

ПОДХОДЫ К ПЛАНИРОВАНИЮ СКОРОСТИ ДЛЯ НАЗЕМНЫХ БЕСПИЛОТНЫХ ТРАНСПОРТНЫХ СРЕДСТВ

Лившиц А. Б.¹, Темкин И. О.², Фадеев А. Ю.³
(НИТУ МИСИС, Москва)

Статья представляет обзор методов планирования профиля скорости для наземных беспилотных транспортных средств. Технология автономного вождения находится в фазе активной разработки, и многие задачи уже решены, но с недостаточным для безопасной эксплуатации и масштабирования технологии качеством. Таким образом, эта область является перспективной с научной и инженерной точек зрения. В работе подробно разбираются постановки задачи, описываются различные виды ограничений к итоговому решению, а также рассматривается построение целевой функции в зависимости от требований и используемых методов. Особое внимание уделено ST-графу как инструменту моделирования взаимодействия с другими участниками движения. Также детально анализируются практические сценарии взаимодействия агентов в различных ситуациях, часто возникающих в дорожном потоке. Авторы концентрируют внимание на классе методов построения профиля скорости по известной геометрической траектории. В частности, рассматривается два семейства подходов: динамическое программирование и сведение непрерывной задачи к задаче квадратичного программирования. Подробно разбираются существующие современные методы, основанные на этих подходах, а также предлагаются улучшения и доработки в контексте различных дополнительных требований к решению. Статья служит методической основой для разработчиков систем автономного вождения.

Ключевые слова: беспилотные транспортные системы, профиль скорости, динамическое программирование, непрерывная оптимизация, ST-граф.

1. Введение

Беспилотные транспортные средства в последние годы активно развиваются и все больше используются в различных сферах. Одной из наиболее интересных и сложных областей применения беспилотного транспорта является осуществление

¹ Ася Борисовна Лившиц, бакалавр, выпускник (asyalivshits@yandex.ru).

² Игорь Олегович Темкин, д.т.н., профессор (temkin.io@misis.ru).

³ Александр Юрьевич Фадеев, аспирант (frogcatcher@mail.ru).

автономного движения пассажирского (и некоторых типов грузового) транспорта по городским улицам и магистралям, а также по междугородним трассам. Несмотря на то, что на рынке уже появились некоторые разработки, например, Waymo (США) [6], Baidu (Китай), а среди российских компаний активность в области разработки экспериментальных образцов проявляют Яндекс, Камаз, Cognitive Pilot, StarLine, в данный момент речи о широком внедрении этих технологий не идет.

Следует отметить, что ответы на ряд принципиальных вопросов, связанных с возможностью применения автономных автомобилей на дорогах общего назначения, сегодня уже получены. Однако остается ряд задач, без решения которых масштабирование данных технологий невозможно. Одной из таких задач является планирование скорости (Speed planning).

Далее в статье будет подробно описана постановка задачи построения профиля скорости, проанализированы некоторые подходы к решению этой оптимизационной задачи и проведено обсуждение достоинств этих подходов и возникающих при их использовании проблем, а также, будут предложены некоторые возможные решения, которые иллюстрируются конкретными примерами.

2. Постановка задачи планирования профиля скорости

Существует два подхода к решению задачи планирования скорости: методы, одновременно выполняющие вычисление геометрии пути и планирование скорости [3], и методы, которые это делают раздельно – сначала геометрическая траектория [8, 17, 20], затем скорость по уже готовой геометрии [9, 11, 14].

В данной статье рассматриваются методы, занимающиеся исключительно планированием профиля скорости при построенной потенциальной траектории движения.

Рассмотрим движение наземного беспилотного транспортного средства (далее – БТС) вдоль полностью рассчитанной геометрической непрерывной траектории в двумерном евклидо-

вом пространстве, тогда эту кривую можно задать как непрерывные дифференцируемые $x(d)$ и $y(d)$ для параметра $d \in [0, D]$.

Определим $s(d)$ как расстояние вдоль кривой, пройденное от начала до точки $(x(d), y(d))$, т.е.

$$(1) \quad s(d) = \int_0^d \sqrt{(x'(l))^2 + (y'(l))^2} \, dl.$$

Также пусть имеется предсказание траекторий других агентов и препятствий, полученное одним из алгоритмов Motion Prediction [10, 13, 22]. Они даны в следующей форме: для каждой точки на кривой $p(d) = (x(d), y(d))$ и каждого момента времени $t \in [0, T]$ есть информация, будет ли данная точка занята другим агентом/препятствием. Иными словами, произойдет ли коллизия, если БТС окажется через время t в точке $p(d)$.

Пусть в каждой точке $s_k = s(t_k)$ задан радиус кривизны $R_k = R(s_k)$.

Задача состоит в генерации оптимальной (с точки зрения целевой функции, заданной в зависимости от типа транспортного средства, погодных условий, решаемой данным БТС задачи и пр.) функции $s(t)$ – зависимости пройденного расстояния от времени, удовлетворяющей ограничениям. Все остальные параметры – скорость, ускорение в момент времени t – можно получить из функции $s(t)$.

Очевидно, что при планировании скорости движения автономного транспортного средства в потоке требуется учитывать как физические возможности автомобиля [15], так и окружающую обстановку [18], а также не забыть комфорт пассажиров [5, 7, 12]. Для этого требуется ввести ограничения на определенные характеристики системы, затем сформулировав их в рамках поставленной задачи, что подробно изложено в пунктах 2.1–2.2.

В рамках заданных ограничений возможно множество (возможно, бесконечное) потенциальных решений задачи планирования скорости. Среди них нужно выбрать лучшее по метрике, которая может формироваться из следующих компонент:

1. Минимизация времени прохождения траектории – оптимальное распределение скоростей для максимально быстрого преодоления заданного маршрута.

2. Обеспечение комфорта пассажира – учет ограничений на боковые, продольные ускорения и рывки, для минимизации неприятных ощущений у пассажиров.

3. Безопасность при взаимодействии с другими агентами – адаптация скорости в зависимости от положения и действий окружающих объектов (других транспортных средств, пешеходов) для предотвращения возможных столкновений и аварийных ситуаций.

Для получения искомого требуется сначала выбрать метод, с помощью которого будет оптимизировано решение в соответствии с обозначенными целями.

Рассмотрим два крупных подхода для решения задачи: на основе поисковых алгоритмов [4, 19, 21] и на основе непрерывной оптимизации [9, 11, 16, 21]. Подробно целевые функции для каждого из методов описаны в пунктах 2.3–2.4.

