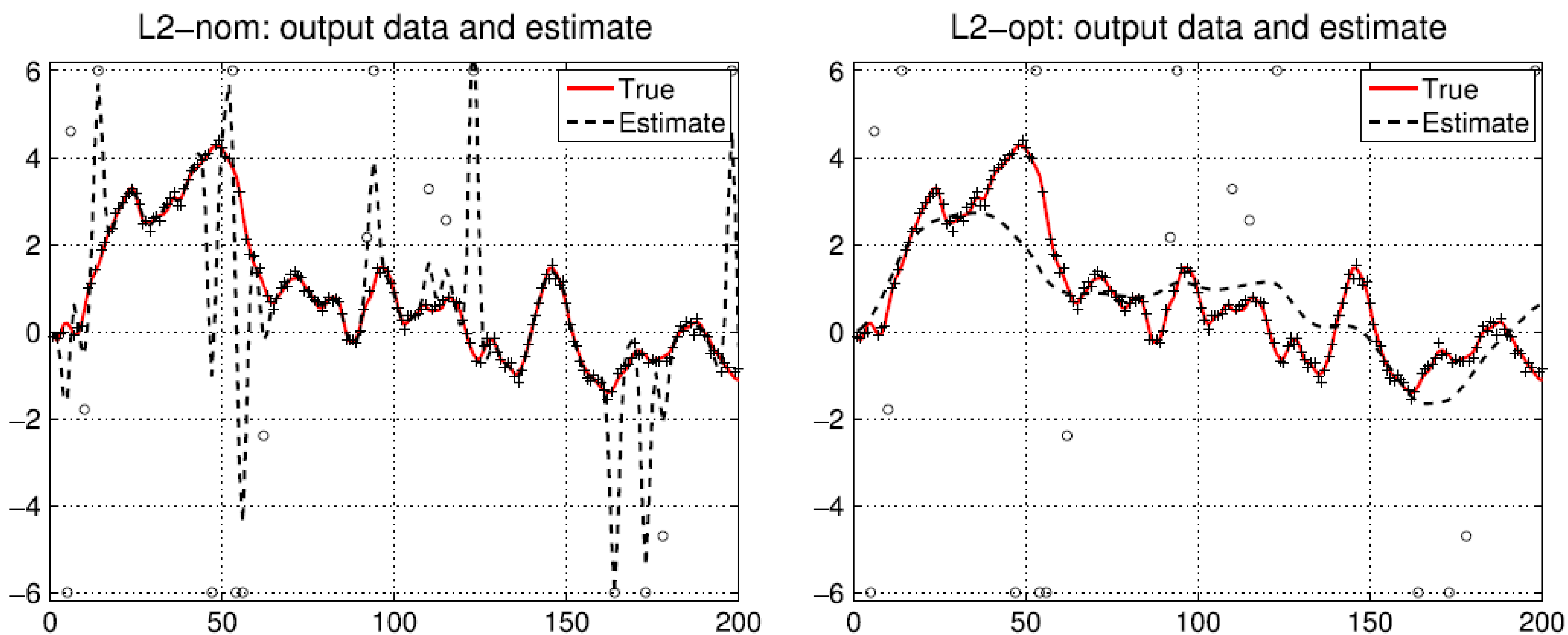
$$K_{t+1} = \left(AP_tA^T + Q\right) H^T \left[H\left(AP_tA^T\right) H^T + R\right]^{-1} \tag{10.11}$$

$$\hat{\boldsymbol{x}}(t+1) = A\hat{\boldsymbol{x}}(t) + Bu(t) + K_{t+1}\left[z(t+1) - H\left(A\hat{\boldsymbol{x}}(t) + Bu(t)\right)\right] \tag{10.12}$$

$$P_{t+1} = \left(I - K_{t+1}H\right)\left(AP_tA^T + Q\right) \tag{10.13}$$

where $\hat{\boldsymbol{x}}(t+1)$ is the recursive estimation of the state $\boldsymbol{x}(t)$ at time t. K_t and P_t are the corresponding estimated covariance matrices.

10.1. Kalman Filtering



10.2. L_1 Filtering and Total Variation

l_1 filtering aims to suppress the noise of one-dimensional signal. It produces piecewise linear trend estimation results and is therefore fitful to time series with an underlying piecewise linear trend.

Given a scalar time series $y(t)$, $t = 1, \ldots, k$, l_1 filtering aims to obtain an estimated scalar time series $x(t)$ [9]-[11], such that

$$\min_{x(t)} \sum_{t=1}^{k} [y(t) - x(t)]^2 + \lambda \sum_{t=2}^{k-1} |x(t-1) - 2x(t) + x(t+1)| \tag{10.14}$$

where $\lambda \geq 0$ is the regularization parameter used to control the trade-off between smoothness of $x(t)$ and the size of the residual $y(t)x(t)$.

Differently, HodrickPrescott filtering [12]-[13] obtains an estimated scalar time series $x(t)$, such that

$$\min_{x(t)} \sum_{t=1}^{k} [y(t) - x(t)]^2 + \lambda \sum_{t=2}^{k-1} [x(t-1) - 2x(t) + x(t+1)]^2 \tag{10.15}$$

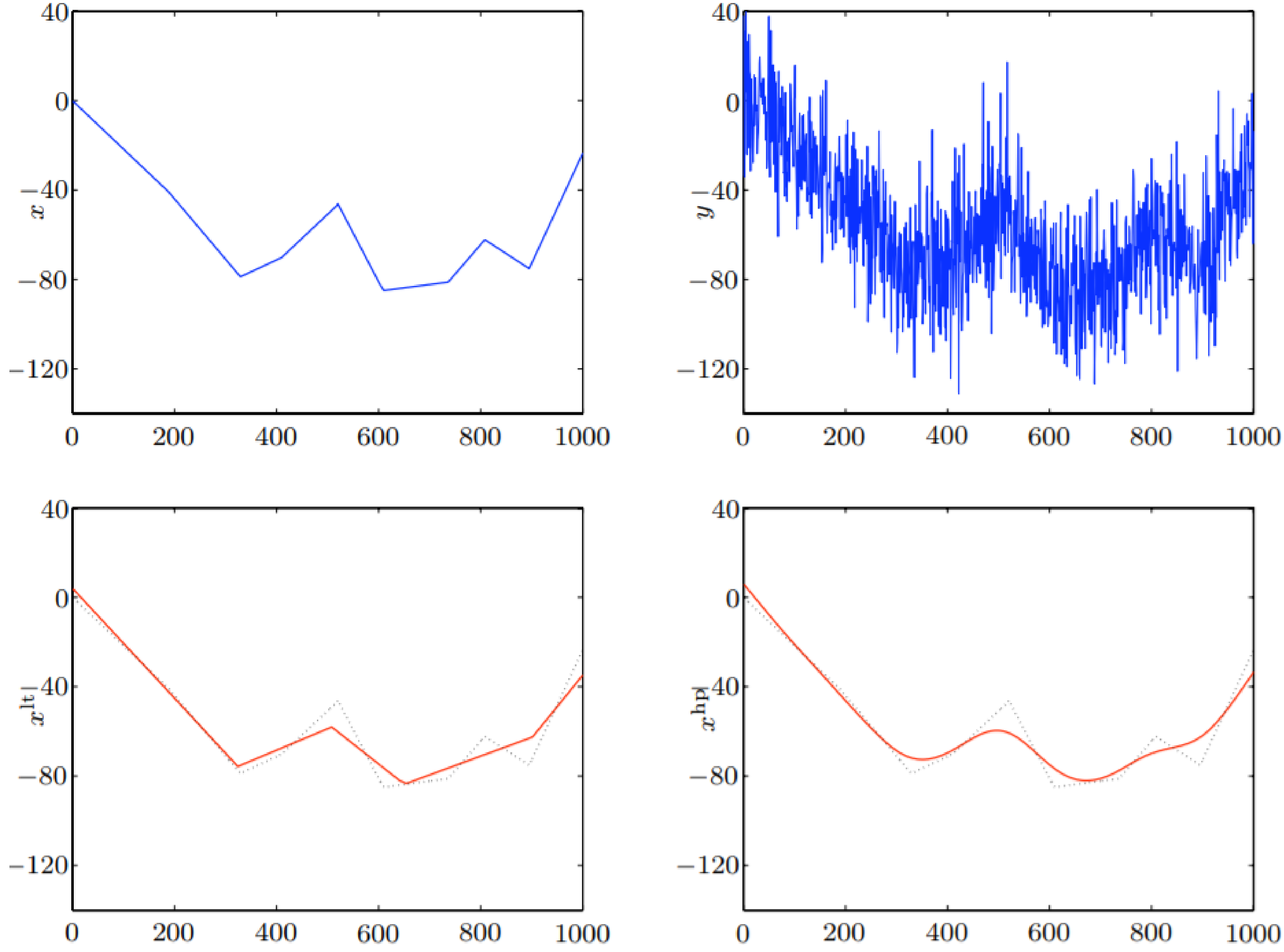
Clearly, l_1 applies 1-norm regularization and HodrickPrescott filtering applies Tikhonov regularization for the twice-order difference of the scalar time series $x(t)$.

