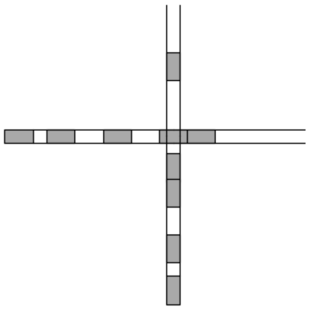

Figure 1: Illustration of a single intersection with vehicles drawn as grey rectangles. Vehicles approach the intersection from the east and from the south and cross it without turning. Note that the first two waiting vehicles on the south lane kept some distance before the intersection, such that they are able to reach full speed whenever they cross.

Single intersection

We are interested in algorithms to efficiently control the movement of vehicles through a network of intersections under safety constraints. In order to keep the problem simple, we assume that vehicle routes are fixed and known whenever a vehicle enters the network. This means that we can consider the longitudinal position $x_i(t)$ of each vehicle i along its route, for which we use the well-known *double integrator* model

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= u_i(t), \\ 0 \leq v_{\max} &\leq v_{\max}, \\ |u_i(t)| &\leq a_{\max}, \end{aligned} \tag{1}$$

where $v_i(t)$ is the vehicle's velocity and $u_i(t)$ its acceleration, which is set by the controller. Let $D_i(s_{i,0})$ denote the set of all trajectories $x_i(t)$ satisfying these dynamics, given some initial state $s_{i,0} = (x_i(0), v_i(0))$.

Before we consider networks of intersections, we first state the problem for a single intersection as illustrated in Figure 1. Assume there are two incoming lanes, identified by indices $\mathcal{R} = \{1, 2\}$. The corresponding two routes are crossing the intersection from south to north and crossing from west to east. We identify vehicles by their route and by their relative order on this route, by defining the vehicle index set

$$\mathcal{N} = \{(r, k) : k \in \{1, \dots, n_r\}, r \in \mathcal{R}\}, \tag{2}$$

where n_r denotes the number of vehicles following route r. Smaller values of k correspond to being closer to the intersection. Given vehicle index $i = (r, k) \in \mathcal{N}$, we also use the notation $r(i) = r$ and $k(i) = k$. We assume that each vehicle is represented as a rectangle of length L and width W and that $x_i(t)$ is measured at the front bumper. In order to maintain a safe distance between consecutive vehicle on the same lane, vehicle trajectories need to satisfy

$$x_i(t) - x_j(t) \geq L, \tag{3}$$

for all t and all pairs of indices $i, j \in \mathcal{N}$ such that $r(i) = r(j), k(i) + 1 = k(j)$. Let $\mathcal{C}$ denote the set of such ordered pairs of indices. Note that these constraints restrict vehicle from overtaking each other, so the initial relative order is always maintained. For each $i \in \mathcal{N}$, let $\mathcal{E}_i = (B_i, E_i)$ denote the open interval such that vehicle i occupies the intersection's conflict area if and only if $B_i < x_i(t) < E_i$. Using this notation, collision avoidance at the intersection is achieved by requiring

$$(x_i(t), x_j(t)) \notin \mathcal{E}_i \times \mathcal{E}_j, \tag{4}$$

for all t and for all pairs of indices $i, j \in \mathcal{N}$ with $r(i) \neq r(j)$, which we collect in the set $\mathcal{D}$. Suppose we have some performance criterion $J(x_i)$ that takes into account travel time and energy efficiency of the trajectory of vehicle i, then the offline trajectory optimization problem for a single intersection can be compactly written as

$$\begin{aligned}
\min_{\mathbf{x}(t)} \quad & \sum_{i \in \mathcal{N}} J(x_i) && && \text{(5a)} \\
\text{s.t.} \quad & x_i \in D_i(s_{i,0}), && \text{for all } i \in \mathcal{N}, && \text{(5b)} \\
& x_i(t) - x_j(t) \geq L, && \text{for all } (i,j) \in \mathcal{C}, && \text{(5c)} \\
& (x_i(t), x_j(t)) \notin \mathcal{E}_i \times \mathcal{E}_j, && \text{for all } \{i,j\} \in \mathcal{D}, && \text{(5d)}
\end{aligned}$$

where $\mathbf{x}(t) = [\, x_i(t) : i \in \mathcal{N} \,]$ and constraints are for all t.