2.1. ОГРАНИЧЕНИЯ НА СКОРОСТЬ И ЕЕ ПРОИЗВОДНЫЕ

Для корректного описания движения будем использовать модель Friction Circle [15], изображенную на рис. 1. Ускорение и рывок (jerk) будем рассматривать в двух аспектах: a_{long} – продольной компоненты ускорения, направленной вдоль кривой пути в точке, и a_{lat} – поперечной компоненты, которая направлена перпендикулярно траектории в данной точке. Соответственно, полное ускорение равно

$$(2) \quad a = \sqrt{a_{2,long}^2 + a_{2,lat}^2} .$$

По тем же соображениям для рывка продольная компонента – j_{long} , поперечная – j_{lat} , а полный рывок равен

$$(3) \quad j = \sqrt{j_{2,long}^2 + j_{2,lat}^2} .$$

Если величина ускорения или рывка выходит за границы предельно допустимых параметров, то такое движение становится потенциально опасным из-за риска заноса транспортного средства. Изменение параметров для ускорения и торможения отражается на «Friction Circle», который приобретает вид двух полуэллипсов – верхнего и нижнего.

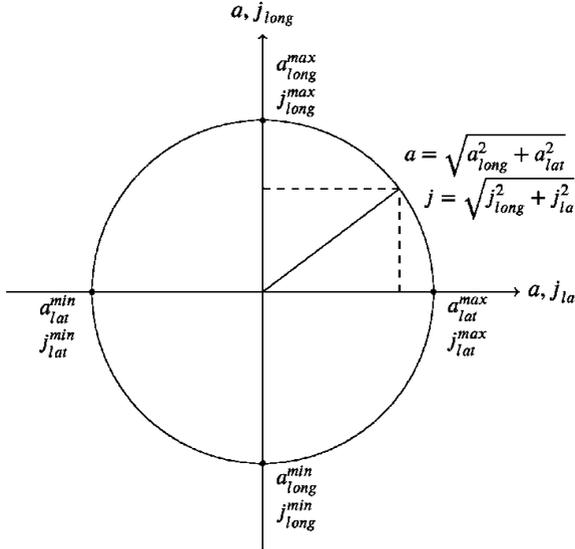


Рис. 1. Модель Friction Circle

Поскольку такие ограничения тяжело включить в модель, зачастую используют упрощенную модель «Friction Box», в которой ограничения аппроксимируются прямоугольником (рис. 2). Эти ограничения иногда называют «box constraints».

Далее перечислены часто встречающиеся виды ограничений:

1. Ограничения на скорость.

Мы рассматриваем движение вдоль кривой пути, т.е. скорость принимает вид функции $v(s)$.

Говоря о движении в городе, стоит упомянуть, что ландшафт, основные препятствия и особенности дороги и окружения известны заранее. К примерам таких особенностей окружения относятся:

- ограничения максимальной скорости на дороге, регулируемое ПДД;
- знаки ПДД («лежачий полицейский», другие дорожные знаки);
- пешеходные переходы;

- «вафельная» разметка на перекрестках, запрещающая остановку на ней;
- финишная точка пути, требующая остановиться.

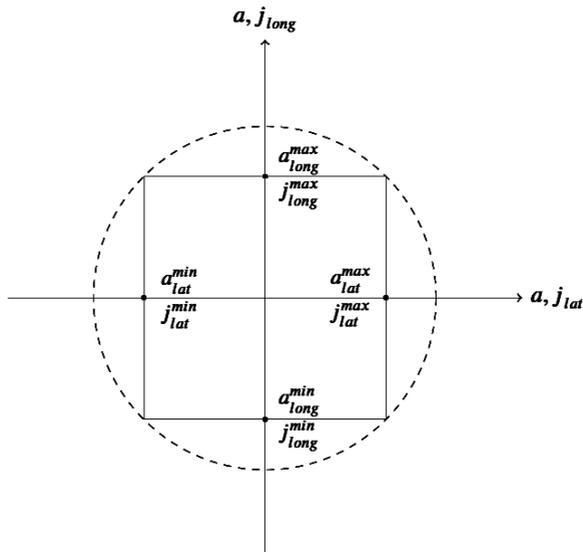


Рис. 2. Ограничения *box constraints*, получившиеся из *Friction Circle*

Такие особенности местности могут накладывать ограничения на скорость, отсюда

$$(4) \quad v(s_k) \leq v_k^{max}$$

либо

$$(5) \quad v(s_k) \geq v_k^{min},$$

где k – определенная точка пути с данным ограничением.

На рис. 3 показан пример возможных ограничений на скорость. На отрезке $[s_1, s_2]$ – крутой поворот, на котором во избежание заноса (из-за большого бокового ускорения) требуется снизить скорость. На отрезке $[s_3, s_4]$ требуется снизить скорость ниже максимальной из-за пешеходного перехода. Поскольку

остановка на нем запрещена, также водится ограничение на минимальную скорость.

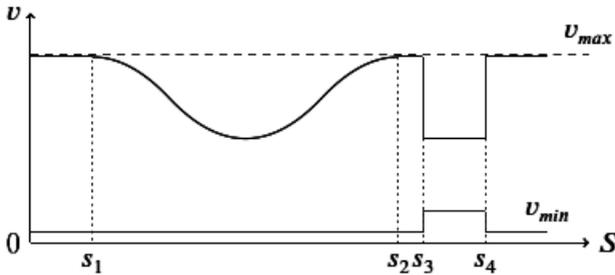


Рис. 3. Зависимость ограничений на скорость от пройденного расстояния вдоль кривой пути

2. Ограничения на продольное ускорение.

Такие ограничения требуется вводить в силу следующих факторов:

- сцепление с дорогой: сила, действующая на автомобиль, не должна превышать силу трения шин об дорогу, чтобы не произошло заноса;
- погодные условия (дождь, снег, наледь и пр.);
- комфорт пассажиров.

Эти ограничения по модели Friction Voh выглядят так (рис. 2):

$$(6) \quad \forall t: a_{min} \leq a_{long}(t) \leq a_{max}.$$

3. Ограничения на боковое (поперечное) ускорение.

Ограничение бокового ускорения помогает поддерживать устойчивость транспортного средства во время поворотов и маневров. Если на транспортное средство действуют слишком большие боковые ускорения, то это может привести к потере сцепления шин с дорогой, что увеличивает риск заноса или опрокидывания.

