This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

Optimal conflict resolution for vehicles with intersecting and overlapping paths

Johan Karlsson, Nikolce Murgovski and Jonas Sjöberg

***Abstract*—A collaborative centralized model predictive controller solving the problem of autonomous vehicles safely crossing an intersection is presented. The solution gives optimal speed trajectories for each vehicle while considering collision avoidance constraints between vehicles traveling on the same path before, after and/or within the intersection. This extends earlier results, where collision avoidance was only considered for vehicles with intersecting paths, with the possibility of vehicles on the same path and by this, the controller is not only one step closer to handling complex traffic intersections but can now be used for merging and splitting of roads, roundabouts and intersection networks. The proficiency of the extended controller is demonstrated by applying it to a four-way intersection. It is shown that the controller provides smooth, collision free trajectories in scenarios with and without vehicles traveling in the same lane. Further, it is evaluated how the solutions differ when using various cost functions and how the controller handles disturbances in the form of a sudden lane blockage. Lastly, it is discussed how the presented controller could also be extended to handle mixed-traffic scenarios and how soft constraints can be used to avoid infeasibility in the case of missing or noisy traffic data.**

I. Introduction

The development of Intelligent transportation systems (ITS) has been a major subject for the automotive industry as well as governmental institutions in recent years, due to its potential for safer, smarter and greener solutions [1]. Autonomous driving is one of the areas undergoing extensive research within ITS.

One of the most extensively researched applications of autonomous driving is the intersection problem, [2], [3], [4]. The intersection problem is the problem of determining the crossing order and collision free trajectories of a number of autonomous vehicles traveling through an intersection.

Among the main motivations for studying the intersection problem are that they are prone to congestion and accidents. For example, in Europe, intersection-related accidents are responsible for 21% of traffic related deaths and 43% of non-fatal injuries [5]. Similar numbers have been reported for the U.S. [6]. Due to this accident risk, intersections are highly regulated by traffic lights, signs and road markings, which increases the risk for congestion. Autonomous vehicles have been suggested as one way of reducing the accident risk and congestion, simultaneously.

Many solution techniques have been suggested for controlling autonomous vehicles in intersections including scheduling formulations [7], [8], graph-based approaches [9], multi-agent approaches [10], and model predictive controllers (MPC); both centralized [11], [12], [13] and decentralized [14], [15], [16].

To solve the intersection problem using MPC, the problem is modeled as an optimization program, where constraints enforce collision avoidance, as well as state and control bounds. This optimization program, called an optimal control program, is solved over a prediction horizon to determine the future state and control signals of the vehicles. However, to account for deviations in the traffic environment from the predicted one, on which the optimization is based, it is necessary to regularly take new traffic information into account. This is done by repeatedly solving the optimization problem over a moving prediction horizon, containing updated traffic information [17].

One difficulty of solving the intersection problem is that it is a combinatorial problem when the crossing order of the vehicles is unknown. This leads to a computationally expensive MPC. To handle this, it is common to split the intersection problem into two parts; one combinatorial and one non-combinatorial.

The non-combinatorial part consists of determining the optimal trajectories, including longitudinal speed and acceleration, of each vehicle crossing the intersection, under the assumption that it is known in which order the vehicles cross and that each vehicle is traveling along a fixed path inside the intersection. There are several suggested algorithms for solving the non-combinatorial part of the intersection problem. The main difference between them is how the collision avoidance constraints are handled. In [11], [12] collision avoidance is modeled by introducing constraints that prohibit more than one vehicle to occupy the intersection at the same time. This is suboptimal since it excludes solutions where vehicles with non-intersecting paths cross simultaneously. Less conservative but more complicated collision avoidance constraints can be implemented by either splitting the physical area into quadrants [18] or by introducing local critical zones centered around the points where the predefined paths of vehicles intersect [13]. Another difference between the MPC algorithms is if the sampling is done in traveling time or distance. The main advantages of sampling in distance is that all samples at which collision is possible is known a-priori and the simplicity of minimizing time by introducing traveling time as an optimization variable. The resulting optimization problems are usually quadratic problems (QP) or nonlinear QPs, both

Manuscript received ...; This work was partially supported by the Wallenberg AI, Autonomous Systems and software Program (WASP) and partially by the European Commission Seventh Framework Program under the project AdaptIVe, grant agreement number 610428. Nikolce Murgovski acknowledges the support of the Transport Area of Advance within Chalmers University of Technology, Sweden

J. Karlsson, N. Murgovski and J. Sjöberg are with Chalmers university, Gothenburg, Sweden (email: jokarls@chalmers.se; nikolce.murgovski@chalmers.se; jonas.sjoberg@chalmers.se)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

NOMENCLATURE

List of Symbols

L	Number of intersection legs.
M	Number of predefined paths.
N	Number of vehicles.
x, y	Longitudinal and lateral coordinate.
v	Speed.
a	Acceleration.
v^{sl}	Speed limit of the road.
s	Arc length of predefined path.
p	Arc length position relative vehicle starting point.
ψ	Path angle.
t	Travel time.
t^{hw}_{nm}	Headway time between Vehicle n and m.
z	Inverse speed.
$\mathbf{x}$, u	State and control vector.
A, B	State and control matrices.
$\mathbf{A}, \mathbf{B}$	State and control block matrices.
O	Crossing order vector.
κ	Curvature of path.
$a_{n\text{lat}}$	Lateral acceleration of Vehicle n.
Δp	Discretized sampling step.
$\boldsymbol{\xi}$	Discrete state vector for all vehicles.
$\mathbf{u}$	Discrete control vector for all vehicles.
$J(\cdot)$	Cost function.
w	Cost function weight.
Δp_{mn}	Sample shift for Vehicles n and m.
Q, R, S	Weighting matrices.
f	Function.

List of Index sets

$\mathbb{R}$	Set of real numbers.
$\mathcal{N}$	Index set of vehicles.
$\mathcal{I}$	Index set of predefined paths.
P	Index set of discretized samples.

List of Indeces

$(\cdot)_i, (\cdot)_j$	Variable for path i, j.
$(\cdot)_n$	Variable for Vehicle n.
$(\cdot)^{\text{min}}, (\cdot)^{\text{max}}$	Minimum and maximum value.
$(\cdot)^{\text{d}}$	Discrete.
$(\cdot)_{k,p_0}$	Variable at discrete sample k starting at p_0.
$(\cdot)_{\text{r}}$	Reference.
$(\cdot)_0$	Initial value.
$(\cdot)_{\text{f}}$	Final value.
$(\cdot)_{\text{lin}}$	Linearization value.
$\widetilde{(\cdot)}$	Linearized function.
$(\cdot)^*$	Optimal value.

List of Sets

D_1, D_2	Sets of vehicle pairs with same exit leg.
P_{nm}	Set of collision points between Vehicle n and m.
$P^{\text{f1}}_{nm}, P^{\text{f2}}_{nm}$	The set of samples in the exit lane where the paths of Vehicles n and m does not overlap.
U_{p_0}	Set of discrete control signals along the prediction horizon.

List of Operators

$(\cdot)'$	Derivative with respect to distance.
$H(\cdot)$	Hessian operator.
$\partial/\partial p$	Partial derivative.
$\text{D}(\cdot)$	Discrete derivative.
$\lVert\cdot\rVert_p, \lVert\cdot\rVert_Q$	P-norm and weighted 2-norm.

of which can be solved efficiently using existing methods.

The combinatorial part consists of finding the order in which the vehicles should cross the intersection. Since the number of possible crossing orders grows rapidly with the number of vehicles, a crossing order is often found by applying a heuristic to reduce the computational complexity. Proposed heuristics range from reservation systems, in which vehicles reserve a time at which to cross the intersection, [19], [20], to more advanced heuristics where the crossing order is determined based on the solution of a mixed-integer program, [18]. The downside of heuristics is that there is no guarantee that the optimal crossing order is found, and the chosen one might be much worse than the optimal one. This would lead to higher emissions, higher congestion and making safety critical solutions more prevalent. An alternative to heuristics is to solve an optimization program to determine the optimal trajectories of the vehicles for all possible crossing orders. Solving for all crossing orders gives the optimum but is computationally expensive due to the vast number of possible crossing orders. It is, however, possible to somewhat reduce the number of crossing orders through removal of apparent non-optimal orderings in an ad-hoc fashion. Additionally, by assuming no overtaking and removing orders which give solutions that are identical to another solution the number of crossing orders can be reduced even further. This can greatly reduce the number of crossing orders that need to be solved for some scenarios, [13].