Direct transcription

Although computationally demanding, problem (5) can be numerically solved by direct transcription to a non-convex mixed-integer linear program by discretization on a uniform time grid. Let K denote the number of discrete time steps and let Δt denote the time step size. Using the forward Euler integration scheme, we have

$$\begin{aligned}
x_i(t + \Delta t) &= x_i(t) + v_i(t)\Delta t, \\
v_i(t + \Delta t) &= v_i(t) + u_i(t)\Delta t,
\end{aligned}$$

for each $t \in \{0, \Delta t, \dots, K\Delta t\}$. Following the approach in [1], the collision-avoidance constraints between lanes can be formulated using the well-known big-M technique by the constraints

$$\begin{aligned}
x_i(t) &\leq B_i + \delta_i(t)M, \\
E_i - \gamma_i(t)M &\leq x_i(t), \\
\delta_i(t) + \delta_j(t) + \gamma_i(t) + \gamma_j(t) &\leq 3,
\end{aligned}$$

where $\delta_i(t), \gamma_i(t) \in \{0, 1\}$ for all $i \in \mathcal{N}$ and M is a sufficiently large number. Finally, the follow constraints can simply be added as

$$x_i(t) - x_j(t) \geq L,$$

for each $t \in \{0, \Delta t, \dots, K\Delta t\}$ and each pair of consecutive vehicles $(i, j) \in \mathcal{C}$ on the same lane. For example, consider the objective functional

$$J(x_i) = \int_{t=0}^{t_f} \left((v_d - v_i(t))^2 + u_i(t)^2 \right) dt,$$

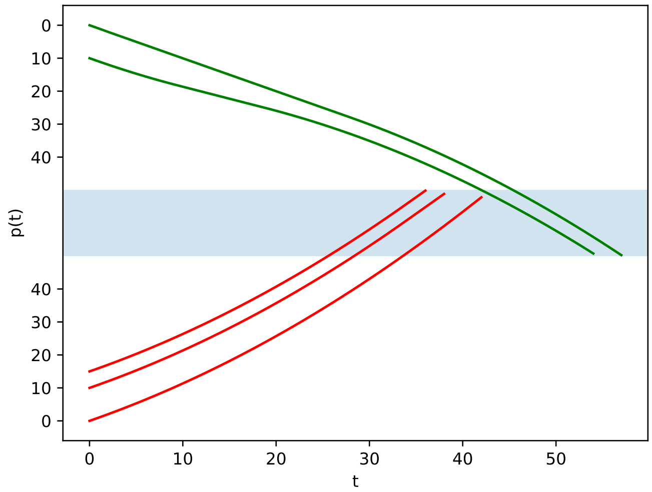

Figure 2: Example of optimal trajectories obtained using the direct transcription method with $L = 5$, $\mathcal{E}_i \equiv \mathcal{E} = [50, 70]$, $v_d = 20$, $T = 120$, $\Delta t = 0.1$ and initial conditions as given in Table 1. The y-axis is split such that each part corresponds to one of the two lanes and the trajectories are inverted accordingly and drawn with separate colors. The intersection area $\mathcal{E}$ is drawn as a shaded region. Whenever a vehicle has left the intersection, we stop drawing its trajectory for clarity.

where v_d is some reference velocity and t_f denotes the final time, then the optimal trajectories are shown in Figure 2.

i	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)
$x_i(0)$	15	10	0	10	0
$v_i(0)$	10	10	10	10	10

Table 1: Example initial conditions $s_{i,0} = (x_i(0), v_i(0))$ for problem (5).

Occupancy time slots

For the case where only a single vehicle is approaching the intersection for each route, so $n_r = 1$ for each route $r \in \mathcal{R}$, it has been shown that problem (5) can be decomposed into two coupled optimization problems, see Theorem 1 in [1]. Roughly speaking, the *upper-level problem* optimizes the time slots during which vehicles occupy the intersection, while the *lower-level problem* produces optimal safe trajectories that respect these time slots. When allowing multiple vehicles per lane, we show without proof that a similar decomposition is possible. Given $x_i(t)$, the *crossing time* of vehicle i, when the vehicle first enters the intersection, and the corresponding *exit time* are respectively

$$\inf\{t : x_i(t) \in \mathcal{E}_i\} \quad \text{and} \quad \sup\{t : x_i(t) \in \mathcal{E}_i\}. \tag{6}$$

The upper-level problem is to find a set of feasible occupancy timeslots, for which the lower-level problem generates trajectories. We will use decision variable