Боковые ускорения могут вызывать дискомфорт у пассажиров и даже привести к травмам при резких маневрах транспортного средства. Например, если автомобиль резко поворачивает на высокой скорости, пассажиры могут испытывать значитель-

ные силы, которые могут их сместить с сидений или даже выбросить из машины.

Ограничение на поперечное ускорение можно описать формулой

$$(7) \quad a_{lat}^{max} \leq \frac{v^2(t)}{R(s(d))}.$$

Ограничения описываются в виде нелинейной функции на $v(s)$, пример показан на отрезке $[s_1, s_2]$ (рис. 3).

4. Ограничение на продольный и поперечный рывки (jerk).

Ограничение на рывок нужно для комфорта пассажира, так как слишком быстрое изменение ускорения может привести к плохому самочувствию, а также необходимо из-за физических ограничений автомобиля. Ограничение на продольный рывок, по Friction Vox, имеет вид (рис. 2)

$$(8) \quad j_{min} = j_{long}(s) = j_{max}.$$

Ограничение на поперечный рывок – это фактически ограничение на изменение поперечного ускорения:

$$(9) \quad j_{lat}(s) = (a_{lat}(s))'_t.$$

Поэтому его можно включить в ограничение на продольное ускорение (рис. 4).

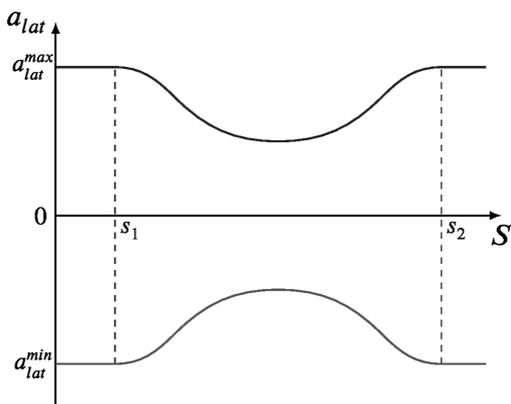


Рис. 4. Ограничения на боковое ускорение на повороте. Для того чтобы держать боковой рывок низким, максимальные значения на самом повороте (отрезок $s_1 - s_2$) требуется снизить

2.2. ОГРАНИЧЕНИЯ ОТ СТАТИЧЕСКИХ И ДИНАМИЧЕСКИХ ПРЕПЯТСТВИЙ

Другие объекты дорожной сцены могут создавать потенциально аварийные ситуации, во избежание которых требуется корректно обработать их виды и сформулировать в виде ограничений на профиль скорости. В отличие от ограничений предыдущего вида, где требовалось соблюдать скоростной режим и пр., данные ограничения концентрируются на взаимодействиях с другими объектами – препятствиями, автомобилями, пешеходами и пр.

Для проработки ограничений, связанных с препятствиями, удобно использовать ST-граф [9, 11].

ST-граф (Spatio-Temporal Graph) – это модель для представления движения и взаимодействия с другими агентами (автомобилями, пешеходами, стационарными препятствиями) в пространстве и времени. Обычно изображается как двумерный график, ось X которого представляет прошедшее от текущего момента время, а ось Y – пройденное расстояние вдоль зафиксированного геометрического пути.

1. Статические ограничения

Примерами статических препятствий могут служить:

- бочка, находящаяся на пути движения, куча песка или другой массивный неподвижный объект;
- неподвижно стоящий автомобиль на пути движения (причиной ситуации может стать авария, авария, стоячая пробка и пр.).

Данные объекты находились все время от начала движения БТС в неподвижном состоянии и будут находиться в нем и далее (рис. 5).

2. Динамические ограничения

Подобная ситуация встречается в разных случаях:

- когда БТС следует за агентом;
- когда агент следует за БТС;
- когда БТС перестраивается в чужую полосу;
- когда другой автомобиль перестраивается в текущую полосу;
- при пересечении перекрестков и пр.

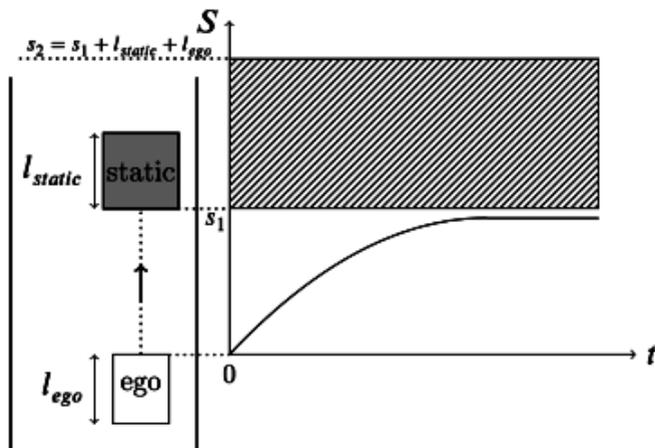


Рис. 5. Пример: закрашенной областью на ST-графе выделены такие точки, в которых статический объект находится в коллизии с БТС: по оси t – от 0 до ∞ , по оси S – от s_1 до s_2 . Профиль скорости пересекаться с этой областью не должен

Более подробный анализ ситуации, когда другой автомобиль встраивается в полосу беспилотного транспортного средства, приведен на рис. 6. Здесь известны момент времени и местоположение на кривой пути s_{coll} , когда он окажется в опасности столкновения. Возможны два решения для избежания аварии: $s_1(t)$ – снизить скорость, пропустив автомобиль вперед; $s_2(t)$ – увеличить скорость, обогнав автомобиль

3. «Мягкие ограничения».

Ограничения на ST-графе, которые рассматривались выше, образовывали на ST-графе регионы столкновений, отображенные на рис. 6. Если профиль скорости пересекает один из таких регионов, то он описывает столкновение БТС с агентом.

Для более детальной обработки ситуаций может помочь введение «мягких» ограничений. Например, соблюдение безопасной дистанции до другого агента является показательным примером, представленным на рис. 7.