This paper expands upon the approach previously presented in [11] and [13]. Therein, a collaborative centralized MPC with an optimal control program sampled in distance is proposed for control of all autonomous vehicles in the intersection. The approach presented in [11] and [13] is limited to a single vehicle in each entry and exit lane when computing the crossing order as well as the vehicle trajectories. In [13], the collision avoidance inside the intersection is performed using local critical zones between every vehicle pair with intersecting paths to decrease the conservativeness of the algorithm and increase the throughput. In this paper, the MPC is expanded by allowing any number of vehicles in the same entry lane, the same exit lane or traveling along the same path throughout the intersection (this is the case if vehicles have the same entry and exit lane), which demands the introduction of additional collision avoidance constraints to prevent rear-end collisions. It is shown how the collision avoidance for local critical zones and vehicles along the same path can be written using the same linear formulation applied at different samples. Secondly, in a case study results are presented in which it is shown how the controller behaves for different cost functions and in the presence of disturbances in the form of lane blockage; which forces the central control to redirect vehicles onto different paths. Thirdly, it is concluded that the optimal control program can be extended to a wider set of applications, including, but not limited to, roundabouts and intersection grids due to its function as a cooperative speed controller when vehicles travel along the same paths.

The paper is organized as follows. Section II discusses the features and limitations of the intersections which the controller presented within this paper handles. In Section III the intersection problem is formulated and all constraints needed for the optimal control program are presented. In Section IV the full continuous optimal control program is given. In Section V the optimization problem introduced in Section IV is discretized and formulated in a receding horizon fashion. Section VI contains a case study in which it is demonstrated that the MPC can plan collision free and smooth trajectories for vehicles traveling through a four-way intersection. It is further shown that the local zones are less conservative than using the full intersection as a critical zone, how solutions differ when applying different cost functions and that the MPC can handle a sudden lane blockage. Section VII discusses extensions of the controller, such as inclusion of non-autonomous vehicles, solving for more general intersections, roundabouts

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

and intersection grids. Lastly, it discusses the possibility to introduce soft constraints and how to determine the crossing order, before conclusions are drawn in Section VIII.

II. Intersections

A traffic intersection may include crosswalks, bicycle paths, lanes with limited travel options and divisional islands [21]. While the methods in this paper can be applied to such a general intersection, the example used is a four-way intersection containing motorized vehicles only.

The *physical area* of an intersection is the area in which the roads merge, [22]. Each road that connects to the physical area is referred to as an *intersection leg* or leg for short. In this paper, the legs are numbered $1, 2, \ldots, L$. Besides its leg number, a leg is identified via its *leg angle*. The leg angle is the angle measured counterclockwise from the horizontal axis to the leg. Each leg consists of a number of lanes; the lanes where vehicles are traveling towards the physical area are called *entry lanes* and the lanes in which vehicles travel away from the physical area are called *exit lanes*. When vehicles enter the physical area via one of the entry lanes, they can travel through the physical area to any or a subset of the exit lanes, depending on the traveling limitations of the intersection. Lastly, a closed curve containing the physical area within its interior is introduced. Any vehicle inside this closed curve is assumed to be part of the intersection. This closed curve is called the *control boundary*.

As an example, consider the four-way intersection depicted in Fig. 1, where vehicles are allowed to turn right, drive straight or turn left inside the physical area, depicted in gray. There are four legs labeled Leg 1, Leg 2, Leg 3 and Leg 4 with leg angles 0°, 90°, 180° and 270°. Each leg consists of one entry lane and one exit lane, as shown by the arrows implying the direction of travel. There are a total of 12 predefined paths, three from each entry lane leading to each exit lane which does not belong to the same leg as the entry lane, see Fig. 1. It is assumed that all vehicles are traveling along one of these paths. Thus, a vehicle can change its predefined path to another path, only before reaching the physical area. Doing so inside the physical area would necessitate the introduction of additional paths. This is possible, but outside the scope of this paper. Lastly, the red circle in Fig. 1 is the control boundary.

III. Problem formulation

Mathematically, any intersection is represented by the tuple

$$(x_i(s),\ y_i(s),\ \kappa_i(s),\ \psi_i(s),\ v^{\text{sl}}(s),\ \mathcal{I},\ s),$$

where $x_i(s), y_i(s)$ are the global path coordinates for the predefined path $i \in \mathcal{I} = \{1, 2, \ldots, M\}$, $\kappa_i(s)$ is the path curvature and $\psi_i(s)$ is the path angle defined by the tangent angle at the sample s, which is the distance traveled along the arch length of the path. In general, the speed limit of the roads can vary between legs, but for simplicity it is in this paper assumed that all legs have the same speed limit $v^{\text{sl}}(s)$.

Now, consider $N > 0$ autonomous vehicles traveling through the intersection. For each Vehicle n within the set of vehicle indices $\mathcal{N} = \{1, \ldots, N\}$, it is assumed that

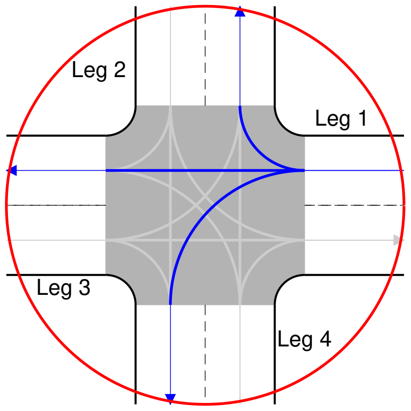


Fig. 1: An intersection with four legs, each with one entry and exit lane. There are 12 predefined paths, 3 starting from each entry lane. The paths starting from the first leg are depicted in blue. The red circle represents the control boundary. The gray area is the physical area.

1) the vehicle travels along one of the M predefined paths;
2) the assigned path is perfectly followed;
3) the speed $v_n \in \mathbb{R}$ and acceleration $a_n \in \mathbb{R}$ of the vehicle along the path can be varied;
4) the vehicle is not overtaking within the control boundary;
5) the vehicle clock is synchronized with that of the other vehicles.

A. Vehicle paths

Each Vehicle $n \in \mathcal{N}$ is traveling along a path $(x_n(p), y_n(p), \kappa_n(p), \psi_n(p), p)$ that overlaps with one of the intersection paths, i.e., $x_n(p) = x_i(p)$, $y_n(p) = y_i(p)$, $\kappa_n(p) = \kappa_i(p)$ and $\psi_n(p) = \psi_i(p)$ for all samples $p \in [p_{n0}, p_{nf}]$ and some $i \in \mathcal{I}$. Here, p_{n0} is the vehicles' initial position and p_{nf} is the vehicles' final position. A vehicle is part of the intersection, and therefore considered in the optimal control program, if p_{n0} is inside of the control boundary. It is also assumed that the final position p_{nf} is outside of the control boundary, i.e., the vehicles longitudinal motion is controlled at least until it leaves the physical area.

Choosing all sampling intervals $[p_{n0}, p_{nf}]$ to be of equal lengths p_{f}, each vehicle path is written as

$$(x_n(p + p_{n0}), y_n(p + p_{n0}), \kappa_n(p + p_{n0}), \psi_n(p + p_{n0}), p),$$

for all $n \in \mathcal{N}$ and all $p \in [0, p_{\text{f}}]$. Thus, p is a sampling variable representing the arc length of any vehicle along its predefined path relative to its starting position p_{n0}. For the sake of brevity, $x_n(p + p_{n0})$ will be written as $x_n(p)$ in the rest of the paper.

B. Longitudinal dynamics

Since sampling is done in distance via the sampling variable p, the traveling time of a vehicle at any sample is unknown. Therefore, the traveling time of each vehicle, $t_n(p)$, is chosen as a state and fulfills $t'_n(p) = 1/v_n(p)$, where $(\cdot)'$ denotes the derivative with respect to distance, i.e., $\mathrm{d}/\mathrm{d}p$. To get a linear state space model, n additional states are introduced, representing the inverse speed $z_n(p) = 1/v_n(p)$ of each vehicle, see [11] for details. The state vector for Vehicle n reads

$$\mathbf{x}_n(p) = [t_n(p), z_n(p)]^T,$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

resulting in a total of $2N$ states, two for each vehicle. Each Vehicle n is represented by a linear system

$$\mathbf{x}_n'(p) = A\mathbf{x}_n(p) + Bu_n(p),$$

where the control signal $u_n(p)$ is the derivative of the inverse speed of Vehicle n, i.e., $u_n(p) = z_n'(p)$, and

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

C. State and control constraints

Each Vehicle $n \in \mathcal{N}$ is subject to state and control constraints

$$\begin{aligned} &\mathbf{x}_n(p) \in [\mathbf{x}_n^{\min}(p), \mathbf{x}_n^{\max}(p)], \\ &u_n(p) \in [u_n^{\min}(p, z_n), u_n^{\max}(p, z_n)], \end{aligned} \tag{1}$$

where each inequality is imposed for all $p \in [0, p_{\mathrm{f}}]$ and

$$\mathbf{x}_n^{\min}(p) = [0, 1/v_n^{\max}(p)]^T, \quad \mathbf{x}_n^{\max}(p) = [\infty, 1/v_n^{\min}(p)]^T.$$

Here, $0 < v_n^{\min}(p) \leq v_n^{\max}(p)$ denote the minimum and maximum speed of Vehicle n, respectively.