$y(i)$ for the crossing time and write $y(i) + \sigma(i)$ for the exit time. It turns out that trajectories can be generated separately for each route, which yields the decomposition

$$\min_{y,\sigma} \quad \sum_{r \in \mathcal{R}} F(y_r, \sigma_r) \tag{7a}$$

$$\text{s.t.} \quad y(i) + \sigma(i) \leq y(j) \text{ or } y(j) + \sigma(j) \leq y(i), \qquad \text{for all } (i,j) \in \mathcal{D}, \tag{7b}$$

$$(y_r, \sigma_r) \in \mathcal{S}_r, \qquad \text{for all } r \in \mathcal{R}, \tag{7c}$$

where $F(y_r, \sigma_r)$ and $\mathcal{S}_r$ are the value function and set of feasible parameters, respectively, of the lower-level *route trajectory optimization* problem

$$F(y_r, \sigma_r) = \min_{x_r} \quad \sum_{i \in \mathcal{N}(r)} J(x_i) \tag{8a}$$

$$\text{s.t.} \quad x_i \in D_i(s_{i,0}), \qquad \text{for all } i \in \mathcal{N}_r, \tag{8b}$$

$$x_i(y(i)) = B_i, \qquad \text{for all } i \in \mathcal{N}_r, \tag{8c}$$

$$x_i(y(i) + \sigma(i)) = E_i, \qquad \text{for all } i \in \mathcal{N}_r, \tag{8d}$$

$$x_i(t) - x_j(t) \geq L, \qquad \text{for all } (i,j) \in \mathcal{C} \cap \mathcal{N}_r, \tag{8e}$$

where we used $\mathcal{N}_r = \{i \in \mathcal{N} : r(i) = r\}$ and similarly for x_r, y_r and σ_r to group variables according to route. Note that the set of feasible parameters $\mathcal{S}_r$ implicitly depends on the initial states s_r and system parameters.

Delay objective

Assume that the trajectory performance criterion is exactly the crossing time, so $J(x_i) = \inf\{t : x_i(t) \in \mathcal{E}_i\}$. This assumption makes the problem significantly easier, because we have

$$F(y_r, \sigma_r) \equiv F(y_r) = \sum_{i \in \mathcal{N}_r} y(i). \tag{9}$$

Furthermore, we assume that vehicles enter the network and cross the intersection at full speed, so $v_i(0) = v_i(y(i)) = v_{\max}$, such that we have

$$\sigma(i) \equiv \sigma = (L + W)/v_{\max}, \quad \text{for all } i \in \mathcal{N}. \tag{10}$$

Therefore, we ignore the part related to σ in the set of feasible parameters $\mathcal{S}_r$, which can be shown that to have a particularly simple structure under these assumptions. Observe that $r_i = (B_i - x_i(0))/v_{\max}$ is the earliest time at which vehicle i can enter the intersection. Let $\rho = L/v_{\max}$ be such that $y(i) + \rho$ is the time at which the rear bumper of a crossing vehicle reaches the start line of the intersection, then it can be shown that $y_r \in \mathcal{S}_r$ whenever

$$r_i \leq y(i), \text{ for all } i \in \mathcal{N}_r, \tag{11a}$$

$$y(i) + \rho \leq y(j), \text{ for all } (i,j) \in \mathcal{C} \cap \mathcal{N}_r. \tag{11b}$$

Therefore, under the stated assumptions, problem (5) reduces to the following *crossing time scheduling* problem

$$\begin{aligned}
\min_{y} \quad & \sum_{i \in \mathcal{N}} y(i) && && \text{(12a)} \\
\text{s.t.} \quad & r_i \leq y(i), && \text{for all } i \in \mathcal{N}, && \text{(12b)} \\
& y(i) + \rho \leq y(j), && \text{for all } (i,j) \in \mathcal{C}, && \text{(12c)} \\
& y(i) + \sigma \leq y(j) \text{ or } y(j) + \sigma \leq y(i), && \text{for all } (i,j) \in \mathcal{D}, && \text{(12d)}
\end{aligned}$$

which can be solved using off-the-shelf mixed-integer linear program solvers, after encoding the *disjunctive constraints* (12d) using the big-M technique. Given optimal y^*, any set of trajectories $[x_i(t) : i \in \mathcal{N}]$ that satisfies

$$\begin{aligned}
x_i \in D_i(s_{i,0}), &\quad \text{for all } i \in \mathcal{N}, && \text{(13a)} \\
x_i(y^*(i)) = B_i, &\quad \text{for all } i \in \mathcal{N}, && \text{(13b)} \\
x_i(y^*(i) + \sigma) = E_i, &\quad \text{for all } i \in \mathcal{N}, && \text{(13c)} \\
x_i(t) - x_j(t) \geq L, &\quad \text{for all } (i,j) \in \mathcal{C}, && \text{(13d)}
\end{aligned}$$