In contrast, the one dimensional Total Variation (TD) denoising [14]-[18] obtains an estimated scalar time series $x(t)$, such that

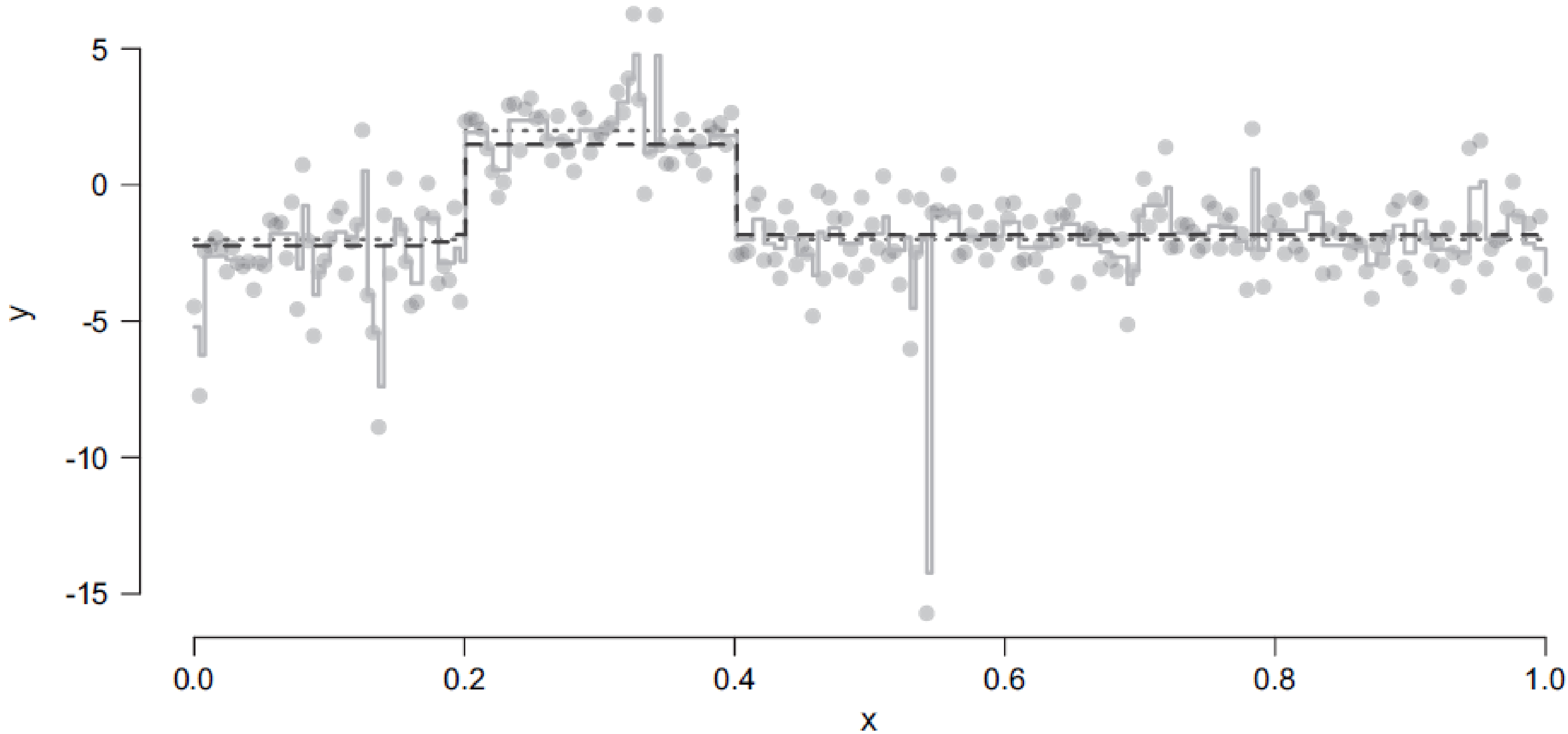
$$\min_{x(t)} \sum_{t=1}^{k} [y(t) - x(t)]^2 + \lambda \sum_{t=2}^{k} |x(t-1) - x(t)| \tag{10.16}$$

Clearly, it applies 1-norm regularization for the first-order difference of the scalar time series $x(t)$.

10.2. L_1 Filtering and Total Variation



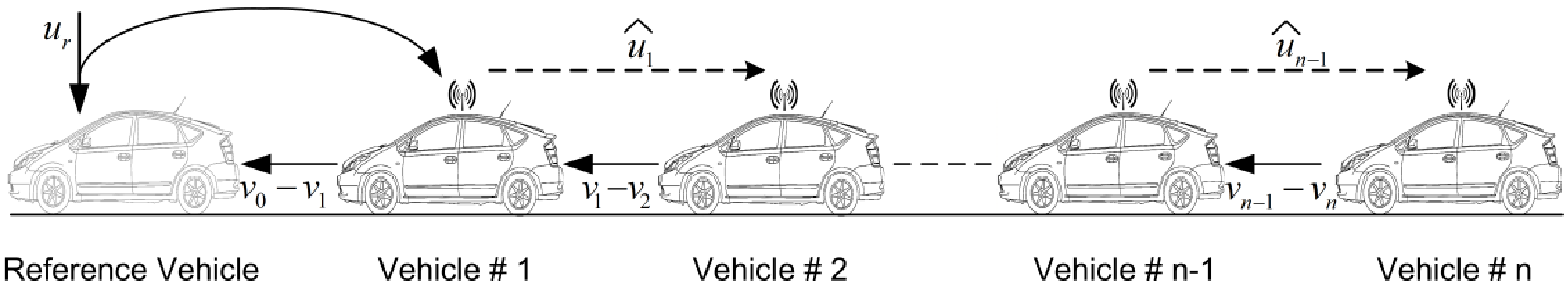
10.2. L_1 Filtering and Total Variation



10.3. 轨迹规划

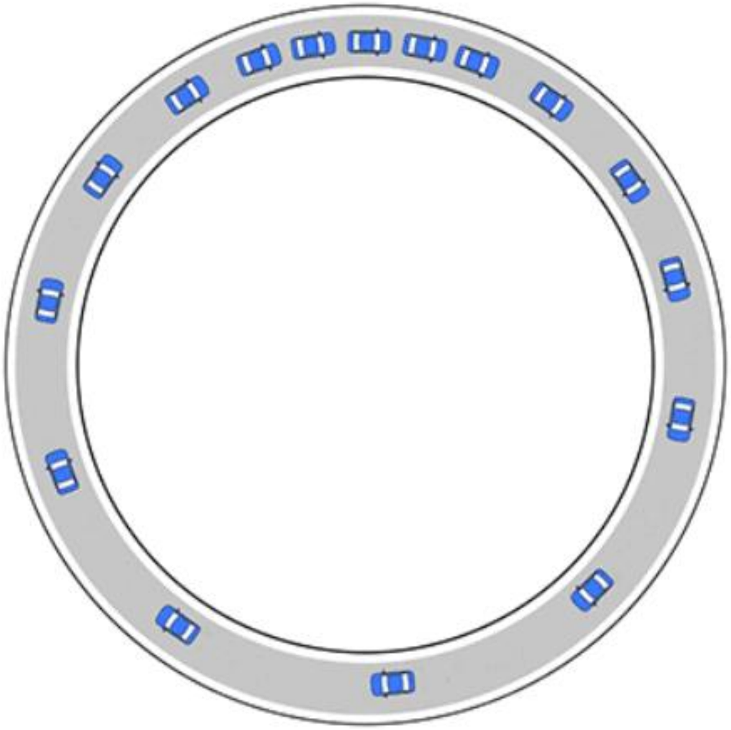
一维协同驾驶主要关注一个车道上多车同方向跟随运动的场景。

研究者特别关注弦稳定性 String Stability，希望通过合适的通讯和控制实现：由于扰动形成的车间距变化向上游传播的过程中，其幅值按排序严格衰减，避免幽灵阻塞产生