An expression for the minimum and maximum bounds $u_n^{\min}(p, z_{\mathrm{n}})$ and $u_n^{\max}(p, z_{\mathrm{n}})$ on the control signal is found by using the lower and upper limit of the acceleration denoted $a_n^{\min}(p) < 0$ and $a_n^{\max}(p) > 0$, respectively. Observe that

$$\begin{aligned} u_n(p) &= z_n'(p) = \left(\frac{1}{v_n(p)}\right)' = -\frac{v_n'(p)}{v_n^2(p)} = -v_n'(p)z_n^2(p), \\ v_n'(p) &= \frac{\mathrm{d}}{\mathrm{d}p}v_n(p) = \frac{a_n(p)}{v_n(p)} = a_n(p)z_n(p), \end{aligned}$$

which combined yields $u_n(p) = -a_n(p)z_n^3(p)$. From this, the bounds on the control signal in (1) are given by

$$\begin{aligned} u_n^{\min}(p, z_n) &= -a_n^{\max}(p)z_n^3(p), \\ u_n^{\max}(p, z_n) &= -a_n^{\min}(p)z_n^3(p). \end{aligned} \tag{2}$$

For details on the choice of bounds, see [11] and [13].

1) Motion on curved paths: When vehicles turn right or left they move on paths with significant curvature, see Fig. 2. Thus, they are subjected to non-negligible lateral forces, which should be constrained to lie within a range that is realizable by actual vehicles. Since, the largest part of the lateral acceleration in a curve stems from the centripetal force, a maximum bound on the lateral acceleration is implicitly applied through the choice of the upper bound on the longitudinal speed. If the maximum lateral acceleration of Vehicle n is given by $a_{n\mathrm{lat}}^{\max}$, the upper bound on the longitudinal speed is equal to $\sqrt{a_{n\mathrm{lat}}^{\max}/\kappa_n(p)}$. Adopting the convention that division by a curvature of zero yields infinity, the maximum speed for Vehicle n at each sample is given by

$$v_n^{\max}(p) = \min\left(v^{\mathrm{sl}}(p), \sqrt{a_{n\mathrm{lat}}^{\max}/\kappa_n(p)}\right).$$

This means that when traveling along a straight part of the path the maximum speed is $v^{\mathrm{sl}}(p)$ while in a curve the maximum speed is the smallest of $\sqrt{a_{n\mathrm{lat}}^{\max}/\kappa_n(p)}$ and $v^{\mathrm{sl}}(p)$.

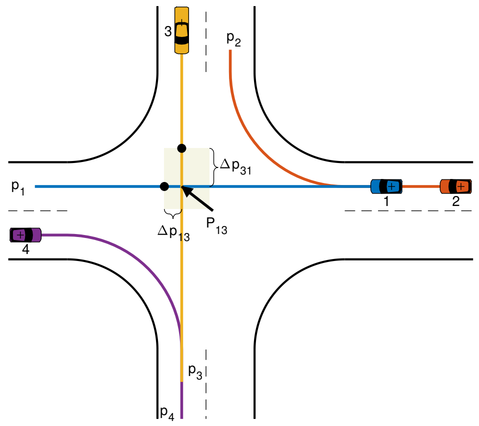


Fig. 2: Autonomous vehicles in an intersection. Each Vehicle n has a predefined path p_n depicted by a line. Vehicle 2 and Vehicle 4 are traveling along paths with significant curvature. Vehicles 1 and 3 have intersecting paths which give rise to the yellow rectangular critical zone. Collision avoidance constraints are imposed such that only one vehicle resides within the critical zone at the time. Vehicles 1 and 2 have overlapping paths prior to the physical area while vehicles 3 and 4 have overlapping paths after the physical area.

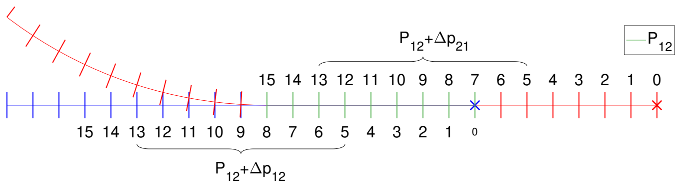


Fig. 3: A zoom in on the paths of Vehicle 1 and 2 in Fig. 2. The horizontal markings are samples, where red markings denote samples along the red path, the blue markings denote samples along the blue path and green markings denote samples which belong to both paths, i.e., the samples in P_{12}. The blue and red cross define the starting position of the Vehicle 1 and Vehicle 2, respectively. The numbers represent the distance, in samples, from the initial position. Rear-end collisions are avoided by restricting the time difference between all samples in the shifted intervals $P_{12} + \Delta p_{12}$ and $P_{12} + \Delta p_{21}$.

D. Collision avoidance constraints

To find collision free solutions, the collision avoidance constraints need to guarantee that there are no collisions along overlapping or intersecting paths. Here, it is understood that two vehicles have intersecting paths if their paths have the same global coordinates at exactly one point, while they have overlapping paths if their paths have the same global coordinates at more than one point. As an example, the paths of Vehicles 1 and 3 in Fig. 2 are said to intersect, while the paths of Vehicles 3 and 4 (and 1 and 2) are said to overlap. A vehicle traveling along a path which does not intersect or overlap with the path of any other vehicle is assumed to not be at risk of collision. A point where two vehicle paths have the same global coordinates is referred to as a collision point. For intersecting paths, the collision avoidance constraints can be interpreted as a critical zone who's entry and exit samples guarantee that only one vehicle resides within the critical zone at any given time, see the yellow rectangle in Fig. 2. For overlapping paths, a safety distance between the vehicles needs to be maintained to prevent rear-end collisions. This is done by imposing collision avoidance constraints at every sample along the overlapping path.

Let $O \in \mathbb{R}^N$ represent a given crossing order, i.e., O is a vector containing a permutation of the vehicle indices in $\mathcal{N}$. Also, let O_l be the l:th entry of O. Then, a Vehicle

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

n with the predefined path $(x_n(p), y_n(p), \kappa_n(p), \psi_n(p), p)$ crossing before a Vehicle m with the predefined path $(x_m(p), y_m(p), \kappa_m(p), \psi_m(p), p)$ has collision points with Vehicle m at the samples

$$p \in P_{nm} = \{p \in [0, p_\mathrm{f}] : (x_n(p), y_n(p)) = (x_m(p_1), y_m(p_1)),\\ p_1 \in [0, p_\mathrm{f}],\ n = O_l,\ m = O_{\bar{l}},\ l < \bar{l}\},$$

where $(x_n(p), y_n(p)) = (x_m(p_1), y_m(p_1))$ indicates that the paths of the two vehicles pass through the same point and $n = O_l,\ m = O_{\bar{l}},\ l < \bar{l}$ mean that Vehicle n crosses before Vehicle m. For vehicle pairs with overlapping paths, the set P_{nm} is an interval, for intersecting paths it is a point and for vehicle pairs which have neither overlapping nor intersecting paths, it is empty. However, imposing collision avoidance constraints directly at the entries of P_{nm} is not enough since it does not take into account the physical dimensions of the vehicles, the reference points of the vehicles nor the fact that the samples in P_{nm} are samples along the path of Vehicle n, which are not necessarily equal to the samples of the path of Vehicle m due to the shape of the path and/or different initial positions, see Fig. 3. To account for all of this, the scalars Δp_{nm}, Δp_{mn} are introduced to shift the samples for Vehicle n and m, respectively.

The collision avoidance constraint between Vehicle n and Vehicle m is a linear constraint on the traveling time

$$t_n(p + \Delta p_{nm}) - t_m(p + \Delta p_{mn}) \leq -t^{\mathrm{hw}}_{nm},\ \forall p \in P_{nm}, \quad (3)$$

where t^{hw}_{nm} is a non-negative, typically positive, headway time between Vehicle n and Vehicle m.

As an example, consider the scenario depicted in Fig. 2 with the crossing order $[1\ 2\ 3\ 4]$. To form the collision avoidance constraints we first find the collision points, i.e., all nonempty sets P_{nm}. The nonempty sets are: P_{13} which contains the sample at which the path of Vehicle 1 intersects with the path of Vehicle 3, P_{34} which contains the samples at which the path of Vehicle 3 overlaps with the path of Vehicle 4 and P_{12} at which the path of Vehicle 1 overlaps with the path of Vehicle 2. For Vehicles 1 and 3 the shifts, $\Delta p_{13} > 0$ and $\Delta p_{31} < 0$, are chosen such that $P_{13} + \Delta p_{13}$ is the sample at which Vehicle 1 exits the critical zone, and $P_{13} + \Delta p_{31}$ is the sample at which Vehicle 3 enters it, see Fig. 2. The collision avoidance constraint then reads

$$t_1(p + \Delta p_{13}) - t_3(p + \Delta p_{31}) \leq -t^{\mathrm{hw}}_{13},$$

and guarantees that Vehicle 1 exits the critical zone before Vehicle 3 enters it.