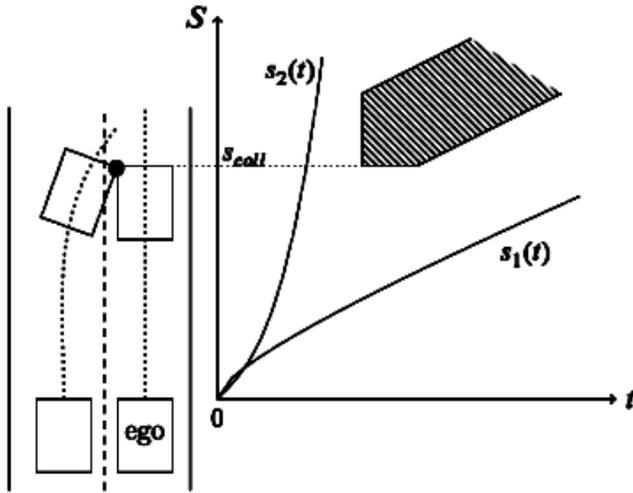


Рис. 6. Пример: другой агент (автомобиль) встраивается в полосу перед беспилотным автомобилем

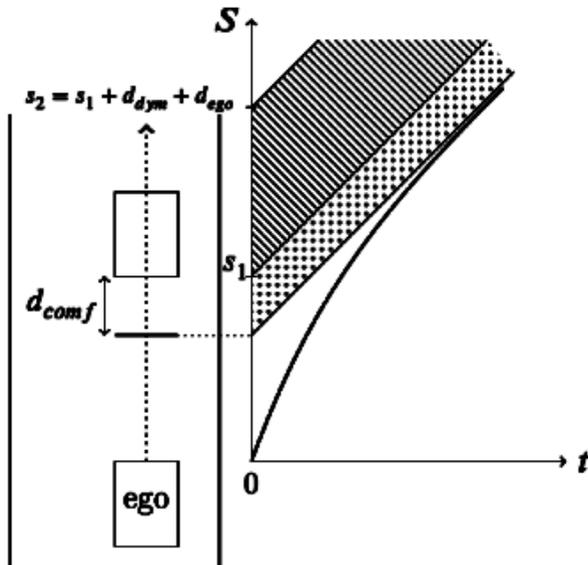


Рис. 7. Пример: соблюдение дистанции с едущим впереди агентом (автомобилем)

2.3. ЦЕЛЕВАЯ ФУНКЦИЯ ПОДХОДА DP

Будем рассматривать данный метод на примере пространства $S-V$. В этом пространстве отметим две оси: расстояние вдоль кривой S с возможными значениями $s \in [0, s_{max}]$ и ось V – возможная скорость движения со значениями в диапазон $v \in [0, v_{max}]$.

В рамках данного метода дискретизируем поисковое пространство. Выберем шаг дискретизации $\Delta s, \Delta v$ по осям S и V соответственно. Все точки, принадлежащие отрезку $[0, \Delta s)$, относятся к точке s'_0 на оси S' в дискретизованном пространстве $S'-V'$, т.е. $[\Delta s \cdot i, \Delta s \cdot (i + 1)) \rightarrow s'_i, i \in \mathbb{N}$. Аналогичным образом поступим с осью V : $[\Delta v \cdot j, \Delta v \cdot (j + 1)) \rightarrow v'_j, j \in \mathbb{N}$.

На рис. 8:

- p_1 с координатами $(2,25 \cdot \Delta s, 8,25 \cdot \Delta v)$ в пространстве $S-V$ переходит в (s'_2, v'_8) в $S'-V'$;
- p_2 с координатами $(2,25 \cdot \Delta s, 8,75 \cdot \Delta v)$ в пространстве $S-V$ тоже переходит в (s'_2, v'_8) в $S'-V'$;
- p_3 с координатами $(5,5 \cdot \Delta s, 4,5 \cdot \Delta v)$ в пространстве $S-V$ переходит в (s'_5, v'_4) в $S'-V'$.

Таким образом, текущее состояние системы описывается парой (s'_i, v'_j) . Из состояния (s'_i, v'_j) можно сделать переходы в состояния (s'_{i+1}, v'_k) , $k \in \mathbb{N}$, где скорость может отличаться в зависимости от величины ускорения, с которой осуществляется переход. Эти переходы таковы, что они не должны нарушать ограничения по боковым (a_{lat}) и продольным (a_{long}) ускорениями.

На рис. 8 показаны допустимые переходы из клеток (s'_2, v'_3) и из (s'_8, v'_3) .

Следуя по различным комбинациям переходов между состояниями, может получиться множество профилей скорости. Из них следует выбрать тот, который наилучшим образом удовлетворяет требованиям к целевой функции. Для этого:

1. Введем функцию стоимости перехода, $C((s'_{i1}, v'_{j1}), (s'_{i2}, v'_{j2}))$. В зависимости от требований к целевой функции, она будет определяться по-разному: например, если требуется оп-

тимизировать время поездки, это может быть время прохождения из состояния (s'_{i1}, v'_{j1}) в состояние (s'_{i2}, v'_{j2}) .



Рис. 8. Пример построения решения методом DP и пример допустимых переходов между состояниями в дискретизованном S - V -пространстве

2. Введем метрику оценки полученного профиля скорости. Для примера выше это может быть сумма стоимостей всех переходов.

3. Для конкретного состояния (s'_i, v'_j) оптимальная стоимость профиля, заканчивающегося в этом состоянии, будет удовлетворять уравнению

$$(10) dp(s'_i, v'_j) = \min_p (dp(s'_{i_p}, v'_{j_p}) + C((s'_{i_p}, v'_{j_p}), (s'_i, v'_j)))$$

$\forall p$, из которых есть переход в (s'_i, v'_j) .

2.4. ЦЕЛЕВАЯ ФУНКЦИЯ МЕТОДА QP

Метод квадратичного программирования (QP) – одна из классических задач оптимизации, и построение профиля скоро-

сти сводится к нему при некоторых допущениях. Чтобы произвести сведение, нужно:

1. Выбрать n переменных управления – вектор-столбец x размерности n .

2. Линеаризовать ограничения. В результате составляется матрица A размера $m \times n$, где m – количество ограничений, и m -мерный вектор-столбец b . Тогда решение x^* должно удовлетворять неравенству $Ax^* \leq b$.

3. Составить выпуклый квадратичный функционал. Составляется симметричная положительно полуопределенная матрица H размера $n \times n$ (гессиан) и градиент – вектор-столбец g размера n :

$$(11) \quad \frac{1}{2} x^T H x + g^T x .$$

Требуется минимизировать выражение, описанное в формуле (11), при ограничениях выше.

3. Обзор методов решения задачи планирования скорости

Сосредоточимся далее на описании методов на основе динамического программирования (DP) и квадратичной оптимизации (QP).