10.3. 轨迹规划

弦稳定性 String Stability 和幽灵阻塞



10.3. 轨迹规划

特别的，我们关注考虑稀疏性的一维协同驾驶

- 如果将连续的多车控制行为视为时间序列，则其中非零元素是稀疏 Sparse/Parsimonious

- 如果将连续的多车通讯行为视为时间序列，则其中非零元素是稀疏 Sparse/Parsimonious

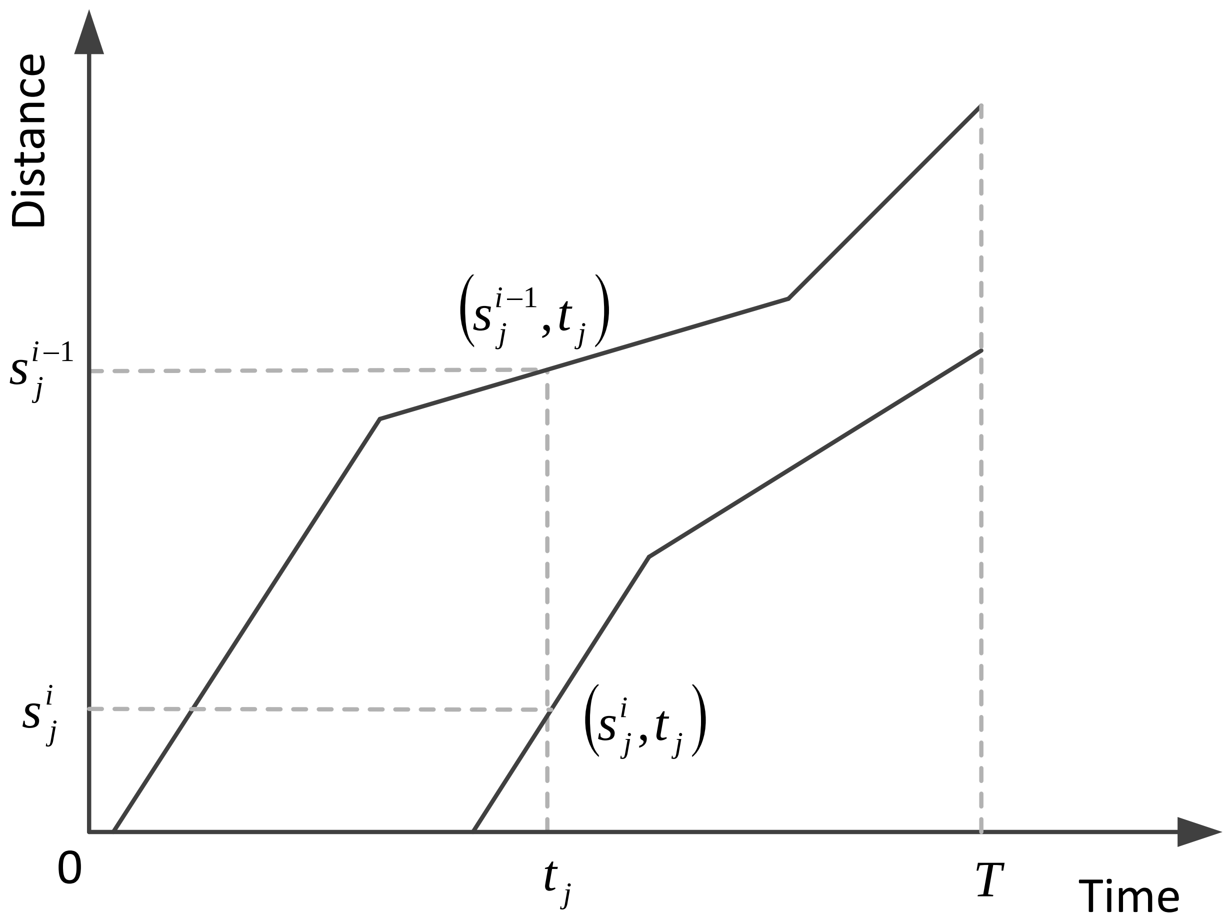
好处包括

- 节能降耗

- 减少信息交互的歧义，提高控制鲁棒性

10.3. 轨迹规划

一维驾驶的离散时间轨迹规划一般框架：决策变量



10.3. 轨迹规划

一维驾驶的离散时间轨迹规划一般框架：约束条件

最大/最小加速度：$$-d_{\max} \le \frac{\frac{s_{j+1}^i - s_j^i}{\delta} - \frac{s_j^i - s_{j-1}^i}{\delta}}{\delta} \le a_{\max}$$

最大/最小速度：$$0 \le \frac{s_j^i - s_{j-1}^i}{\delta} \le v_{\max}$$

碰撞避免：$$g_{\min} \le s_j^{i-1} - s_j^i - l \le g_{\max}$$

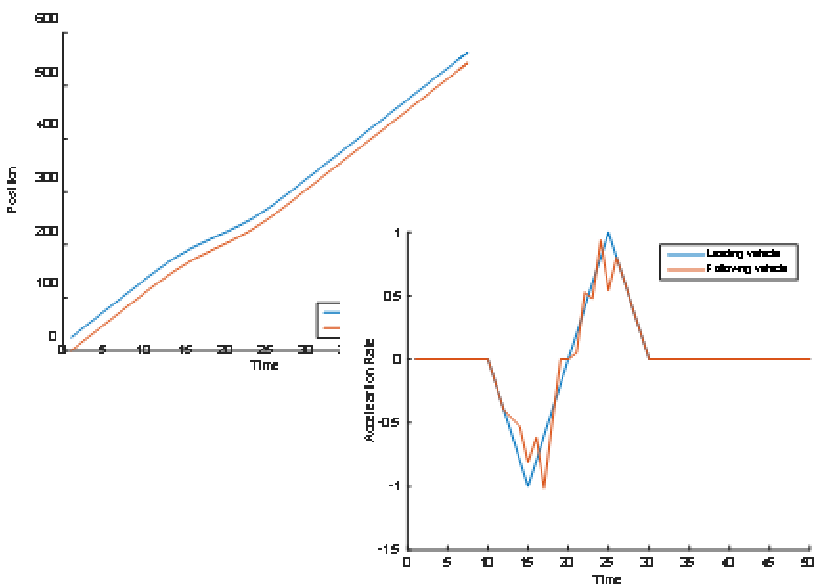
边界条件：$$s_0^i = s_{\text{begin}}^i \quad \frac{s_1^i - s_0^i}{\delta} = v_{\text{begin}}^i$$

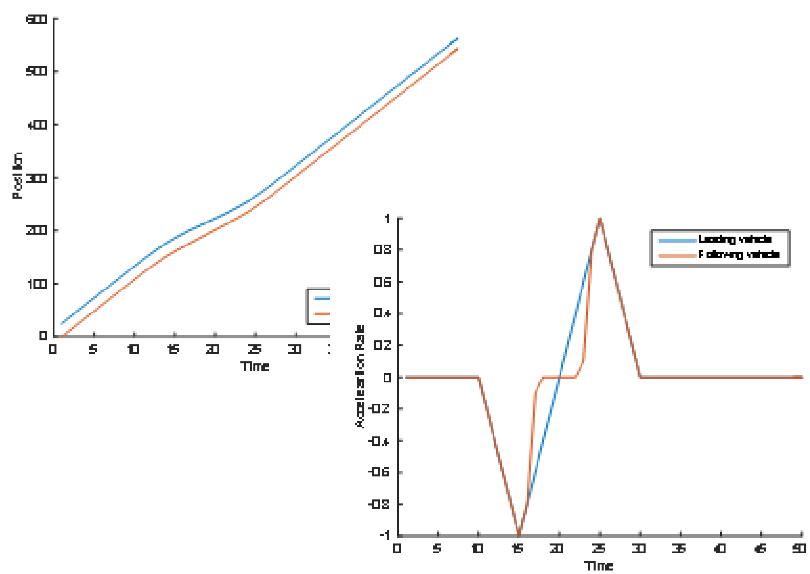
10.3. 轨迹规划

一维协同驾驶的离散时间轨迹规划一般框架：目标函数

光滑性： $$\text{Smoothness}^i = \max_j \left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right|$$

稀疏性： $$\text{Parsimoniousness}^i = \sum_j \text{sgn}\left| \left(s_j^i - s_{j-1}^i\right) - \left(s_{j+1}^i - s_j^i\right) \right|$$