For the collision avoidance between Vehicles 1 and 2 the shift $\Delta p_{21} < 0$ is chosen such that a safety distance between the vehicles is introduced and the shift $\Delta p_{12} > 0$ is chosen to avoid collision at samples shortly after the paths split, see Fig. 3 which depicts a zoom in of the paths of Vehicle 1 and 2 along with samples (horizontal markings) and the sets P_{12}, $P_{12} + \Delta p_{12}$ and $P_{21} + \Delta p_{21}$. Applying similar reasoning to choose the overlapping paths of Vehicle 3 and 4 then gives the collision avoidance constraints

$$t_1(p + \Delta p_{12}) - t_2(p + \Delta p_{21}) \leq -t^{\mathrm{hw}}_{12},\ p \in P_{12},\\ t_3(p + \Delta p_{34}) - t_4(p + \Delta p_{43}) \leq -t^{\mathrm{hw}}_{43},\ p \in P_{34}.$$

E. Feasibility beyond the horizon

The collision avoidance constraints (3) guarantee that no collisions occur along overlapping and intersecting paths defined within the prediction horizon $[0, p_\mathrm{f}]$. However, since vehicles with the same exit lane do not necessarily have the same (absolute) position at the final sample p_f, there is still a risk of collision in the exit lane. As an example, consider again the scenario in Fig. 2. As illustrated by the yellow and purple lines, Vehicles 3 and 4 have the same exit lane and same path length but different final positions due to different starting positions and entry lanes. The collision avoidance constraints (3) only guarantee that Vehicles 3 and 4 do not collide within the prediction horizon, but not that collision is avoidable after the path of Vehicle 3 ends.

To guarantee collision avoidance in the exit lane after a vehicle reaches the end of their predefined path, additional constraints are introduced on the velocity and final acceleration of the vehicles. The constraints on the acceleration at the last point of the horizon read

$$u_n(p_\mathrm{f}) \in [-\epsilon_1, \epsilon_2],\ \forall n \in \mathcal{N}.$$

where ϵ_1 and ϵ_2 are small positive scalar numbers. This constraint guarantees that the acceleration of all vehicles is close to zero at the last sample and it is assumed that after the horizon the acceleration goes directly down to zero. Thus, adding a constraint which enforces that a vehicle traveling behind another vehicle in an exit lane has a velocity that is not higher than the vehicle in front, when the front vehicle reaches its end position within the prediction horizon, would guarantee collision avoidance beyond the horizon. To formulate such a speed constraint, it is necessary to find the set of all vehicle pairs that have the same exit lane. Further, we need to know which vehicle travels the shorter distance in the exit lane. This is found by studying the shifts Δp_{nm}, Δp_{mn} in the collision avoidance constraints (3). If the vehicles n and m have the same exit lane then $\Delta p_{nm} = 0$ if vehicle m is crossing first and $\Delta p_{mn} = 0$ if Vehicle n is crossing first. Hence, exactly one of the shifts is always zero for overlapping paths and identifies which vehicle has the least number of samples in the exit lane. Hence, the set

$$D_1 = \{(n, m) \in \mathcal{N}^2 : \psi_m(p_\mathrm{f}) = \psi_n(p_\mathrm{f}), n = O_l, m = O_{\bar{l}},\\ l < \bar{l}, \Delta p_{mn} < 0\},$$

denotes all vehicle pairs with the same exit lane when the vehicle in front, n, has the shorter path, while

$$D_2 = \{(n, m) \in \mathcal{N}^2 : \psi_m(p_\mathrm{f}) = \psi_n(p_\mathrm{f}), n = O_l, m = O_{\bar{l}},\\ l < \bar{l}, \Delta p_{mn} > 0\},$$

denotes all vehicle pairs with the same exit lane when the vehicle in front, n, has the longer path. Here, $\psi_m(p_\mathrm{f}) = \psi_n(p_\mathrm{f})$ guarantee vehicles have the same exit lane and $n = O_l, m = O_{\bar{l}}, l < \bar{l}$ guarantee Vehicle n crosses before Vehicle m. The speed constraints for the last few samples are then written as

$$z_n(p_\mathrm{f}) \leq z_m(p),\ \forall p \in P^{\mathrm{f1}}_{nm},\ \forall (n, m) \in D_1,\\ z_n(p) \leq z_m(p_\mathrm{f}), \forall p \in P^{\mathrm{f2}}_{nm},\ \forall (n, m) \in D_2,$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

where

$$P_{nm}^{\mathrm{f1}} = [p_\mathrm{f} + \Delta p_{mn}, p_\mathrm{f}], P_{nm}^{\mathrm{f2}} = [p_\mathrm{f} - \Delta p_{mn}, p_\mathrm{f}], \tag{4}$$

are the sets of samples between the endpoints of the paths of Vehicle n and m.

IV. Optimal control program

Let $\mathbf{x}_{n0} = [0, 1/v_{n0}]^T$ be the initial state vector of Vehicle n where v_{n0} is the initial velocity of Vehicle n. Then, the optimal control program, for a given crossing order, is formulated as

$$\underset{u_n(p)}{\text{minimize}} \sum_{n=1}^{N} J_n(\mathbf{x}_n(p), u_n(p), u_n'(p)), \tag{5a}$$

subject to

$$\mathbf{x}_n'(p) = A\mathbf{x}_n(p) + Bu_n(p), \qquad \forall n \in \mathcal{N}, \tag{5b}$$

$$\mathbf{x}_n(p) \in [\mathbf{x}_n^{\min}(p), \mathbf{x}_n^{\max}(p)], \qquad \forall n \in \mathcal{N}, \tag{5c}$$

$$u_n(p) \in [u_n^{\min}(p, z_n), u_n^{\max}(p, z_n)], \forall n \in \mathcal{N}, \tag{5d}$$

$$t_n(p + \Delta p_{nm}) - t_m(p + \Delta p_{mn}) \le -t_{nm}^{\mathrm{hw}}, \quad \forall n, m \in \mathcal{N}, \tag{5e}$$

$$z_n(p_\mathrm{f}) \le z_m(p), \qquad \forall (n, m) \in D_1, \tag{5f}$$

$$z_n(p) \le z_m(p_\mathrm{f}), \qquad \forall (n, m) \in D_2, \tag{5g}$$

$$\mathbf{x}_n(0) = \mathbf{x}_{n0}, \ u_n(p_\mathrm{f}) = 0, \qquad \forall n \in \mathcal{N}, \tag{5h}$$

where the constraints (5b)-(5d) hold for all $p \in [0, p_\mathrm{f}]$, (5e) holds for all $p \in P_{nm}$, (5f) holds for all $p \in P_{nm}^{\mathrm{f1}}$ and (5g) holds for all $p \in P_{nm}^{\mathrm{f2}}$.

All constraints in (5) are linear except for the nonlinear and non-convex input constraints (5d). The program can be iteratively solved using linearized convex subproblems. This solution method is commonly known as sequential convex programming (SCP), or, if the subproblem is a quadratic program, sequential quadratic programming (SQP), [23]. The choice of objective function (5a) is discussed in detail after discretization of the optimal control program, in Section V. Next, the linearization of the constraints (5d) is discussed.

A. Linearization of input constraints

To see that the constraints (5d) are concave, recall that the bounds are given by (2) and rewrite the bounds to

$$f_1(z_n, u_n) = -a_n^{\max} z_n^3(p) - u_n(p) \le 0,$$
$$f_2(z_n, u_n) = a_n^{\min} z_n^3(p) + u_n(p) \le 0,$$

and calculate the Hessian of both functions

$$H(f_1) = \begin{bmatrix} \frac{\partial^2 f_1(z_n,u_n)}{\partial z_n^2} & \frac{\partial^2 f_1(z_n,u_n)}{\partial z_n \partial u_n} \\ \frac{\partial^2 f_1(z_n,u_n)}{\partial u_n \partial z_n} & \frac{\partial^2 f_1(z_n,u_n)}{\partial u_n^2} \end{bmatrix} = \begin{bmatrix} -6a_n^{\max} z_n(p) & 0 \\ 0 & 0 \end{bmatrix},$$

$$H(f_2) = \begin{bmatrix} \frac{\partial^2 f_2(z_n,u_n)}{\partial z_n^2} & \frac{\partial^2 f_2(z_n,u_n)}{\partial z_n \partial u_n} \\ \frac{\partial^2 f_2(z_n,u_n)}{\partial u_n \partial z_n} & \frac{\partial^2 f_2(z_n,u_n)}{\partial u_n^2} \end{bmatrix} = \begin{bmatrix} 6a_n^{\min} z_n(p) & 0 \\ 0 & 0 \end{bmatrix}.$$

The Hessians are negative definite since $a_n^{\min} < 0$, $a_n^{\max} > 0$ and $z_n(p) > 0$ for all p, which implies that the constraints are concave, [24]. The linearization of a concave constraint is always an inner approximation; thus the linearization does not introduce non-realizable solutions.

To find the linearized bounds $\tilde{u}_n^{\min}$ and $\tilde{u}_n^{\max}$, $z_n^3(p)$ is linearized around the inverse speed $z_{n\mathrm{lin}}(p)$

$$z_n^3(p) \approx z_{n\mathrm{lin}}^3(p) + 3z_{n\mathrm{lin}}^2(p)(z_n(p) - z_{n\mathrm{lin}}(p)).$$

Using this together with (2) gives

$$\tilde{u}_n^{\min}(p, z_n) = a_n^{\max}(p)(2z_{n\mathrm{lin}}(p) - 3z_n(p))z_{n\mathrm{lin}}^2(p),$$
$$\tilde{u}_n^{\max}(p, z_n) = a_n^{\min}(p)(2z_{n\mathrm{lin}}(p) - 3z_n(p))z_{n\mathrm{lin}}^2(p),$$

which leads to the linearized bounds

$$u_n(p) \in [\tilde{u}_n^{\min}(p, z_n), \ \tilde{u}_n^{\max}(p, z_n)]. \tag{6}$$

Thus, the subproblem to be solved is (5), where (5d) is replaced by (6). This gives a QP that can be solved efficiently.