forms a valid solution. We show how these trajectories can be computed by an efficient direct transcription method. First of all, note that each route may be considered separately and trajectories can be computed in a sequential fashion by repeatedly solving the optimal control problem

$$\begin{aligned}
\texttt{MotionSynthesize}(\tau, B, s_0, x') :=& \\
\arg\min_{x:[0,\tau]\to\mathbb{R}} & \int_0^\tau |x(t)| dt \\
\text{s.t. } & \ddot{x}(t) = u(t), && \text{for all } t \in [0,\tau], \\
& |u(t)| \leq a_{\max}, && \text{for all } t \in [0,\tau], \\
& 0 \leq \dot{x}(t) \leq v_{\max}, && \text{for all } t \in [0,\tau], \\
& x'(t) - x(t) \geq L, && \text{for all } t \in [0,\tau], \\
& (x(0), \dot{x}(0)) = s_0, \\
& (x(\tau), \dot{x}(\tau)) = (B, v_{\max}),
\end{aligned}$$

where τ is set to the required crossing time, B denotes the distance to the intersection, s_0 is the initial state of the vehicle and x' denotes the trajectory of the vehicle preceding the current vehicle.

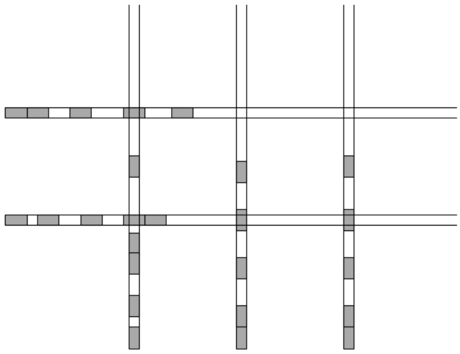

Figure 3: Illustration of some grid-like network of intersections with vehicles drawn as grey rectangles. There are five vehicle routes: two from east to west and three from south to north. Turning at intersections is not allowed.

Trajectories in networks

To extend the model to a network of intersections, illustrated in Figure 3, we need some additional notation to encode the network topology. We define a directed graph (V, E) with nodes V and arcs E, representing the possible paths that vehicles can follow. Nodes of in-degree at least two are called *intersections*. Nodes with only outgoing arcs are *entrypoints* and nodes with only incoming arcs are *exitpoints*. Let $d(v, w)$ denote the distance between nodes v and w. For each route index $r \in \mathcal{R}$, we let

$$\bar{V}_r = (v_r(0), v_r(1), \dots, v_r(m_r), v_r(m_{r+1})) \tag{14}$$

be the path that vehicles $i \in \mathcal{N}_r$ follow through the network. We require that the first node $v_r(0)$ is an entrypoint and that the last node $v_r(m_{r+1})$ is an exitpoint and we write

$$V_r = \bar{V}_r \setminus \{v_r(0),\, v_r(m_{r+1})\} \tag{15}$$

to denote the path restricted to intersections. We write $(v, w) \in V_r$ for some $(v, w) \in E$ whenever v and w are two consecutive nodes on the path. We require that routes can only overlap at nodes by making the following assumption.

Assumption 1. *Every arc $(v, w) \in E$ is part of at most one route V_r.*

Assume that physical lanes are axis-aligned and that routes are such that no turning is required. For each $v \in V_r$ define the conflict zone $\mathcal{E}_r(v) = (B_r(v), E_r(v))$ and consider the union

$$\mathcal{E}_r = \bigcup_{v \in V_r} \mathcal{E}_r(v) \tag{16}$$

corresponding to the positions of vehicles $i \in \mathcal{N}_r$ for which it occupies an intersection on its path V_r. By reading $\mathcal{E}_i \equiv \mathcal{E}_r$ for $r(i) = r$, the single intersection problem (5) naturally extends to the network case. Like before, the resulting problem can be numerically solved by a direct transcription method. The natural extension of the two-stage decomposition is to solve the occupancy time scheduling problem

$$\begin{aligned}
&\min_{y,\sigma} \quad \sum_{r\in\mathcal{R}} F(y_r,\sigma_r) && && (17a)\\
&\text{s.t.}\ \ y(i,v)+\sigma(i,v)\leq y(j,v) \text{ or} && && (17b)\\
&\qquad y(j,v)+\sigma(j,v)\leq y(i,v), && \text{for all } \{i,j\}\in\mathcal{D}^v \text{ and } v\in V, && (17c)\\
&\qquad (y_r,\sigma_r)\in\mathcal{S}_r, && \text{for all } r\in\mathcal{R}, && (17d)
\end{aligned}$$