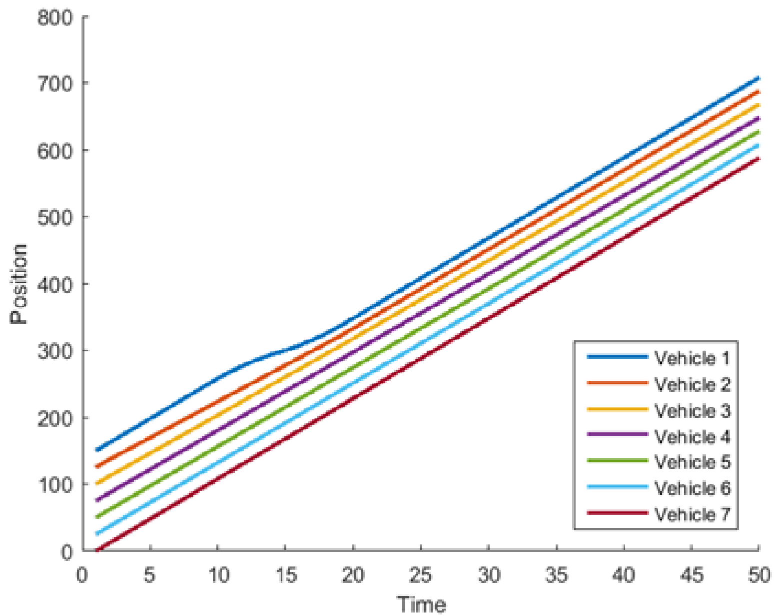




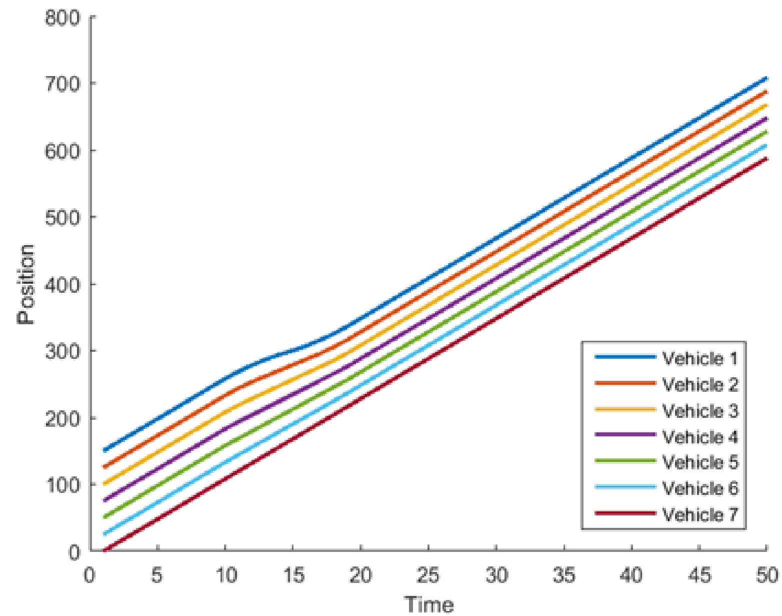
10.3. 轨迹规划

案例研究：堵塞消散

弦稳定性： $s_j^{i-1} - s_j^{i} \le s_{j-1}^{i-1} - s_{j-1}^{i}$



without string stability



with string stability

10.3. 轨迹规划

案例研究：节能驾驶

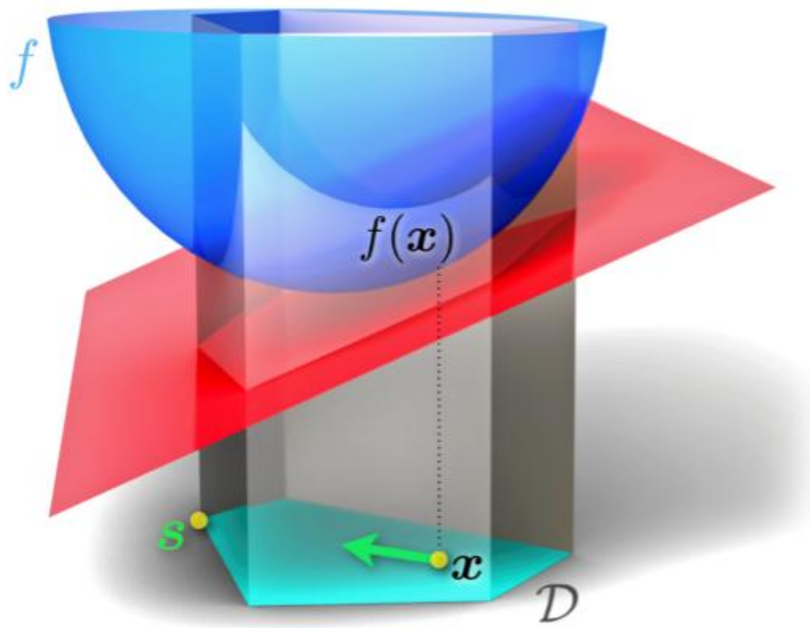
复杂的能耗函数： $J=\sum_{j=2}^{n} F_j$

$$F_j=\begin{cases}\sum_{p=0}^{3}\sum_{q=0}^{3} W_{p,q} v_j^p a_j^q & \text{for } a_j \geq 0 \\ \sum_{p=0}^{3}\sum_{q=0}^{3} M_{p,q} v_j^p a_j^q & \text{for } a_j < 0\end{cases}$$

$$v_j^i=\frac{s_j^i-s_{j-1}^i}{T/n} \quad a_j^i=\frac{\frac{s_{j+1}^i-s_j^i}{T/n}-\frac{s_j^i-s_{j-1}^i}{T/n}}{T/n}$$

10.3. 轨迹规划

采用 Frank-Wolfe 算法迭代求解次优解



Frank-Wolfe 算法一次迭代示意图
（图片来源：Stephanie Stutz）

初始化：寻找一个符合约束的解；

Step 1：求解方向寻找子问题：

$$\min_{s_j^i} s^T \nabla J_{\text{sparse}}(\{x_j^i|^{(k)}\})$$
$$s.t.\, s \in \Omega$$

Step 2：确定搜索步长；

Step 3：更新结果。

10.3. 轨迹规划

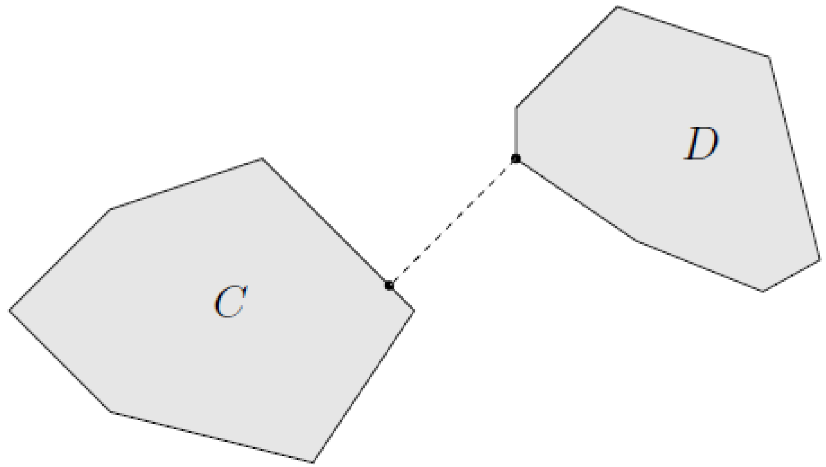
Frank-Wolfe 算法 vs. 遗传算法

遗传算法经过 50,000 次迭代，得到局部最优的目标函数值为 4.42。Frank-Wolfe 算法经过 4 次迭代，得到局部最优的目标函数值为 4.67。因此，所获得的目标函数值差异小于 3%，这对于大多数应用都是可以接受的

而遗传算法至少花费了 10 倍于 Frank-Wolfe 算法

10.3. 轨迹规划

避撞轨迹规划



Euclidean distance between polyhedra

Let C and D be two polyhedra described by the sets of linear inequalities $A_1x \preceq b_1$ and $A_2x \preceq b_2$, respectively. The distance between C and D is the distance between the closest pair of points, one in C and the other in D, as illustrated in figure 8.2. The distance between them is the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & \|x-y\|_2 \\ \text{subject to} & A_1x \preceq b_1 \\ & A_2y \preceq b_2. \end{array} \tag{8.4}$$

We can square the objective to obtain an equivalent QP.

10.3. 轨迹规划

The dual of the problem (8.3) of finding the distance between two convex sets has an interesting geometric interpretation in terms of separating hyperplanes between the sets. We first express the problem in the following equivalent form:

$$\begin{array}{ll} \text{minimize} & \|w\| \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & g_i(y) \le 0, \quad i = 1, \ldots, p \\ & x - y = w. \end{array} \tag{8.5}$$

The dual function is

$$g(\lambda, z, \mu) = \inf_{x,y,w} \left(\|w\| + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \mu_i g_i(y) + z^T(x - y - w) \right)$$

$$= \begin{cases} \inf_x \left(\sum_{i=1}^m \lambda_i f_i(x) + z^T x \right) + \inf_y \left(\sum_{i=1}^p \mu_i g_i(y) - z^T y \right) & \|z\|_* \le 1 \\ -\infty & \text{otherwise,} \end{cases}$$

which results in the dual problem

$$\begin{array}{ll} \text{maximize} & \inf_x \left(\sum_{i=1}^m \lambda_i f_i(x) + z^T x \right) + \inf_y \left(\sum_{i=1}^p \mu_i g_i(y) - z^T y \right) \\ \text{subject to} & \|z\|_* \le 1 \\ & \lambda \succeq 0, \quad \mu \succeq 0. \end{array} \tag{8.6}$$

10.3. 轨迹规划

We can interpret this geometrically as follows. If λ, μ are dual feasible with a positive objective value, then

$$\sum_{i=1}^{m} \lambda_i f_i(x) + z^T x + \sum_{i=1}^{p} \mu_i g_i(y) - z^T y > 0$$

for all x and y. In particular, for $x \in C$ and $y \in D$, we have $z^T x - z^T y > 0$, so we see that z defines a hyperplane that strictly separates C and D.