V. Model predictive control

In this section, the optimal control program (5) is discretized and written in a receding horizon fashion. Let $k \in \{p_0, p_0 + \Delta p, \ldots, p_0 + p_\mathrm{f}\} = P$ be the discretized sampling variable, P the set of discrete distance samples corresponding to the continuous sampling interval $[0, p_\mathrm{f}]$, p_0 the first sample and Δp the length of the discrete sampling interval. To simplify the formulation of the discretized optimal control program, the state and control variables for all vehicles at sample k are stacked into the vectors

$$\boldsymbol{\xi}_{k,p_0} = \left[\mathbf{x}_{1,k,p_0}^T, \ldots, \mathbf{x}_{N,k,p_0}^T\right]^T,$$
$$\mathbf{u}_{k,p_0} = \left[u_{1,k,p_0}, \ldots, u_{N,k,p_0}\right]^T, \tag{7}$$

where $\mathbf{x}_{1,k,p_0}$ is the state vector of the first vehicle at iteration k when the sampling starts at p_0 and u_{1,k,p_0} is the corresponding control signal. Discretizing the dynamical model (5b) using zero-order hold yields

$$\boldsymbol{\xi}_{k,p_0} = \mathbf{A}^\mathrm{d}\boldsymbol{\xi}_{k-\Delta p,p_0} + \mathbf{B}^\mathrm{d}\mathbf{u}_{k-\Delta p,p_0}, \ \forall k \in P \setminus p_0,$$
$$\boldsymbol{\xi}_{p_0,p_0} = \boldsymbol{\xi}_0,$$

,where $\boldsymbol{\xi}_0$ is the initial state vector stacked in the same order as the states in (7). Further, the state and input matrix is given by $\mathbf{A}^\mathrm{d} = \mathrm{Diag}(A^\mathrm{d}, \ldots, A^\mathrm{d}) \in \mathbb{R}^{2N \times 2N}$, and $\mathbf{B}^\mathrm{d} = [B^\mathrm{d}, \ldots, B^\mathrm{d}]^T \in \mathbb{R}^{2N}$, respectively. Here

$$A^\mathrm{d} = \begin{bmatrix} 1 & \Delta p \\ 0 & 1 \end{bmatrix}, \ B^\mathrm{d} = \begin{bmatrix} \frac{\Delta p^2}{2} & \Delta p \end{bmatrix}.$$

Letting the full input sequence be $U_{p_0} = [\mathbf{u}_{p_0,p_0}, \mathbf{u}_{p_0+\Delta_p,p_0}, \ldots, \mathbf{u}_{p_0+p_\mathrm{f},p_0}]$, the discretized optimal control program on receding horizon form reads

$$\underset{U_{p_0}}{\text{minimize}} \ J^\mathrm{d}(\boldsymbol{\xi}_{k,p_0}, \mathbf{u}_{k,p_0}), \tag{8a}$$

subject to

$$\boldsymbol{\xi}_{k+\Delta p,p_0} = \mathbf{A}^\mathrm{d}\boldsymbol{\xi}_{k,p_0} + \mathbf{B}^\mathrm{d}\mathbf{u}_{k,p_0}, \qquad \forall k \in P \setminus P_\mathrm{f}, \tag{8b}$$

$$\boldsymbol{\xi}_{k,p_0} \in [\boldsymbol{\xi}_{k,p_0}^{\min}, \boldsymbol{\xi}_{k,p_0}^{\max}], \qquad \forall k \in P, \tag{8c}$$

$$\mathbf{u}_{k,p_0} \in [\tilde{\mathbf{u}}_{k,p_0}^{\min}(\boldsymbol{\xi}_{k,p_0}), \tilde{\mathbf{u}}_{k,p_0}^{\max}(\boldsymbol{\xi}_{k,p_0})], \forall k \in P, \tag{8d}$$

$$t_{n,k+\Delta p_{nm}} - t_{m,k+\Delta p_{mn}} \le -t_{nm}^{\mathrm{hw}}, \quad \forall k \in P_{nm}^\mathrm{d}, \ \forall n, m \in \mathcal{N}, \tag{8e}$$

$$z_{n,p_0+p_\mathrm{f}} \le z_{m,k}, \qquad \forall k \in P_{nm}^{\mathrm{f1}}, \ \forall (n, m) \in D_1, \tag{8f}$$

$$z_{n,k} \le z_{m,p_0+p_\mathrm{f}}, \qquad \forall k \in P_{nm}^{\mathrm{f2}}, \ \forall (n, m) \in D_2, \tag{8g}$$

$$\boldsymbol{\xi}_{p_0,p_0} = \boldsymbol{\xi}_0, \ \mathbf{u}_{p_0+p_\mathrm{f},p_0} = \mathbf{0}, \tag{8h}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

where $\boldsymbol{\xi}^{\min}_{k,p_0}$, $\boldsymbol{\xi}^{\max}_{k,p_0}$, $\tilde{\mathbf{u}}^{\min}_{k,p_0}$ and $\tilde{\mathbf{u}}^{\max}_{k,p_0}$ are vectors containing the stacked bounds and initial states, respectively, in the same way as the states and inputs in (7). Lastly, the cost function J^{d} is the discretized version of (5a).

After solving the discretized problem (8), the first optimal control signal, $\mathbf{u}^*_{p_0,p_0}$, is used to calculate the states of all vehicles at sample $p_0 + \Delta p$. This corresponds to all vehicles moving Δp meters on their paths. Then $\mathcal{N}$, the shifts Δp_{nm}, Δp_{mn}, and the set P_{nm} are updated along with the state bounds, control bounds, the initial and final states before the optimal control program is resolved over a shifted horizon.

A. Convex cost functions

In this section it is shown how the commonly used cost functions: minimum time, trajectory tracking [25] and minimum discomfort are formulated for the optimal control program (8). For an explanation on how the time formulated costs correlate with the spatial costs presented in this paper, see [11].

1) Trajectory tracking: States and inputs are tracked using the cost function

$$J^{\mathrm{d}}(\cdot) = \sum_{k\in P} \big(\|\boldsymbol{\xi}_{k,p_0} - \boldsymbol{\xi}_{\mathrm{r},k,p_0}\|^2_Q \\ + \|\mathbf{u}_{k,p_0} - \mathbf{u}_{\mathrm{r},k,p_0}\|^2_R + \|\mathbf{D}\mathbf{u}_{k,p_0}\|^2_S\big). \tag{9}$$

Here, Q, R and S are positive definite weighting matrices while $\boldsymbol{\xi}_{\mathrm{r},k,p_0}$ and $\mathbf{u}_{\mathrm{r},k,p_0}$ are the reference trajectories for the states and inputs, respectively. Further, $\mathbf{D}\mathbf{u}_{k,p_0}$ refers to the discrete derivative of $\mathbf{u}_{k,p_0}$ and is included to mitigate high spikes in jerk which may result from using pure tracking costs. Typically, the input reference $\mathbf{u}_{\mathrm{r},k,p_0}$ is zero.

2) Minimum time: A common goal for traffic control of intersections is to maximize the throughput. Mathematically, this is formulated as a cost function that minimizes the p-norm of the final traveling time of each vehicle, i.e,

$$J^{\mathrm{d}}(\cdot) = \|\boldsymbol{\xi}_{p_0+p_{\mathrm{f}},p_0}\|_{Q,p},$$

where $Q = \operatorname{diag}(q_1, 0, q_2, 0, \ldots, q_N, 0) \in \mathbb{R}^{2N\times 2N}$ is the weighting matrix for Vehicle $n \in \mathcal{N}$. The most intuitive choice of norm is $p = 1$ which corresponds to minimizing the weighted sum of all final times and $p = \infty$ which corresponds to minimizing the largest final time.

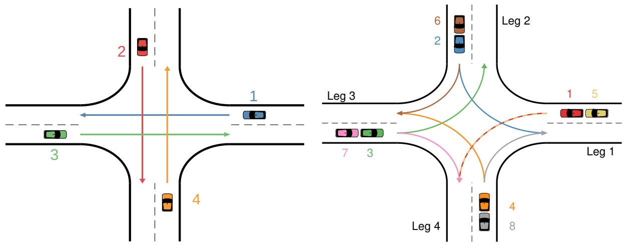
Minimizing only the time can lead to sudden shifts in acceleration and jerk resulting in an uncomfortable ride for the occupants of the vehicle. This is mitigated by adding penalties for the input signal $\mathbf{u}_{k,p_0}$ and its derivative $\mathbf{D}\mathbf{u}_{k,p_0}$, i.e.,

$$J^{\mathrm{d}} = \|\boldsymbol{\xi}_{p_0+p_{\mathrm{f}},p_0}\|_{Q,p} + \sum_{k\in P}(\|\mathbf{u}_{k,p_0}\|^2_R + \|\mathbf{D}\mathbf{u}_{k,p_0}\|^2_S). \tag{10}$$

where R and S are positive definite weighting matrices.