3.1. ПРИМЕР РЕАЛИЗАЦИИ DP

В работе [9] авторы описывают следующий метод сведения задачи к динамическому программированию:

Для начала строится *опорный* профиль скорости (Reference Speed Profile). Фактически это функция зависимости скорости от пройденного вдоль кривой расстояния $v(s)$. Для этого авторами вводятся несколько фильтров (называемых авторами «агентами»), каждый из которых накладывает ограничение:

1) на максимально осуществимую транспортным средством скорость;

2) связанное с соблюдением правил дорожного движения;

3) на скорость, связанное с боковыми ускорениями;

4) на скорость в зависимости от сигнала светофора.

Динамические препятствия (например, другие агенты) обрабатываются с помощью их проекции на ST-граф. Далее происходит дискретизация пространства расстояний S и времени T , по принципу, описанному в пункте 2.2.1. Узлы дискретизации формируют пространство вершин графа.

В получившемся пространственно-временном графе (Station-Time Graph, ST-graph) проводятся ребра между вершинами: $(s_1, t_1) \rightarrow (s_2, t_2)$; $s_1 \leq s_2$; t_2 является следующей точкой дискретизации T после t_1 . При этом нет строгого запрета на проведение ребер, проход по которым будет означать, что скорость транспортного средства нарушает все ограничения.

Такие параметры движения, как скорость, ускорение и рывок (jerk), хранятся в вершинах графа и определяются так:

- скорость v – как функция от s в текущей вершине и s в вершине-предке по ребру: $\frac{s_2 - s_1}{\delta t}$;
- ускорение a – как функция от v в текущей вершине и v в вершине-предке по ребру;
- рывок j – как функция от a в текущей вершине и a в вершине-предке по ребру.

Можно заметить, что фактически ускорение в вершине зависит от s трех последовательных вершин в пути, а рывок – от четырех.

Эти параметры учитываются в функции стоимости пути. Функция J_E является линейной комбинацией функций стоимостей за соблюдение скоростей опорного профиля скорости J_V , ограничений по продольному ускорению J_A , по продольному рывку J_J , соблюдение комфортного расстояния до препятствий J_O .

$$(12) J_E = C_V J_V + C_A J_A + C_J J_J + C_O J_O,$$

где C_V, C_A, C_J, C_O – это некоторые подобранные коэффициенты.

J_V штрафует разницу между скоростью v_{st} в вершине ST-графа и соответствующей ей скоростью v_{ref} из опорного профиля. Линейно, если $v_{st} < v_{ref}$, иначе – квадратично. Также учтен выход из физических ограничений транспортного средства – штраф в этом случае бесконечен.

- J_A штрафует тем сильнее, чем меньше скорость, если ускорение находится в «мягких» пределах, иначе – штраф экспоненциален. Как и в случае со скоростью, штраф бесконечен, если ускорение исполнить невозможно.

- J_J – рывок учитывается квадратично. Ограничений на физическую исполнимость нет.

- J_O – сумма по всем агентам. Для конкретного агента штраф бесконечен, если расстояние до другого агента меньше 0 (т.е. произошла коллизия), иначе обратно пропорционален расстоянию.

Заметим, что боковое ускорение и так было учтено в J_V ; к тому же в итоговой формуле не учтен боковой рывок.

Нетрудно доказать, что реализация модели может найти неоптимальное решение или даже не найти существующее решение вовсе, поскольку для конкретной вершины выбор оптимального ребра происходит среди вариантов вершин только с предыдущего по времени «слоя». Для учета всех вариантов требуется перебирать все возможные комбинации подсоединения до предка глубины 3 через предков глубины 1 и 2, поскольку в функции стоимости рывок – это функция четырех последовательных вершин в пути. Эта проблема становится тем менее значительной, чем меньше шаг дискретизации по S и T , так как разница по s и t между связанными ребрами вершинами становится меньше и, следовательно, возможное нарушение тоже уменьшается.

3.2. ПРИМЕР СВЕДЕНИЯ К QP ЧЕРЕЗ ДИСКРЕТИЗАЦИЮ ВРЕМЕНИ

В работе [1] авторы приводят один из возможных способов сведения к задаче квадратичного программирования – использование метода поиска в ширину (BFS) для поиска пути по ST-графу между агентами, внутри каждого шага BFS происходит решение задачи квадратичной оптимизации.

На оси времени T расставляются N опорных точек t_i , $1 \leq i \leq N$, где время будет зафиксировано (и равно $dt \times (i - 1)$ для заранее выбранного dt), а параметры рывка j , продольного ускорения a , скорости v и пройденного расстояния s (p в самой ста-

ть) являются искомыми. Переменные управления здесь j_i , все остальные параметры несложным образом из них выводятся:

$$\begin{aligned} s_{i+1} &= s_i + v_i \cdot dt, \\ (13) \quad v_{i+1} &= s_i + a_i \cdot dt, \\ a_{i+1} &= s_i + j_i \cdot dt. \end{aligned}$$

Ограничения для избежания коллизий формулируются следующим образом:

$$(14) \quad s_i^{lb} \leq s_i \leq s_i^{ub},$$

где s_i^{lb} и s_i^{ub} берутся напрямую из ST-графа как ближайшие от s_i границы агентов снизу (s_i^{lb}) и сверху (s_i^{ub}) соответственно.

Более того, для удержания комфортного расстояния до агентов вводятся slack-переменные [2]. Каждой точке t_i соответствует одна переменная для агента спереди и одна – для агента сзади:

$$(15) \quad s_i^{lb} + slack_i^{lb} \leq s_i \leq s_i^{ub} - slack_i^{ub}$$

На эти переменные также накладываются дополнительные условия $slack_i^{lb} \geq 0$, $slack_i^{ub} \geq 0$ для сохранения безопасного расстояния до объектов. Для того чтобы эти переменные «отталкивали» от агента, требуется их учесть в целевой функции.

В такую модель отлично ложатся ограничения на продольные ускорения и рывки, а также на скорости, если известны их точные зависимости от времени, т.е. заданы $v(t)$, $a_{long}(t)$ и $j_{long}(t)$ либо ограничения на них константны. В таком случае они выглядят аналогично формуле (14):

$$\begin{aligned} v_i^{lb} &\leq v_i \leq v_i^{ub}, \\ (16) \quad a_i^{lb} &\leq a_i \leq a_i^{ub}, \\ j_i^{lb} &\leq j_i \leq j_i^{ub}. \end{aligned}$$

Несложно показать, что в формуле (13) все a_i , v_i , s_i можно выразить через переменную управления j_i . Можно заметить, что данное сведение неточно, так как, например, между t_i и t_{i+1} скорость и ускорение считаются константными.