Therefore, if strong duality holds between the two problems (8.5) and (8.6) (which is the case when (8.5) is strictly feasible), we can make the following conclusion. If the distance between the two sets is positive, then they can be strictly separated by a hyperplane.

Separating polyhedra

Applying these duality results to sets defined by linear inequalities $A_1x \preceq b_1$ and $A_2x \preceq b_2$, we find the dual problem

$$\begin{array}{ll} \text{maximize} & -b_1^T\lambda - b_2^T\mu \\ \text{subject to} & A_1^T\lambda + z = 0 \\ & A_2^T\mu - z = 0 \\ & \|z\|_* \leq 1 \\ & \lambda \succeq 0, \quad \mu \succeq 0. \end{array}$$

If λ, μ, and z are dual feasible, then for all $x \in C$, $y \in D$,

$$z^Tx = -\lambda^T A_1 x \geq -\lambda^T b_1, \qquad z^Ty = \mu^T A_2 x \leq \mu^T b_2,$$

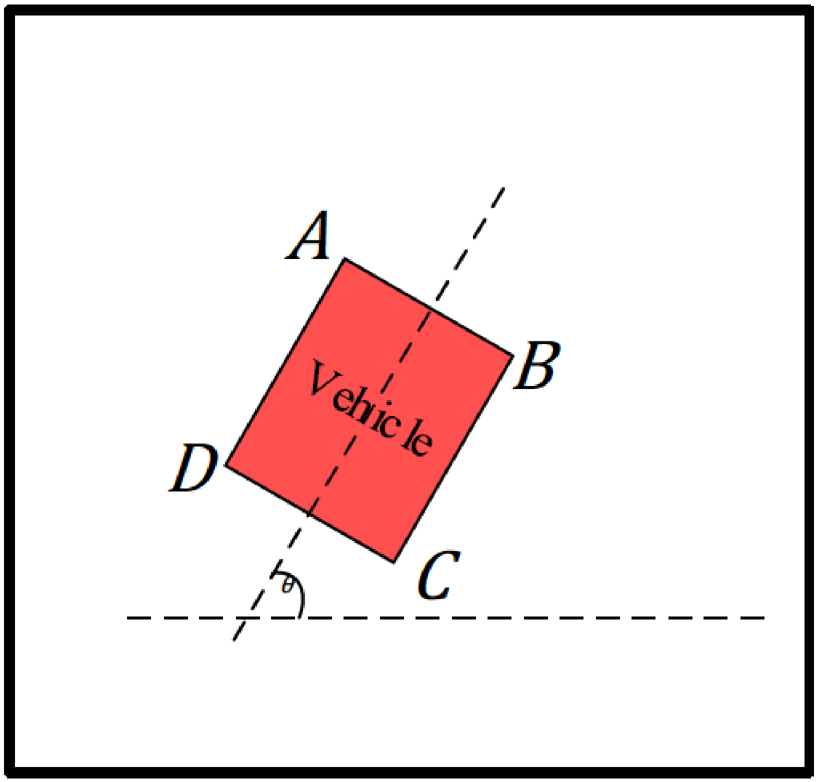
and, if the dual objective value is positive,

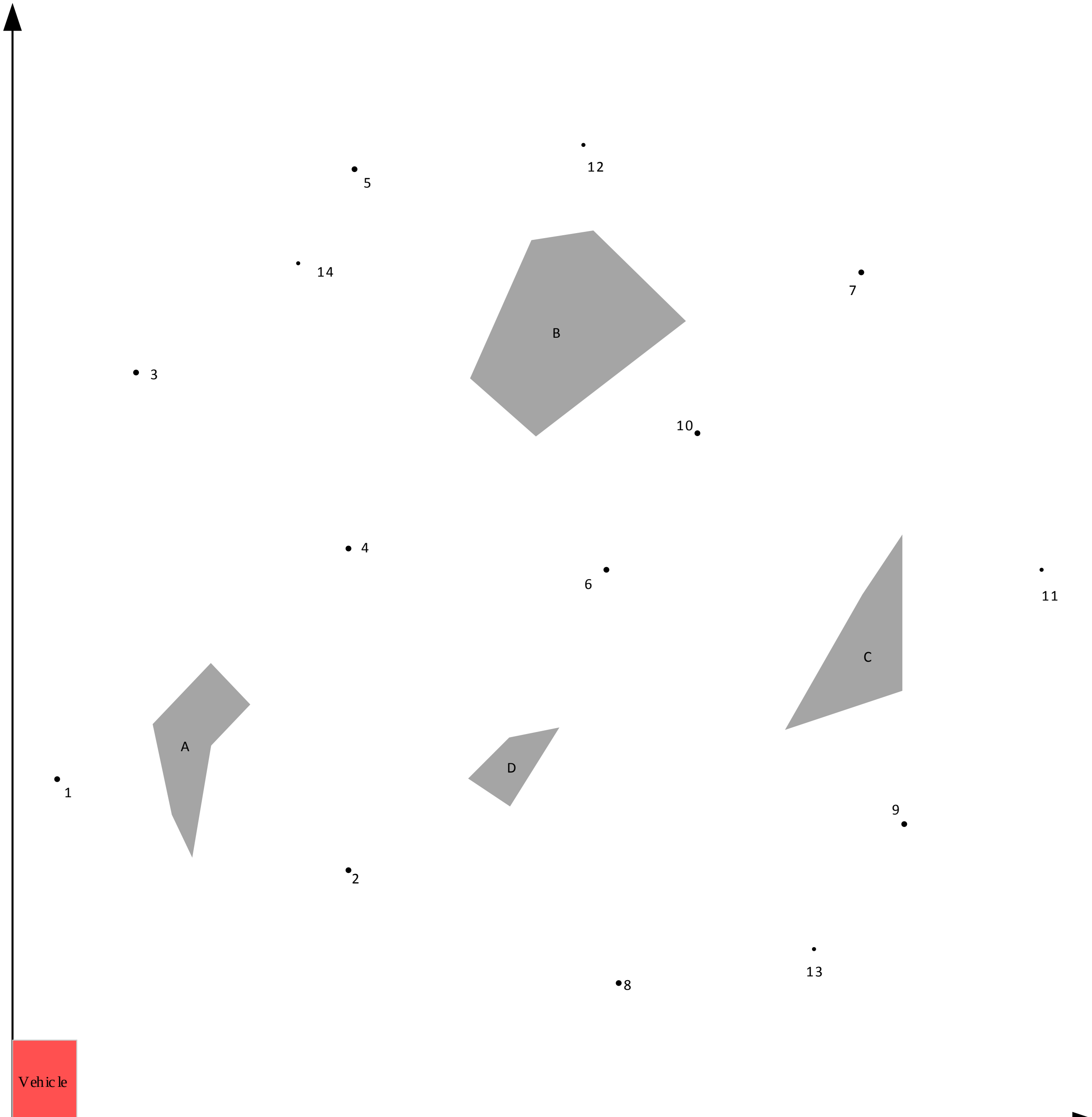
$$z^Tx - z^Ty \geq -\lambda^T b_1 - \mu^T b_2 > 0,$$

i.e., z defines a separating hyperplane.

10.3. 轨迹规划

2020 年课程设计，扫地机器人问题





12
5
14
7
B
3
10
4
6
11
C
A
D
1
9
2
13
8
Vehicle

10.3. 轨迹规划

The planning requirements are as follows:

1) minimize the length of the planned path as much as possible;

2) vehicle must avoid all obstacles;

3) vehicle must clean all "rubbishes";

4) the turning angle of the vehicle should not be too large.

We can characterize a path $\boldsymbol{z}$ by a series of $(n+1)$ waypoints sampled at equal intervals, i.e. $\boldsymbol{z} = [\boldsymbol{z}(0)^T, \boldsymbol{z}(1)^T, \ldots, \boldsymbol{z}(n)^T]^T$. $\boldsymbol{z}(k) = [x(k), y(k), \theta(k)]^T$ represents the state of the vehicle at the kth sampling moment, $P(k) = [x(k), y(k)]^T$ refers to the position of the center of the vehicle, $\theta(k)$ refers to the orientation angle of the vehicle,as shown in Fig.1.

Define the decision variables as $\boldsymbol{z}$ and consider the above requirements, the corresponding optimization problem can be formulated as below.

The Objective Function

To achieve the first requirement that minimize the path, we can define the objective function as

$$\min_{\boldsymbol{z}} \ J(\boldsymbol{z}) = \sum_{k=0}^{n-1} \left[\big(x(k+1) - x(k)\big)^2 + \big(y(k+1) - y(k)\big)^2 \right] \tag{1}$$

To meet the second requirement, the rectangular vehicle should not intersect with all polygonal obstacles at all sampling moment. If the obstacles are convex, since both the rectangular vehicle and polygonal obstacles can be defined as intersecting halfspaces, we can check collisions by judging whether the feasible regions constructed by these halfspaces are empty. For non-convex obstacles, we can split them into convex sub-obstacles or approximate these non-convex obstacles by their convex hull.