VI. Case study

In this section, the proposed MPC is validated on the scenarios in Fig. 4. Besides the solver metrics such as solution time and closeness to the optimal solution there are many metrics that are interesting when studying the effectiveness of an intersection algorithm, including: completion time, i.e., how long it takes before the last vehicle leaves the intersection; how well vehicles track their reference speed (if there is one), average time it takes for a vehicle to travel through the intersection, level of congestion and energy consumption.



(a) Scenario 1: Four vehicles going straight. (b) Scenario 2: Eight vehicles turning.

Fig. 4: Scenarios investigated in the case study.

TABLE I: Vehicle specific parameters for the four vehicles of Scenario 1 and the eight vehicles of Scenario 2. Symbol * implies that the crossing order is optimized and not calculated beforehand.

Description	Values
Initial position (m)	[50, 60, 50, 60]
	[60, 70, 70, 70, 80, 85, 85, 85]
Initial speed (km/h)	[36, 38, 40, 42]
	[37, 36, 40, 30, 40, 40, 40, 40]
Initial acceleration ($\mathrm{m/s^2}$)	[0, 0, 0, 0]
	[0, 0, 0, 0, 0, 0, 0, 0]
Reference speed (km/h)	[36, 38, 40, 42]
	[50, 50, 50, 50, 50, 50, 50, 50]
Crossing order	*
	[1, 3, 2, 5, 6, 7, 4, 8]

Not all these metrics can be optimal simultaneously. For instance, tracking a reference speed and minimizing completion time are competing objectives unless the reference speed is equal to the upper limit of the speed. Which metrics are most important for the current application should factor into the choice of cost function, see Section V-A.

The scenarios in Fig. 4 take place in a four-way intersection where all leg angles are 90° and each leg consists of two lanes with a width of 5 m. The physical area is 30 m across and the control boundary is a circle with a radius of 90 m whose center coincides with the center of the physical area. For both scenarios, it is assumed that $v^{\min}_n(p) = 1\,\mathrm{km/h}$ and

$$v^{\max}_n(p) = \begin{cases} 50\,\mathrm{km/h} & \kappa_n(p) = 0, \\ 18\,\mathrm{km/h} & \kappa_n(p) = 0.08, \\ 21.3\,\mathrm{km/h} & \kappa_n(p) = 0.0571, \end{cases} \tag{11}$$

where the curvature $\kappa_n(p) = 0$ corresponds to a straight path, $\kappa_n(p) = 0.08$ corresponds to a right turn and $\kappa_n(p) = 0.0571$ corresponds to a left turn. The speed values in (11) correspond to a maximum lateral acceleration of $a^{\max}_{n\mathrm{lat}} = 2\,\mathrm{m/s^2}$ for all $n \in \mathcal{N}$ and a constant road speed limit $v^{\mathrm{sl}} = 50\,\mathrm{km/h}$ for all intersection legs. The longitudinal acceleration limits are $a^{\min}_n = -3.5\,\mathrm{m/s^2}$ and $a^{\max}_n = 2\,\mathrm{m/s^2}$ for all $n \in \mathcal{N}$. The sampling interval is chosen to be $\Delta p = 1\,\mathrm{m}$.

Two cost functions are used in the case study. The first cost function is a combination of speed tracking and minimization

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

TABLE II: Minimum headway difference between vehicles inside a local critical zone. Values higher than $-1.10\,\text{s}$ would imply a violation of the collision avoidance constraints.

	Scenario 1	Scenario 2	
	local cz	Tracking	Min. Time
$t_1 - t_2$	$-1.10\,\text{s}$	$-3.53\,\text{s}$	$-4.10\,\text{s}$
$t_1 - t_4$	$-1.36\,\text{s}$	$-7.045\,\text{s}$	$-7.67\,\text{s}$
$t_3 - t_2$	$-2.67\,\text{s}$	$-1.10\,\text{s}$	$-1.10\,\text{s}$
$t_3 - t_4$	$-1.10\,\text{s}$	$-11.09\,\text{s}$	$-11.20\,\text{s}$
$t_2 - t_5$	N/A	$-1.10\,\text{s}$	$-1.10\,\text{s}$
$t_5 - t_4$	N/A	$-1.10\,\text{s}$	$-1.10\,\text{s}$

of discomfort written on the form (9), with the weights

$$\begin{aligned} Q &= \operatorname{diag}(0, q_1, 0, q_2, \dots, 0, q_N) \in \mathbb{R}^{2N\times 2N}, \\ R &= \operatorname{diag}(r_1, r_2, \dots, r_N) \in \mathbb{R}^{N\times N}, \\ S &= \operatorname{diag}(s_1, s_2, \dots, s_{N-1}) \in \mathbb{R}^{(N-1)\times(N-1)}, \end{aligned}$$

where

$$q_n = \Delta p v_{\text{nm}}^3 q_{\text{nv}},\ r_n = 2\Delta p v_{\text{nm}}^5 q_{\text{na}},\ s_n = \frac{2 q_{n\text{j}} v_{nm}^7}{\Delta p},$$

and $q_{\text{nv}} = 1$, $q_{\text{na}} = 1$ and $q_{n\text{j}} = 0.5$ for all $n \in \mathcal{N}$ are the weights corresponding to a time formulated problem, see [11]. Further, v_{nm} is the mean velocity of the linearization velocity, i.e., $v_{nm} = \text{mean}(1/z_{n\text{lin}})$, and the reference vectors are

$$\boldsymbol{\xi}_{\text{r},k,p_0} = [0, v_{r,k,1}, \dots, 0, v_{r,k,N}],\ \boldsymbol{u}_{\text{r},k,p_0} = \mathbf{0}$$

where $v_{r,k,n}$ is the reference velocity of Vehicle n. This cost function is referred to as the tracking cost in this case study.

The second cost function is a combination of minimizing the sum of the final vehicle traveling times and discomfort. It is written on the form (10) with R and S chosen as for the tracking cost, $p = 1$ and

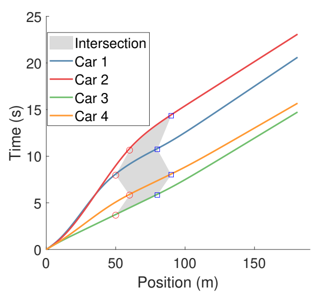
$$Q = \operatorname{diag}(1, 0, 1, 0, \dots, 1, 0) \in \mathbb{R}^{2N\times 2N}.$$

This is referred to as the minimum time cost in this case study. In the first MPC iteration, when no previous solution is available, the inverse linearization speed which the input constraints (8d) depend on is chosen to be the inverse reference speed of each vehicle $n \in \mathcal{N}$ for the tracking cost and the upper speed limit for the minimum time cost. Thereafter, for both cost functions, the linearization is done about the previous solution, shifted by one example.

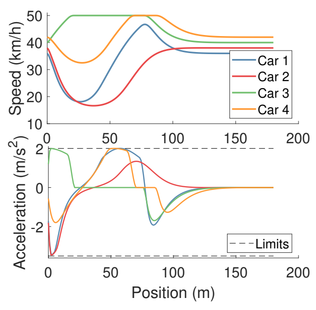
The proposed controller is implemented in MATLAB and the optimal control program (8) is solved using the built-in solver quadprog. All simulations are performed on a laptop with Intel Core i7-5600 CPU at 2.60GHz with 16 GB RAM.

The case study is divided into three parts: comparison between local and global critical zones, a comparison between cost functions and an investigation of how well the algorithm handles a sudden lane blockage. In addition to the figures and analysis presented within this paper there is also an animation showing the optimal solution of each case solved[1].

[1]https://chalmersuniversity.app.box.com/folder/140303688929

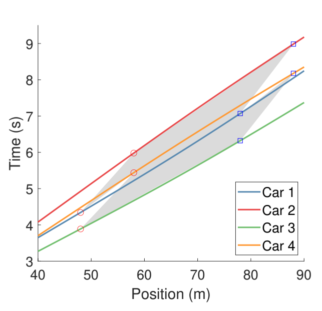


(a) Traveling time at each position for all vehicles. The solution is collision free since there is only one vehicle inside the physical area (depicted in gray) at any given time.

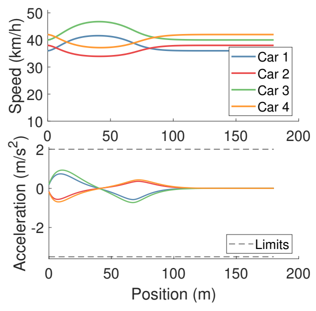


(b) Speed and acceleration for all vehicles. The acceleration constraints are fulfilled since the acceleration of all vehicles is between the dashed limits.