Точность модели можно улучшить несложным образом:

$$s_{i+1} = s_i + v_i \cdot dt + a_i \cdot \frac{dt^2}{2} + j_i \cdot \frac{dt^3}{6},$$

$$(17) \quad v_{i+1} = v_i + a_i \cdot dt + j_i \cdot \frac{dt^2}{2},$$

$$a_{i+1} = a_i + j_i \cdot dt.$$

Для расширения функциональности в такую модель можно напрямую включить:

1. Учет комфорта пассажира. Для этого добавим в целевую функцию $w_j \Sigma(j_i)^2$, где w_j – положительный коэффициент. Также можно добавить $w_a \Sigma(a_i)^2$ для учета ускорений, где w_a положительно. Ускорения a_i достаточно выразить через рывки j_i по формуле выше. И рывки, и ускорения, таким образом, добавляются в гессиан H . Чем больше коэффициенты w_a и w_j , тем меньше ускорения и рывки в оптимальном решении и, следовательно, выше комфорт пассажира.

2. Максимизация пройденного расстояния. Для этого достаточно добавить $w_s s_{N-1}$ в градиент g , где w_s – отрицательный коэффициент.

3. Комфортное расстояние до агентов. Поскольку slack-переменные являются отдельными переменными управления, в целевую функцию они добавляются тоже отдельно. Их можно добавить как в гессиан $(w_{slack} \cdot \sum((slack_i^{lb})^2 + (slack_i^{ub})^2))$, так и в градиент $(w_{slack} \cdot \sum(slack_i^{lb} + slack_i^{ub}))$, где w_{slack} – вес этой компоненты.

Проблемой данной модели является то, что зависимость параметров от пройденного расстояния невозможно вставить в модель в качестве линейного ограничения. Например, это может касаться ограничений по скорости на пешеходных переходах: они зафиксированы в определенных точках пройденного пути, но нет прямой зависимости скорости в этих точках от времени.

То же касается и ограничений на боковые ускорения, поскольку они напрямую зависят не от времени, а от кривизны геометрической траектории в каждой её конкретной точке. Эту же проблему отмечают Alexandre Miranda Aïon и коллеги в ра-

боте [1]. Они используют «funnel technique» – несложную эвристику для аппроксимации ограничений по скорости от расстояния. Возможным решением может стать разложение таких ограничений в ряд Тейлора по принципу, похожему на использующийся в работе [16].

3.3. ПРИМЕР СВЕДЕНИЯ К QP ЧЕРЕЗ ДИСКРЕТИЗАЦИЮ РАССТОЯНИЯ

В работе [11] авторы Changliu Liu и др. производят сведение задачи построения профиля скорости к задаче квадратичного программирования.

В данном случае вдоль оси S расставляются $N + 1$ опорная точка s_i , $0 \leq i \leq N$, в которых будет зафиксировано пройденное расстояние. Изначально переменными управления являются скорость $v(s_i)$, продольное ускорение $a(s_i)$ и $\delta_t(s_i)$ – разности во времени достижения точек s_i и s_{i+1} . Время достижения точки s_i можно представить как $\sum_{j=0}^{i-1} \delta_t(s_j)$.

Ограничения устроены следующим образом:

1. $0 \leq v(s_i) \leq \bar{v}(s_i)$, $v(s_0) = v_0$, $v(s_N) = v_N$, где $\bar{v}(s_i)$ – ограничение по скорости из ПДД и боковых ускорений. Поскольку они зависят от геометрической траектории, данные ограничения вставляются в модель напрямую. Ограничение на последнюю точку можно опустить.

2. $\underline{a}(s_i) \leq a(s_i) \leq \bar{a}(s_i)$, $a(s_0) = a_0$, $a(s_N) = a_N$ – ограничения по ускорению. Ограничения на последнюю точку в этом случае тоже можно опустить.

3. $\underline{t}(s_i) \leq t_i \leq \bar{t}(s_i)$, где $\underline{t}(s_i)$ и $\bar{t}(s_i)$ – верхние и нижние ограничения на ST-коробки, вычисляемые из ST-графа.

4. $v(s_i)\delta_t(s_i) = \Delta s$. Поскольку это ограничение не является линейным, оно линеаризуется с использованием разложения в ряд Тейлора по переменным $v(s_i)$ и $\delta_t(s_i)$ в точке (v_*, δ_*) . Также добавляются slack-переменные для улучшения сходимости. Итоговое выражение выглядит так:

$$(18) \quad \delta_* v(s_i) + v_* \delta_t(s_i) - v_* \delta_* = \Delta s + slack_v(s_i)$$

5. $a(s_i)\delta_t(s_i) = v(s_{i+1}) - v(s_i)$. Здесь применяется тот же прием с разложением в ряд Тейлора:

$$(19) \delta_* a(s_i) + a_* \delta_i(s_i) - a_* \delta_* = v(s_{i+1}) - v(s_i) + \text{slack}_a(s_i).$$

Данная задача решается в несколько итераций, пока решение не окажется достаточно близко к оптимуму. Значения a_* , v_* , δ_* берутся из предыдущей итерации решения квадратичной задачи (или из опорного профиля, если это первая итерация). Это требуется, поскольку ограничения в их оригинальном виде приближенные.

Целевая функция представляет из себя линейную комбинацию $C_T T + C_R R(s) + C_D D(s) + C_K K(s)$ заранее заданных коэффициентов C_T , C_R , C_D , C_K со следующими функциями:

1. $T = t(s_N) = \sum_{j=0}^{N-1} \delta_i(s_j)$ – время достижения финальной точки s_N , которое будет минимизировано за счет добавления в линейную часть квадратичного функционала (в вектор g – градиент).

2. $R(s) = \sum_{i=0}^{N-1} (a(s_{i+1}) - a(s_i))^2$ – добавляется в гессиан квадратичной задачи для обеспечения гладкости профиля.

Можно заметить, что данная компонента минимизирует сумму дискретных производных не по времени (рывки), а по расстоянию (так называемые «псевдо-рывки»). Следствием является то, что на малых скоростях рывок штрафует сильнее, чем на больших, так как Δs преодолевается за большее время, а значит, с большим изменением ускорения. Это может быть как положительным, так и отрицательным аспектом.

3. $D(s) = \sum_{i=0}^{N-1} (t(s_i) - t_d(s_i))^2$, где $t_d(s_i)$ – рекомендуемое расстояние до агента спереди. Добавляется в гессиан.