10.3. 轨迹规划

To meet the third requirement, we can define the constraints as

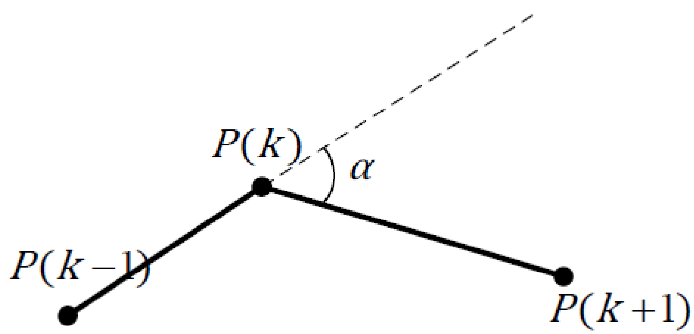
$$\forall m_i, \exists j, \text{s.t.} m_i \in V(j). \tag{2}$$

where m_i refers to the position of the ith "rubbish" and $V(j)$ refers to the area occupied by the vehicle at the jth sampling moment.

As shown in Fig.3, we define the turning angle of the vehicle, $\alpha\big(P(k)-P(k-1), P(k+1)-P(k)\big)$, as the intersection angle between the vector $(P(k)-P(k-1))$ and $(P(k+1)-P(k))$. As mentioned above, $P(k)=[x(k),y(k)]^T$ refers to the position of the center of the vehicle at the kth sampling moment. Therefore, to meet the last requirement, we can define the constraints as

$$\alpha\big(P(k)-P(k-1), P(k+1)-P(k)\big) \le \alpha_{max}. \tag{3}$$

where α_{max} is the maximum permittable turning angle.



10.3. 轨迹规划

杨润钊 陈一帆 谢佳辰 严相杰 于铭瑞 组作业

约束条件 1

在清洁车进行垃圾拾取时，定义拾取成功的条件是：清洁车在第 j 采样时刻的覆盖区域 $V(j)$ 覆盖了垃圾点坐标。

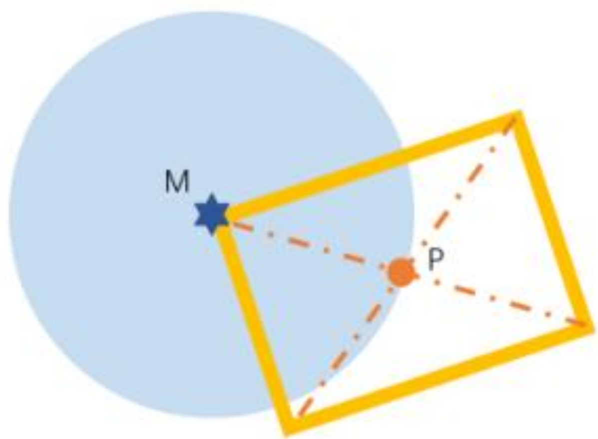


图 4: 清洁圆模型

10.3. 轨迹规划

首先对于问题进行简化。由于在第 j 采样时刻的清洁车角度可以任意定义，因此定义以清洁车对角线为直径，以垃圾点为圆心的圆（后文简称为收集圆），如图4所示。因此当清洁车的中心坐标处于收集圆中时，可以通过旋转垃圾车，使其一角指向垃圾以实现拾取垃圾的要求。

因此针对于该问题的约束条件 1，只需要对于垃圾车的位置进行优化，保证其处于收集圆中，并根据垃圾车与垃圾相对位置确定清洁车方向角，即可完成收集。问题转化为，如何以最短的路径遍历所有清洁圆。

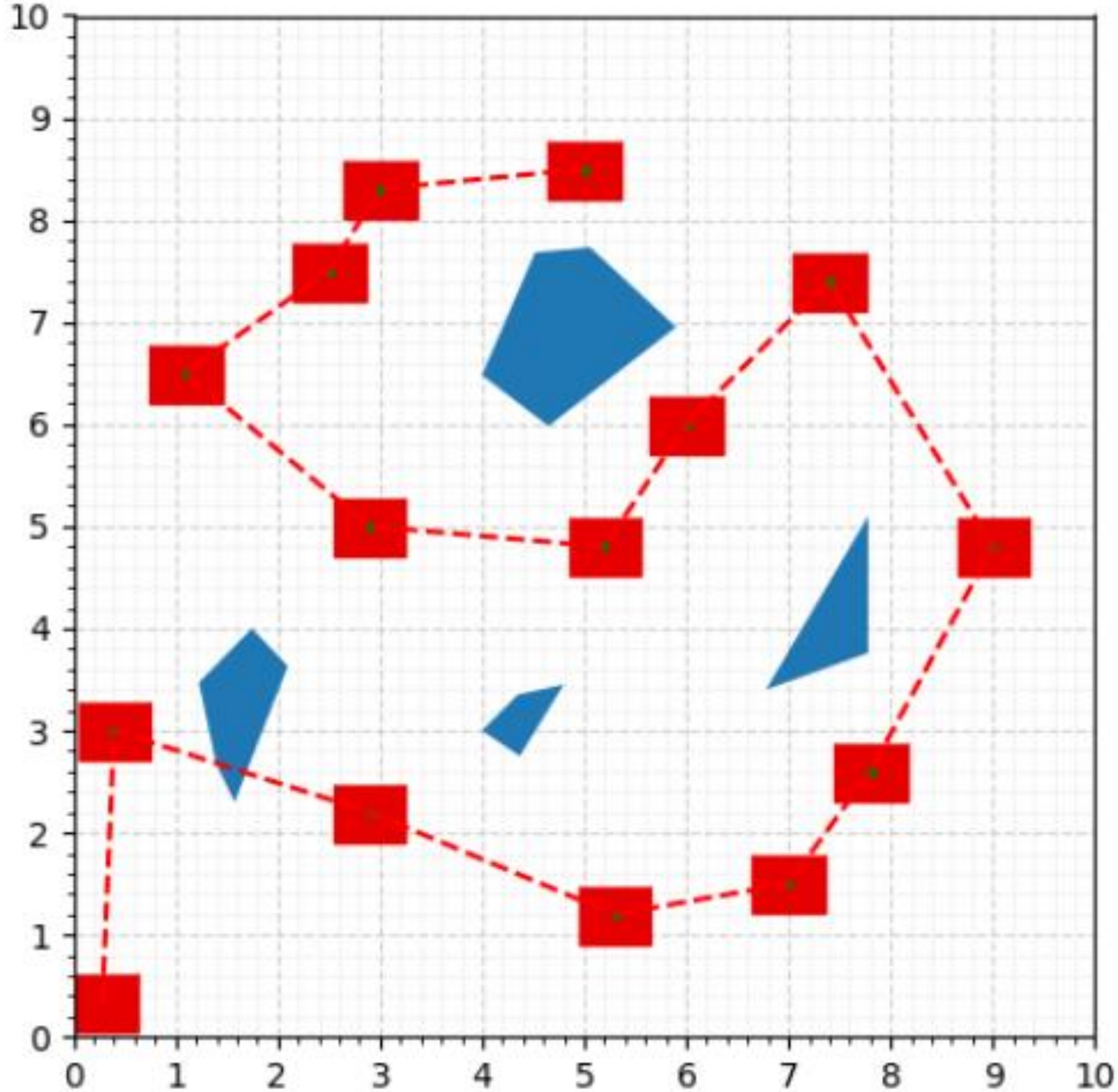
再进一步，可以将该问题分解成解清洁圆的路径顺序 $M(k)$ 与各个清洁圆的接触位置 $P(k)$。如果两个单位圆间没有障碍，其间距离小于两圆心连线距离，大于两圆心连线距离减清洁圆直径。因此可以的不同的路径顺序进行上下界估计，大大缩小其可能解的范围。再通过凸优化方式对于接触位置进行优化，以求得该遍历顺序下最短路径，如图5所示，并排除所以下界小于该最优解的遍历顺序。最后对于可能解的优化结果进行排序，即可选出最优解。

首先确定路径顺序。

以清洁圆圆心作为需要遍历中心位置。这是一个旅行推销员问题（Travelling salesman problem, TSP），该问题有大量次优解求解方法。但是由于本问题位置点较少较为简单，因此本文使用较为保守的遍历方式进行最优解搜索。首先计算清洁圆之间的距离。同时因为存在障碍物干扰，因此需要针对连线之间有障碍物的圆的距离进行适当矫正，并获得所有清洁圆之间的距离矩阵。进而对于所有路径方式进行遍历。

在遍历过程中为了加速运算，进行部分经验性约束。在搜索下一个路径点时，只遍历距当前点距离最短的 n 点；同时如果下一个路径点的距离大于最近待搜路径点的 k 倍时，直接舍弃。可以通过选择 k 与 n 改变遍历效果。