Fig. 5: Results of solving the optimal control program for Scenario 1 with a tracking cost function and using the physical area as a global critical zone.



(a) Traveling time at each sample.



(b) Velocity and acceleration for all vehicles.

Fig. 6: Results of solving the optimal control program for Scenario 1 with a tracking cost function and using local critical zones.

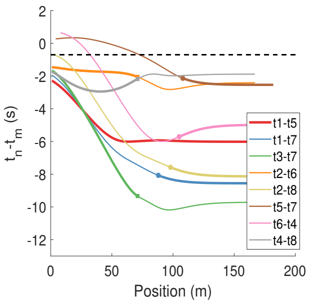
A. *Comparison of local and global critical zone*

In this part of the case study, the proposed controller is applied to Scenario 1; a scenario where four vehicles are moving straight through an intersection, as depicted in Fig. 4a. This scenario is similar to the one we solved in [13] and is resolved here using the new MPC formulation for repeatability and for its simplicity, before the more complicated Scenario 2 is solved. In addition to the parameters introduced in the previous section, the time headway is chosen to be $t_{nm}^{\text{hw}} = 1.1\,\text{s}$ for all $n, m \in \mathcal{N}$ and the initial conditions are as shown in the column labeled Scenario 1 of Table I. The tracking cost is chosen as the cost function.

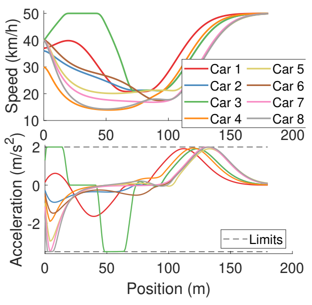
Scenario 1 is solved for all crossing orders and the crossing order with the lowest cost is chosen, i.e., the crossing order and vehicle trajectories are optimized. A solution is found for both the case of local critical zones and the case where the full physical area is considered a global critical zone within which only one vehicle is allowed at the time.

The result for the global critical zone is presented in Fig. 5. The Fig. 5a shows the traveling time of each vehicle plotted against its position. The gray area represents the physical area, which is also the global critical zone. Since no vehicles have overlapping paths the solution is collision free if there is at most one vehicle inside the physical area at the time. It can be seen that this is the case for the solution in Fig. 5a since the time at which a vehicle enters the physical area, represented by a red circle, is always lower than the exit time of the preceding vehicles, represented by a blue square.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

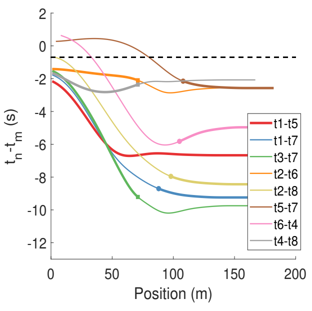


(a) Time headway between vehicles with overlapping paths.

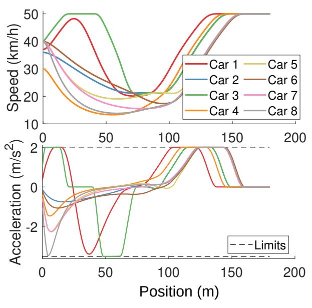


(b) Velocity and acceleration for all vehicles.

Fig. 7: The solution for Scenario 2 using the tracking cost. Fig. a) The thick part of the coloured lines represent the positions at which the paths of the vehicle pairs overlap. Overlapping paths are collision free if the thick part of the lines are below the negative headway $-0.7\,\mathrm{s}$, depicted by the black dotted line. b) The acceleration bounds are fulfilled since the acceleration of all vehicles is between the dotted limits.



(a) Time headway between vehicles with overlapping paths.



(b) Velocity and acceleration for all vehicles.

Fig. 8: The solution for Scenario 2 using the minimum time cost. Fig. a) The thick part of the coloured lines represent the positions at which the paths of the vehicle pairs overlap. Overlapping paths are collision free if the thick part of the lines are below the negative headway $-0.7\,\mathrm{s}$, depicted by the black dotted line. b) The acceleration bounds are fulfilled since the acceleration of all vehicles is between the dotted limits.

The results for the local critical zones are presented in Fig. 6. Here, the traveling times plotted against the vehicle positions are shown in Fig. 6a. However, since the physical area is no longer the critical zone it is not enough to study this plot to determine if the collision avoidance constraints are fulfilled or not. Indeed, as expected, it is seen in this figure that several vehicles occupy the physical area at the same time. Instead, collision avoidance is verified when there is only one vehicle within each local critical zone at the time. That this is the case in this Scenario is seen in the second column of Table II, which lists the time difference between the exit of the first vehicle and entry of the second vehicle into each local critical zone, and since all of them are equal to or less than $-1.1\,\mathrm{s}$, the solution fulfills the collision avoidance constraints.

From Fig. 5a it is seen that the optimal crossing order is $[3,\ 4,\ 1,\ 2]$ when the critical zone is the full physical area. When the problem is solved with local critical zones the optimal crossing order is instead $[3,\ 1,\ 4,\ 2]$ (this is not easy to discern from Fig. 6 and Table II). Thus, when using local critical zones, compared to a global critical zone, Vehicle 1 swaps places with Vehicle 4 in the crossing order. This together with the less conservative collision avoidance constraints that allow for several vehicles inside the physical area at the same time makes it possible for the vehicles to accelerate/decelerate less and keep a speed closer to their reference throughout the simulation, compare Fig. 5b and Fig. 6b. Especially vehicles 1 and 2 have to decelerate significantly more in the case of the global critical zone and due to this, it takes the last vehicle, Vehicle 2, $14.34\,\mathrm{s}$ to leave the physical area for the global critical zone while only $8.87\,\mathrm{s}$ to leave the physical area in case of the local critical zones.

B. Comparison of cost functions

In this section, a scenario is solved with two vehicles in each of the four lanes, see Fig. 4b. This scenario is referred to as Scenario 2 throughout the rest of the paper. Scenario 2 is here solved using the tracking cost and the minimum time cost, see Section V-A.

In addition to the parameters introduced in Section VI, the time headway is chosen to be $t_{nm}^{\mathrm{hw}} = 0.7\,\mathrm{s}$ for all vehicles $n, m \in \mathcal{N}$ with overlapping paths and $t_{nm}^{\mathrm{hw}} = 1.1\,\mathrm{s}$ for all vehicles $n, m \in \mathcal{N}$ with intersecting paths. Finally, the initial conditions are as shown in the third column of Table I.

The solution obtained when applying the tracking cost is presented in Fig. 7, while the corresponding solution for using the minimum time cost is shown in Fig. 8.

To determine that the tracking cost solution is collision free, we study Fig. 7a and the third column of Table II.The Table II lists the headway difference for all vehicles with intersecting paths, in the same way as for Scenario 1. Fig. 7a, depicts the headway difference between all vehicles with overlapping paths against the position. The thicker part of the lines corresponds to the positions at which collision avoidance constraints are active. For example, the red line labeled $t_1 - t_5$ is thick for all the plotted samples since Vehicles 1 and 5 have the same path, while the brown line, labeled $t_5 - t_7$, is thick from $110\,\mathrm{m}$ onward since the Vehicles 5 and 7 have overlapping paths after the physical area. Here, a circle indicates the start of the overlapping path and a square represents its end. Since the headway difference is below $-0.7\,\mathrm{s}$ for all vehicle pairs with overlapping paths at all samples where there is a risk of a rear-end collision and the headway difference is below $-1.1\,\mathrm{s}$ for all vehicle pairs with intersecting paths, the solution obtained by applying the tracking cost is collision free. The corresponding plot and table column for the minimum time objective, Fig. 8a and the fourth column of Table II, show that the minimum time solution is also collision free.

The headway difference plots for the tracking cost, Fig. 7a, and the minimum time cost, Fig. 8a, are similar and so are the headway differences for the intersecting paths, see Table II. However, if one studies the velocities and accelerations plotted in Fig. 7b and Fig. 8b there are some differences. In the case of the minimum time cost Vehicle 1 applies maximum acceleration at the start of the scenario and almost reaches the speed limit of $50\,\mathrm{km/h}$. However, for the tracking cost, where $50\,\mathrm{km/h}$ is the speed limit and the speed tracked, Vehicle 1 stays far below that value until after the physical area has been passed. A similar pattern is seen for most other vehicles, where the speed for the earlier samples are kept higher in the minimum time solution than in the tracking solution. Further, the velocities of Vehicles 7 and 8 have smoother trajectories

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

TABLE III: Minimum headway difference between vehicle pairs at risk of collision. Values lower than $-0.7\,\mathrm{s}$ for overlapping paths and lower than $-1.1\,\mathrm{s}$ for intersecting paths imply there is no violation of the collision avoidance constraints.