Согласно данной компоненте, выгодно жестко следовать выбранной дистанции, что является минусом данного подхода. Это может сказываться в ситуации, когда агент находится на большом расстоянии – такой механизм будет излишне «подгонять» БТС, даже если в этом нет необходимости. Возможно, подход из [1] через софт-ограничения и slack-переменные окажется эффективнее.

4. $K(s) = \sum_{i=0}^{N-1} (\text{slack}_v^2(s_i) + \text{slack}_a^2(s_i))$ – добавляется в гессиан для минимизации slack-переменных.

Преимуществом подхода является поддержка как ограничений на скорость, так и на удержание от агентов «из коробки» с быстрым действием (согласно результатам экспериментов в самой работе), позволяющим использовать данный метод на системах реального времени. Минусом является отсутствие поддержки ограничений на продольные и поперечные рыбки.

Недостатком подхода является невыпуклость модели, поскольку присутствует гиперболическое уравнение для связи пути и скоростей, представленное формулой (19).

4. Предложения по использованию

4.1. ОБЛАСТЬ ПРИМЕНЕНИЯ МЕТОДОВ

Описанные методы и подходы можно использовать как по отдельности, так и в комбинации, в зависимости от поставленной задачи.

Если в задаче имеются невыпуклые ограничения, и они ярко выражены, то метод DP использовать может быть эффективнее, так как аппроксимация данных ограничений будет качественнее. Реализация метода QR при этом не всегда возможна, в силу сложности сведения всех компонент (т.е. целевой функции и ограничений) к единой QR-задаче. Хотя типичным выходом из этой ситуации является разбиение решения на серию задач, последовательное решение которых сходится к оптимуму (SQP – sequential quadratic programming), оно далеко не всегда эффективно или возможно.

Метод DP может использоваться для построения опорного профиля скорости. Само по себе это решение не подходит для непосредственного исполнения, так как является «грубой» оценкой и требует дополнительного сглаживания. Более того, на реальных двигателях такое управление может приводить к дополнительному износу деталей или быть попросту невыполнимым.

Данное опорное решение далее можно использовать для инициализации следующего метода. Например, используя метод QR, можно добиться гладкого варианта профиля скорости, относительно первоначально полученного из DP.

Если в задаче необходимо управление производными $s(t)$ высокого порядка – от третьей производной (продольных рывков) и далее, и при этом требуется высокая точность при решении задачи, то использование метода DP может быть проблематичным. Причиной является так называемое «проклятие размерности» («curse of dimensionality»), которое заключается в том, что метод плохо масштабируется при увеличении количества измерений поискового пространства и при уменьшении шага дискретизации.

4.2. МЕТОД DP В КОНТЕКСТЕ ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ

Отличительной особенностью метода DP является возможность применения для нахождения допустимых альтернатив; в частности, если в задаче присутствуют другие агенты и заранее не известно, требуется ли обогнать или следовать за каждым из них.

Примером задачи с принятием решений является ситуация на рис. 6. В данном случае динамическое программирование позволяет смоделировать все варианты: обогнать агента или уступить дорогу, и выбрать подходящий. В проекции $S-V$ этого добиться трудно, но просто в проекции $S-T$ (или $S-V-T$).

Еще одна ситуация: БТС оказывается вблизи светофора в момент, когда начинает мигать зеленый сигнал и загорается желтый. В этот момент необходимо оценить, требуется ли продолжать ехать, возможно увеличив скорость для того, чтобы успеть пересечь пешеходный переход, либо наоборот, затормозить и остановиться, пропустив пешеходов. Пример такой ситуации приведен на рис. 9: при нахождении в отмеченной штрихом зоне будут нарушены ПДД. Принятие решения сводится к выбору между альтернативами вида $s_1(t)$ и $s_2(t)$.

Во всех ситуациях, изложенных выше, с помощью DP можно эффективно обрабатывать все исходы и осуществлять выбор из них без необходимости перезапускать решатель несколько раз для обработки каждого из вариантов по отдельности. Для этого требуется разобраться, как именно текущий сценарий влияет на пространство состояний. Задача является инженерной,

в каждом конкретном сценарии требуется отдельная проработка, но во всех описанных ранее примерах задачу можно решить путем ввода дополнительных ограничений на ST-графе.

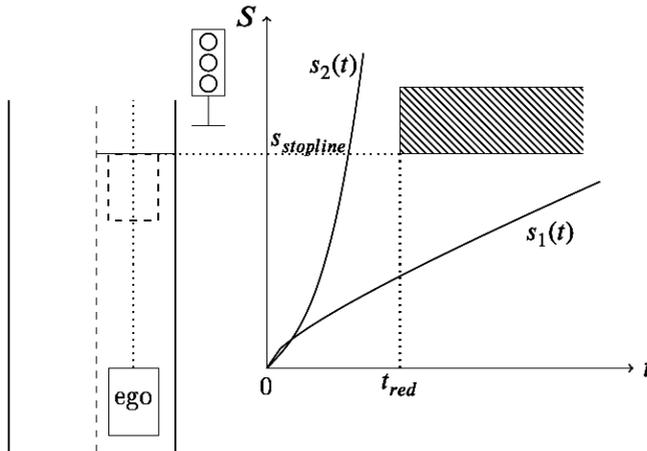


Рис. 9. Ограничения на ST-графе от загорающего запрещающего сигнала светофора. T_{red} обозначает момент времени, в который загорится красный сигнал

Таким образом, запустив метод один раз, удастся получить сразу все множество допустимых решений, из которых далее уже может быть осуществлен выбор.

5. Выводы

Рассмотренные в работе подходы планирования профиля скорости с помощью динамического программирования и квадратичной оптимизации являются актуальными и перспективными для дальнейших улучшений и научных изысканий. Поскольку коммерчески успешных проектов, использующих наземные БТС, пока нет, известны не все требуемые параметры задачи, необходимые для создания решения, обладающего достаточным для массового коммерческого использования качеством.

В статье были подробно разобраны сценарии работы методов. Методы находят свое применения на реальных задачах, сценарии применения вариативны. Авторами показаны преимущества и недостатки в конкретных ситуациях.

Авторами отмечено перспективное направление использования метода динамического программирования в рамках задачи принятия решения. Подход позволяет экономить вычислительные мощности при генерации допустимых альтернатив.

В дальнейшем авторы собираются исследовать возможность составления полностью выпуклой модели для решения задачи построения профиля скорости методом QP, а также исследовать границы применимости метода DP в пространстве S–V–T.