10.3. 轨迹规划



10.3. 轨迹规划

然后确定接触位置。

我们定义如下优化问题

$$\min_{\boldsymbol{z}}\ J(\boldsymbol{z})=\sum_{k=0}^{n-1}\parallel P(k+1)-P(k)\parallel_2 \tag{5}$$

$$\begin{aligned} s.t.\ \ &\parallel P(1)-M(1)\parallel_2\leq\ d/2\\ &\cdots\\ &\parallel P(n)-M(n)\parallel_2\leq\ d/2 \end{aligned} \tag{6}$$

其中 $P(k)$ 为路径中第 k 个点，$M(k)$ 为以确定的遍历路径中第 k 个清洁圆心，d 为小车的对角线长度，即清洁圆大小。该问题为凸问题，可以使用凸优化方法求解，获得每个清洁圆的接触位置 $P(k)$。

10.3. 轨迹规划

我们基于所谓的可视图路径规划算法解决原优化问题的避障约束，最原始的可视图法由 Lozano-Perez 和 Wesley 在 1979 年提出 [1]，是机器人领域全局运动规划的经典算法，应用到我们的场景中，将清洁车建模为质点，障碍物用多边形表示，设 V_o 是所有障碍物的顶点构成的集合，将起始点 S、目标点 G 和障碍物的顶点进行组合连接形成可视图，要求起始点和障碍物各顶点之间、目标点和障碍物各顶点之间以及各障碍物顶点与顶点之间的连线均不能穿越障碍物，即直线是“可视的”，给图中的边加上权值，即构成可视图 $G(V,E)$，其中点集 $V=V_o\bigcup\{S,G\}$，如图6所示，O_1,O_2 为两个已知大小和位置的多边形障碍物，E 为所有“可视边”的集合。由于任意两直线的顶点都是“可视”的，从当前位置沿着这些直线到达目标点的所有路径均是车的无碰路径。搜索最优路径的问题就转化为从起始点 S 到目标点 G 经过这些可视直线的最短距离问题。

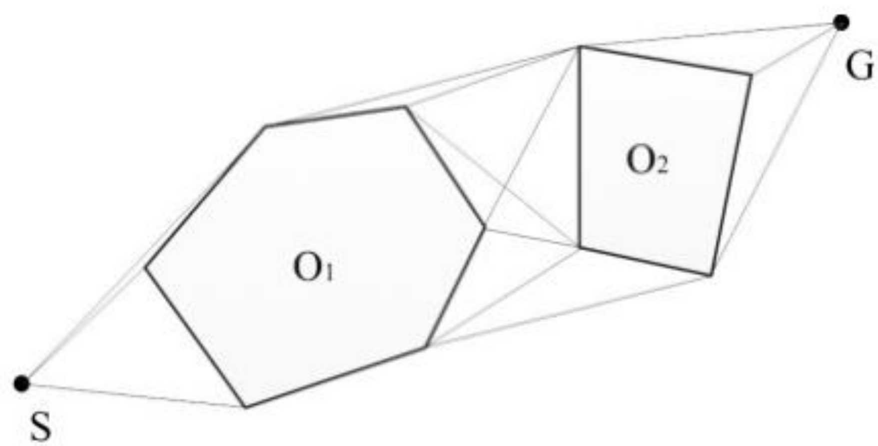


图 6: 可视图示意图

10.3. 轨迹规划

然而实际场景中，清洁车的尺寸不能忽略，我们对多边形障碍物的轮廓进行 padding 处理，填充大小为清洁车的半短轴，并且多边形顶点附近通过增加采样点使得轮廓更加光滑，如图7所示，用清洁车质心位置代表车在某时刻的位置，多边形障碍物某一转角用 $\overline{P_1OP_2}$ 表示，填充后的障碍物轮廓用 $\overline{P_1'A_1}$、$\widehat{A_1A_4}$ 和 $\overline{A_4P_2'}$ 表示，$\overline{SA_2}$，$\overline{GA_3}$，$\overline{P_1'A_1}$，$\overline{P_2'A_4}$ 分别是圆 O 的四条切线，且 $\overline{OA_1} = \overline{OA_2} = \overline{OA_3} = \overline{OA_4} =$ 清洁车半短轴，那么在无障碍时，注意到 $\overline{SG}$ 会直接穿越障碍物，从起始点 S 到目标点 G 最短路径为 $\overline{SOG}$，在有障碍物时最短路径为 $\overline{SA_2} \rightarrow \widehat{A_2A_3} \rightarrow \overline{A_3G}$。

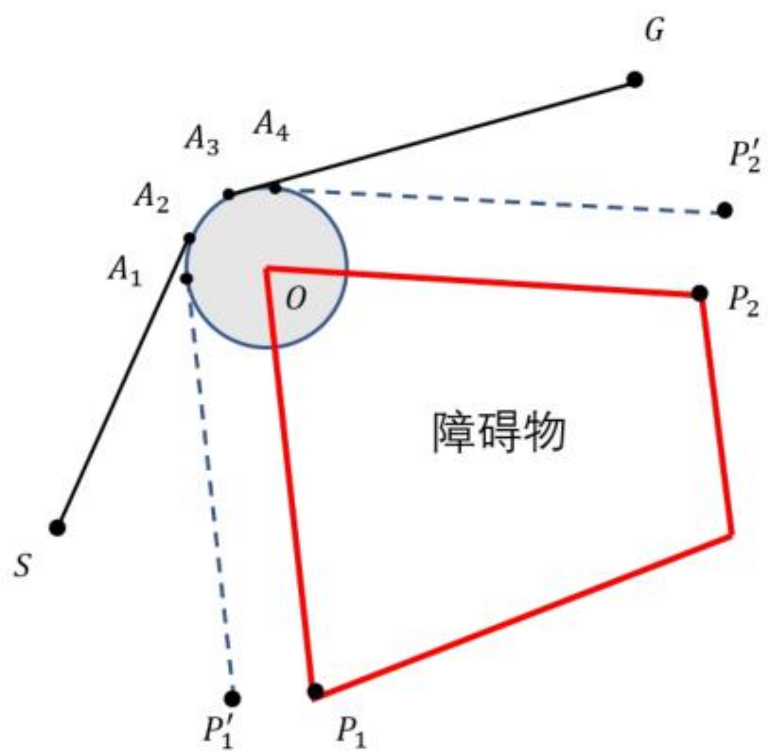


图 7: 避障示意图

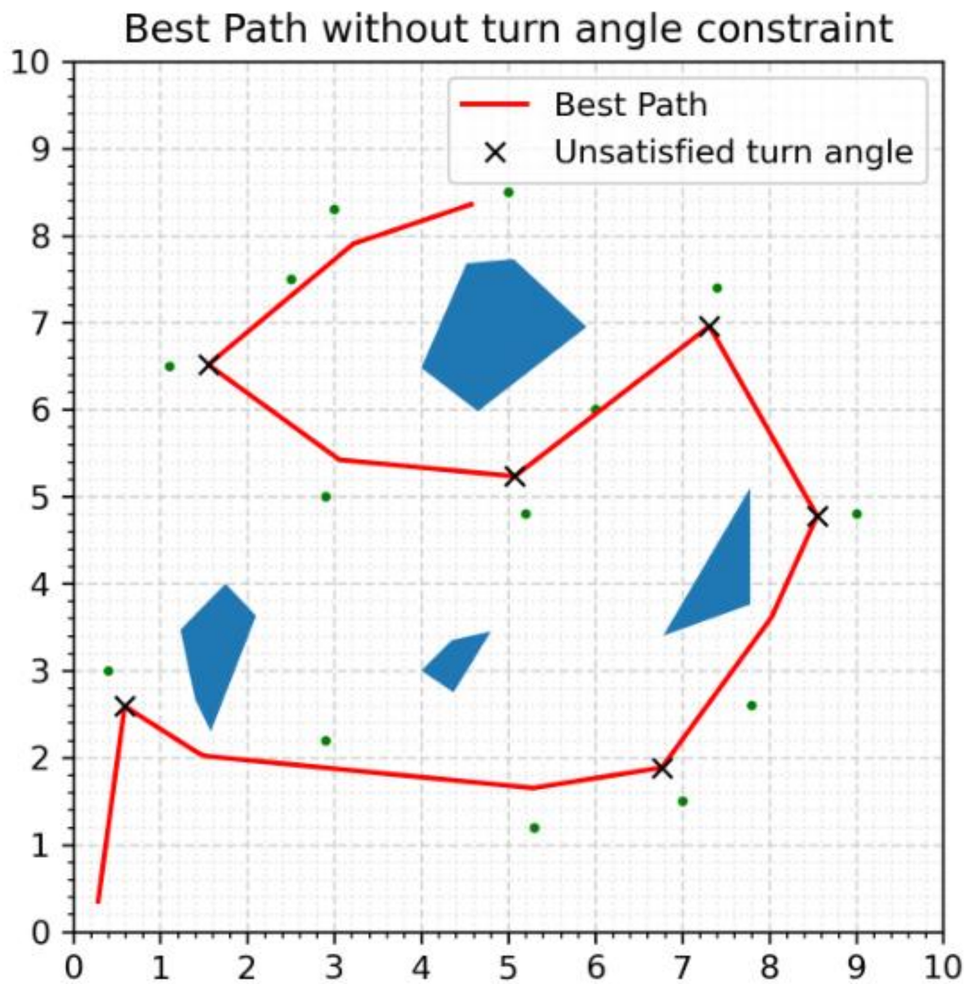


图 8: 无转向角度约束下最优路径

其中黑色叉号标注的是不满足式 (4) 中的转角约束的路径点。为了解决此约束，我们用后处理的思想，对图8中展示的路径进行修正，即可得到满足式 (4) 的新路径。同时，对约束 1 做进一步的放缩，即可得到满足以上所有约束的最终路径，具体方法如下。