where the set of conflict pairs at node v is defined as

$$\mathcal{D}^v = \{\{i,j\}\subset\mathcal{N} : r(i)\neq r(j), v\in V_{r(i)}\cap V_{r(j)}\}, \tag{18}$$

where $F(y_r,\sigma_r)$ and $\mathcal{S}_r$ are the value function and set of feasible parameters, respectively, of the parametric trajectory optimization problems

$$\begin{aligned}
F(y_r,\sigma_r) = &\min_{x_r} \sum_{r\in\mathcal{R}} J(x_i) && && (19a)\\
&\text{s.t.}\ \ x_i(t)\in D_i(s_{i,0}), && \text{for } i\in\mathcal{N}_r, && (19b)\\
&\qquad x_i(y(i,v)) = B_r(v), && \text{for } v\in V_r, i\in\mathcal{N}_r, && (19c)\\
&\qquad x_i(y(i,v)+\sigma(i,v)) = E_r(v), && \text{for } v\in V_r, i\in\mathcal{N}_r, && (19d)\\
&\qquad x_i(t)-x_j(t)\geq L, && \text{for } (i,j)\in\mathcal{C}\cap\mathcal{N}_r. && (19e)
\end{aligned}$$

Delay objective

Suppose we use use the crossing at the last intersection as performance measure, by defining the objective function as

$$J(x_i) = \inf\{t : x_i(t)\in\mathcal{E}_r(v_r(m_r))\}. \tag{20}$$

We show how to reduce the resulting problem to a scheduling problem, like we did in the single intersection case. Again, we require vehicles to drive at full speed whenever they occupy any intersection such that $\sigma_i\equiv\sigma=(L+W)/v_{\max}$. For each $r\in\mathcal{R}$, we provide a sufficient set of constraints on the crossing times

$$y(i,v) = \inf\{t : x_i(t)\in\mathcal{E}_r(v)\} \tag{21}$$

such that the corresponding lower-level problems are feasible, which we write as $y_r\in\mathcal{S}_r$, so we let $r\in\mathcal{R}$ be an arbitrary route in the following discussion. Like for the single intersection, let $\rho=L/v_{\max}$ denote the minimum time between two crossing times of vehicles of the same class. First of all, vehicles of the same class are required to keep safe distance, so we need

$$y(i,v)+\rho\leq y(j,v), \quad \text{for all } (i,j)\in\mathcal{C} \text{ and } v\in V. \tag{22}$$

Assuming that the initial velocity satisfies $v_i(0)=v_{\max}$, we require

$$r_i = (B_{r(i)}(v_r(l)) - x_i(0))/v_{\max} \leq y(i, v_r(l)), \tag{23}$$

where l is the smallest integer such that $x_i(0)\leq B_{r(i)}(v_r(l))$. For every $(v,w)\in V_r$, we require the *travel constraint*

$$y(i,v)+d(v,w)/v_{\max}\leq y(i,w) \tag{24}$$

for every $i \in \mathcal{N}_r$ to model the minimum time necessary for a vehicle to move from v to w. The minimum time required for a vehicle to come to a full stop or, equivalently, to accelerate to full speed from a stop, is given by

$$T = \frac{v_{\max}}{a_{\max}}.$$

The trajectory of full acceleration is given by

$$x_{\text{full}}(t) = \frac{t^2 a_{\max}}{2},$$

so the minimum distance required for a vehicle to fully accelerate or decelerate is given by

$$x_{\text{full}}(T) = \frac{v_{\max}^2}{2a_{\max}}.$$

By requiring that vehicles drive at full speed as long as they occupy an intersection, a vehicle crossing some intersection $v \in V_r$ can only start decelerating after $x_i(t) \geq B_r(v) + L + W$. Suppose that we want to design the network such that the lane segment (v, w) has capacity for at least $c(v, w)$ stationary vehicles, then we must have

$$d(v, w) \geq L + W + 2x_{\text{full}}(T) + (c(v, w) - 1)L.$$

Conversely, given lane length $d(v, w)$, this bound allows us to compute the maximum capacity as

$$c(v, w) = \texttt{floor}\left(\frac{d(v, w) - W - 2x_{\text{full}}(\hat{t})}{L}\right). \tag{25}$$

In order to guarantee that crossing time schedule y allows feasible trajectories, we need to define the *buffer constraints*

$$y(i, w) - \frac{d(v, w)}{v_{\max}} + c(v, w)\rho \leq y(j, v), \tag{26}$$

for every vehicle $i, j \in \mathcal{N}_r$ with $k(i) + c(v, w) = k(j)$ and $(v, w) \in V_r$. Figure 4 provides a sketch of the intuition behind the shape of the buffer constraints.