Литература

1. AÑON A.M. et al. *Multi-profile quadratic programming (MPQP) for optimal gap selection and speed planning of autonomous driving* // IEEE Int. Conf. on Robotics and Automation (ICRA–2024). – IEEE, 2024. – P. 12158–12164.
2. BOYD S.P., VANDENBERGHE L. *Convex optimization*. – Cambridge University Press, 2004.
3. DIACHUK M., EASA S.M. *Simultaneous Trajectory and Speed Planning for Autonomous Vehicles Considering Maneuver Variants* // Applied Sciences. – 2024. – Vol. 14, No. 4. – P. 1579.
4. DU Z. et al. *Speed profile optimisation for intelligent vehicles in dynamic traffic scenarios* // Int. Journal of Systems Science. – 2020. – Vol. 51, No. 12. – P. 2167–2180.
5. ELBANHAWI M., SIMIC M., JAZAR R. *In the passenger seat: investigating ride comfort measures in autonomous cars* // IEEE Intelligent Transportation Systems Magazine. – 2015. – Vol. 7, No. 3. – P. 4–17.
6. *Expanding Waymo's testing to the city that keeps it weird* [Электронный ресурс]. – URL: <https://waymo.com/blog/2023/03/expanding-waymos-testing-to-austin/> (дата обращения: 16.03.2025)

7. FU D. et al. *Drive like a human: Rethinking autonomous driving with large language models* // IEEE/CVF Winter Conf. on Applications of Computer Vision Workshops (WACVW–2024). – IEEE, 2024. – P. 910–919.
8. KLANČAR G. et al. *Drivable path planning using hybrid search algorithm based on E* and Bernstein–Bézier motion primitives* // IEEE Trans. on Systems, Man, and Cybernetics: Systems. – 2019. – Vol. 51, No. 8. – P. 4868–4882.
9. KRÜGER T.J., GÖHRING D., ULBRICH F. *Graph-Based Speed Planning for Autonomous Driving*. – PhD thesis, 2019.
10. LEE D. et al. *Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC* // IEEE 20th Int. Conf. on Intelligent Transportation Systems (ITSC–2017). – IEEE, 2017. – P. 1–6.
11. LIU C., ZHAN W., TOMIZUKA M. *Speed profile planning in dynamic environments via temporal optimization* // IEEE Intelligent Vehicles Symposium (IV). – IEEE, 2017. – P. 154–159.
12. MARTIN D., LITWHILER D. *An investigation of acceleration and jerk profiles of public transportation vehicles* // Annual Conference & Exposition. – 2008. – P. 13.194.1–13.194.13.
13. MOZAFFARI S. et al. *Deep learning-based vehicle behavior prediction for autonomous driving applications: A review* // IEEE Trans. on Intelligent Transportation Systems. – 2020. – Vol. 23, No. 1. – P. 33–47.
14. SHIMIZU Y. et al. *Jerk constrained velocity planning for an autonomous vehicle: Linear programming approach* // Int. Conf. on Robotics and Automation (ICRA–2022). – IEEE, 2022. – P. 5814–5820.
15. SINGH K.B., SIVARAMAKRISHNAN S. *Extended pacejka tire model for enhanced vehicle stability control* // arXiv preprint: arXiv:2305.18422. – 2023.
16. XU W. *Motion planning for autonomous vehicles in urban scenarios: a sequential optimization approach* : PhD thesis. – Carnegie Mellon University, 2021.
17. YANG G., YAO Y. *Vehicle local path planning and time consistency of unmanned driving system based on convolutional*

- neural network* // Neural Computing and Applications. – 2022. – P. 1–14.
18. YEO J., LEE J., JANG K. *The effects of rainfall on driving behaviors based on driving volatility* // Int. Journal of Sustainable Transportation. – 2021. – Vol. 15, No. 6. – P. 435–443.
 19. ZHANG J., JIN H. *Optimized calculation of the economic speed profile for slope driving: Based on iterative dynamic programming* // IEEE Trans. on Intelligent Transportation Systems. – 2020. – Vol. 23, No. 4. – P. 3313–3323.
 20. ZHANG Y. et al. *Optimal vehicle path planning using quadratic optimization for baidu apollo open platform* // IEEE Intelligent Vehicles Symposium (IV). – IEEE, 2020. – P. 978–984.
 21. ZHAO F. et al. *On-Road Trajectory Planning of Connected and Automated Vehicles in Complex Traffic Settings: A Hierarchical Framework of Trajectory Refinement* // IEEE Access. – 2024. – Vol. 12. – P. 7456–7468.
 22. ZHAO T. et al. *Multi-agent tensor fusion for contextual trajectory prediction* // Proc. of the IEEE/CVF conference on computer vision and pattern recognition. – 2019. – P. 12126–12134.

APPROACHES TO SPEED PLANNING FOR GROUND-BASED AUTONOMOUS VEHICLES

Asya Livshits, NUST MISIS, Moscow, B.Sc. (asyalivshits@yandex.ru).

Igor Temkin, NUST MISIS, Moscow, Doctor of Technical Science, Professor (temkin.io@misis.ru).

Aleksandr Fadeev, NUST MISIS, Moscow, Postgraduate Student (frogcatcher@mail.ru).

Abstract: The article provides an overview of speed profile planning methods for ground-based unmanned vehicles. Autonomous driving technology is in the active development phase, and many tasks have already been solved, but with insufficient quality for safe operation and scaling of the technology. Thus, this field is promising from a scientific and engineering point of view. The paper examines in detail the formulation of the problem, describes various types of constraints to the final solution, and considers the construction of the task's functionality, depending on the requirements and methods used. Special attention is paid to the ST graph as a tool for modeling interaction with other traffic participants. Practical scenarios of agent interaction in various situations that often arise in traffic are also analyzed in de-

tail. The authors focus on a class of methods for constructing a velocity profile along a known geometric trajectory. Two families of approaches are considered: dynamic programming and reducing a continuous problem to a quadratic programming problem. The existing modern methods based on these approaches are analyzed in detail, and improvements and improvements are proposed in the context of various additional requirements for the solution. The article serves as a methodological basis for developers of autonomous driving systems.

Keywords: unmanned transport systems, speed profile, dynamic programming, continuous optimization, ST graph.

УДК 656.1

ББК 39.37 + 32.816-04

*Статья представлена к публикации
членом редакционной коллегии Л.Б. Рапопортом.*

Поступила в редакцию 30.03.2025.

Опубликована 31.07.2025.