10.3. 轨迹规划

算法 1 转角修正

给定: 最佳路径 $\{P_i\}$

输出: 修正后的最佳路径 $\{P_i\}$

1: **重复运行:**

2: 确定图9中的 P_a, P_b

3: 将 P_2 从 $\{P_i\}$ 中删除

4: 将 P_a, P_b 插入 $\{P_i\}$ 中原 P_2 所在的位置

5: **直到:** $\{P_i\}$ 所有的所有路径点都满足约束

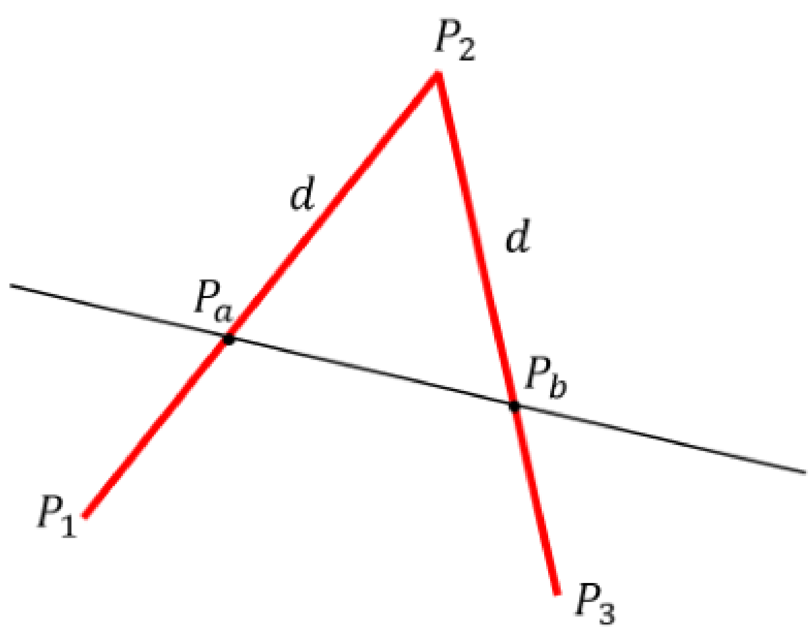


图 9: 路径点修正示意图

在图9中，路径点 P_2 不满足约束。为此，我们分别沿 P_2P_1 与 P_2P_3 两个方向截取长度 d，得到新的路径点

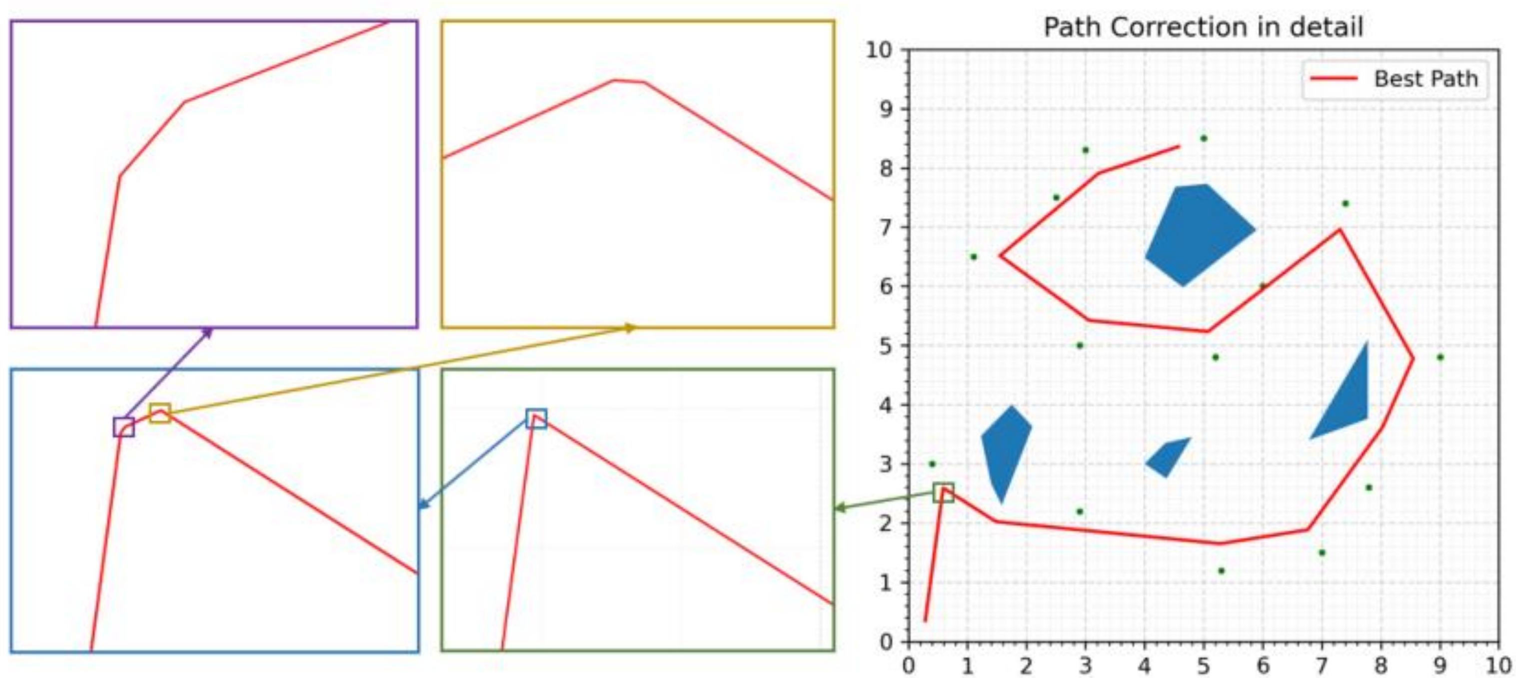
$$
\begin{aligned}
P_a &= P_2 + d \cdot \frac{P_2P_1}{|P_2P_1|} \\
P_b &= P_2 + d \cdot \frac{P_2P_3}{|P_2P_3|}
\end{aligned}
\tag{7}
$$

注意，在算法1中，长度 d 需要随着循环的进行递减，以确保 $d < ||P_xP_y||$，本文采用的递减方法为 $d := d * 0.1$

由于上述方法产生新路径点 P_a, P_b 会“远离”旧路径点 P_2，为了确保修正后的的路径依然满足约束 1，我们在约束 1 的方程6中乘以一个放缩系数 γ，即

$$
\| P(i) - M(i) \|_2 \leq \frac{d}{2} * \gamma, \ 1 \leq i \leq n \tag{8}
$$

图10展示了采用算法1修正后的放大细节。



10.4. References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, vol. 82D, pp. 35-45, 1960.

[2] R. E. Kalman, "New methods in Wiener filtering theory," *Proceedings of the First Symposium of Engineering Application of Random Function Theory and Probability*, pp. 270-388, 1963.

[3] B. Teixeira, "Kalman filters [Ask The Experts]," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 16-18, 2008.

[4] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "l_1 Trend Filtering," *SIAM Review*, vol. 51, no. 2, pp. 339-360, 2009. http://web.stanford.edu/~boyd/papers/l1_trend_filter.html

[5] L. Condat, "A direct algorithm for 1-D total variation denoising," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1054-1057, 2013.

[6] L. Li, F.-Y. Wang, "An integrated design framework for driver/passenger-oriented trajectory planning," *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 1764-1769, 2003.

[7] S. Feng, Y. Zhang, S. Li, Z. Cao, H. Liu, L. Li, "String stability for vehicular platoon control: Definitions and analysis methods," *Annual Reviews in Control*, vol. 47, pp. 81-97, 2019.

[8] L. Li, X. Li, "Parsimonious trajectory design of connected automated traffic," *Transportation Research Part B: Methodological*, vol. 119, pp. 1-21, 2019.

[9] https://github.com/RichealYoung/CleaningRobot