Proposition 1. *When y_r satisfies conjunctive constraints* (22)*, release constraints* (23)*, travel constraints* (24) *and buffer constraints* (26)*, then $y_r \in \mathcal{S}_r$.*

Remark 1. *It might be possible to relax the buffer constraints, i.e., there are situations in which vehicle j is able to cross v earlier.*

Given some crossing time schedule y for vehicles $\mathcal{N}_r$, we show how to compute trajectories by extending the `MotionSynthesize` procedure to multiple *checkpoints*

$$\zeta = ((\tau_1, B_1), \ldots, (\tau_m, B_m)),$$

where τ_n is a crossing time and B_n is the beginning of the nth intersection, measured along the path V_r relative to $v_r(0)$. Given the trajectory x' of the vehicle

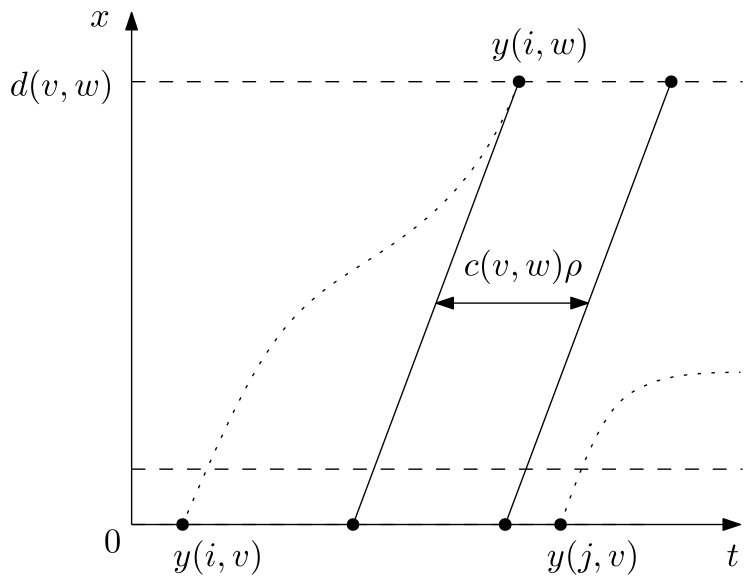

Figure 4: Illustration of the buffer constraint between $y(i,w)$ and $y(j,v)$. The slope of the two parallel lines is $v_{\max}$. The two dotted lines are examples of trajectories $x_i(t)$ and $x_j(t)$. The lower dashed line indicates $E_r(v)$ and the upper dashed line indicates $B_r(w)$.

ahead, the trajectory for a vehicle with checkpoint sequence ζ is computed by the procedure

$$
\begin{aligned}
&\texttt{CheckpointTrajectory}(\zeta, x') := \\
&\qquad \arg\min_{x:[\tau_1,\tau_m]\to\mathbb{R}} \int_{\tau_0}^{\tau_m} |x(t)|dt \\
&\qquad \text{subject to} \quad \ddot{x}(t) = u(t), && \text{for all } t \in [0,\tau_m], \\
&\qquad\qquad 0 \le \dot{x}(t) \le v_{\max}, && \text{for all } t \in [0,\tau_m], \\
&\qquad\qquad |u(t)| \le a_{\max}, && \text{for all } t \in [0,\tau_m], \\
&\qquad\qquad x'(t) - x(t) \ge L, && \text{for all } t \in [0,\tau_m], \\
&\qquad\qquad x(\tau_n) = B_n, && \text{for all } n = 1,\ldots,m, \\
&\qquad\qquad \dot{x}(\tau_n) = v_{\max}, && \text{for all } n = 1,\ldots,m.
\end{aligned}
$$

References

[1] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *2015 American Control Conference (ACC)*, (Chicago, IL, USA), pp. 763–768, IEEE, July 2015.

[2] D. Miculescu and S. Karaman, "Polling-systems-based Autonomous Vehicle Coordination in Traffic Intersections with No Traffic Signals," July 2016.