	$10\,\mathrm{m}$	$30\,\mathrm{m}$	$50\,\mathrm{m}$
Intersection			
$t_1 - t_2$	$-1.69\,\mathrm{s}$	$-3.52\,\mathrm{s}$	$-4.23\,\mathrm{s}$
$t_1 - t_4$	$-5.22\,\mathrm{s}$	$-6.97\,\mathrm{s}$	$-8.95\,\mathrm{s}$
$t_2 - t_5$	$-1.10\,\mathrm{s}$	$-1.10\,\mathrm{s}$	$-1.48\,\mathrm{s}$
$t_5 - t_4$	$-1.10\,\mathrm{s}$	$-1.13\,\mathrm{s}$	$-1.10\,\mathrm{s}$
Overlapping			
$t_1 - t_3$	$-2.13\,\mathrm{s}$	$-2.71\,\mathrm{s}$	$-4.28\,\mathrm{s}$
$t_1 - t_5$	$-2.14\,\mathrm{s}$	$-2.22\,\mathrm{s}$	$-2.22\,\mathrm{s}$
$t_1 - t_7$	$-6.24\,\mathrm{s}$	$-7.93\,\mathrm{s}$	$-10.77\,\mathrm{s}$
$t_3 - t_5$	$-1.42\,\mathrm{s}$	$-2.91\,\mathrm{s}$	$-1.62\,\mathrm{s}$
$t_3 - t_7$	$-1.54\,\mathrm{s}$	$-1.55\,\mathrm{s}$	$-1.55\,\mathrm{s}$
$t_2 - t_6$	$-1.25\,\mathrm{s}$	$-1.41\,\mathrm{s}$	$-1.41\,\mathrm{s}$
$t_2 - t_8$	$-7.83\,\mathrm{s}$	$-9.71\,\mathrm{s}$	$-8.89\,\mathrm{s}$
$t_5 - t_7$	$-2.13\,\mathrm{s}$	$-2.12\,\mathrm{s}$	$-1.24\,\mathrm{s}$
$t_6 - t_4$	$-4.64\,\mathrm{s}$	$-4.75\,\mathrm{s}$	$-6.29\,\mathrm{s}$
$t_4 - t_8$	$-1.70\,\mathrm{s}$	$-1.71\,\mathrm{s}$	$-1.71\,\mathrm{s}$

in the minimum time cost. The price to pay for using the minimum time cost is increased acceleration of Vehicle 1 at the start of the simulation. It is also seen that the vehicles accelerate faster, up to $50\,\mathrm{km/h}$, after the physical area for the minimum time case than in the tracking case. As expected, the sum of travel times until all vehicles reach their final position is shorter for the minimum time cost, $191.4\,\mathrm{s}$, than for the tracking cost, $197.7\,\mathrm{s}$, while the last vehicle (Vehicle 8) reaches the last sample at $30.1\,\mathrm{s}$ and $29.5\,\mathrm{s}$, respectively. Hence, the minimum time solution pushes the vehicles through the intersection faster than the tracking cost.

C. Lane blockage

In this last part of the case study, Scenario 2 is solved with an added disturbance. The disturbance is the blockage of the exit lane of Leg 2. The blockage forces the controller to redirect all vehicles, in this case only Vehicle 3, from the exit lane of Leg 2 to another exit lane. In this scenario it is assumed that Vehicle 3 is redirected to the exit lane of Leg 4, i.e., after the lane blockage all vehicles travel along the paths defined in Fig. 4b except Vehicle 3, which is now traveling along the same path as Vehicle 7. The Scenario is solved for 3 cases: when the blockage occurs after 10, 30 and 50 samples. Note that in all these three cases the blockage occurs prior to Vehicle 3 reaching the physical area to guarantee that it can choose one of the other predefined paths. Lane blockages occurring after Vehicle 3 has entered the physical area would require a method to construct a new path for the vehicle to follow, which is not a part of this paper.

The result for the three cases is presented in Table III and Figs. 9a-9c. The table lists the minimum time headway along overlapping paths and the headway difference for intersecting paths. The time headways are $1.1\,\mathrm{s}$ for all intersecting paths and $0.7\,\mathrm{s}$ for all overlapping paths. Therefore, Table III shows that the solver yields collision free solutions for all three cases.

To see the effect of the blockage, the three Figs. 9a-9c are compared. The direct consequence of the blockage is that Vehicle 3 takes a right turn instead of a left turn, which forces it to lower its speed when getting close to the physical area. As is seen in Fig. 9a-9c this deceleration starts at the moment the blockage occurs, and is more severe the later the blockage occurs. Further, the change of the path of Vehicle 3, makes it possible for Vehicle 2 to get through the physical area faster due to it no longer having to wait for Vehicle 3. This is why the speed and acceleration of Vehicle 2 jumps in Figs. 9a-9b at the sample where the blockage occurs. This jump is not as prominent in Fig. 9c, since Vehicle 2 is close to the physical area when the blocking occurs and has a speed already close to its allowed maximum inside the physical area, $21.3\,\mathrm{km/s}$. Similar behavior is observed for Vehicle 6. This is due to the fact that Vehicle 6 travels behind Vehicle 2 before the physical area, which allows it to accelerate when Vehicle 2 does.

Lastly, the speed and acceleration of Vehicle 1 is affected. This is because it has the same exit lane as Vehicle 3 after the blockage occurs. Since, the crossing order is $[1, 3, 2, 5, 6, 7, 4, 8]$, Vehicle 1 has to accelerate ones the controller becomes aware of the blockage to reach the exit lane before Vehicle 3. This is illustrated by the jump in acceleration at the occurrence of the blockage in Figs. 9a-9c.

VII. Wider applications of the controller

In this paper, it has been shown how the algorithm for cooperative coordination of vehicles driving in traffic intersections is extended to include vehicles traveling in the same lane. This enables the algorithm to be used as a cooperative speed controller before and after the intersection. Further, this enables other extensions, e.g., cooperative control in roundabouts and intersection grids. Other useful extensions include the possibility of handling mixed traffic, optimization of the crossing order and avoidance of feasibility problems in the presence of noisy or non-existing data using soft constraints.

A. Mixed traffic

Since it is not realistic that all of today's vehicles are replaced by autonomous vehicles at the same time, it is beneficial for intersection algorithms to handle mixed traffic, i.e., traffic with both autonomous and non-autonomous vehicles.

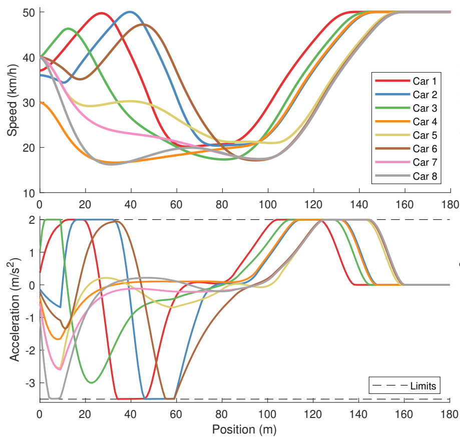
It is possible to include non-autonomous vehicles into the control program (8) given that the centralized controller has access to a predefined path and planned/predicted longitudinal speed along the path for all non-autonomous vehicles. Then, collision avoidance constraints similar to (8e) can be formulated between non-autonomous and autonomous vehicles. However, it is not possible to prevent non-autonomous vehicles from colliding with each other.

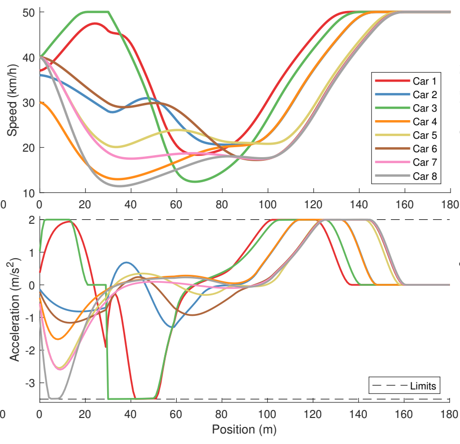
B. A grid of intersections

Since, the optimization program (8) models collision avoidance for vehicles with intersecting and overlapping paths, it is straightforward to generalize the MPC to a grid of intersections, as the one in Fig. 10. For each intersection, a program identical to (8) is solved, and in between intersections, i.e., outside of the control boundaries, the same program but with only overlapping collision avoidance constraints is solved. If the distance between intersections is small, it is possible to use a large control boundary encompassing multiple intersections

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533


(a) Blockage occurs after 10 samples.


(b) Blockage occurs after 30 samples.

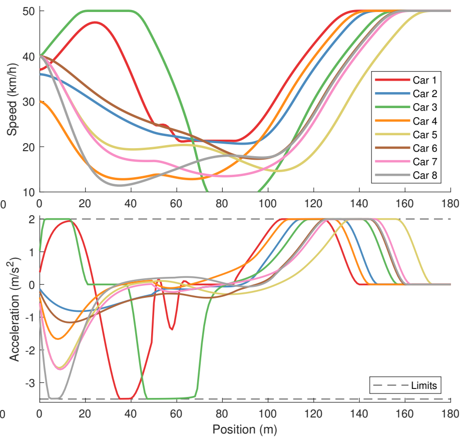

(c) Blockage occurs after 50 samples.

Fig. 9: The speed and acceleration for three cases where Vehicle 3 in Scenario 2 is forced to change its exit lane from Leg 2 to Leg 4 due to a blockage in the exit lane of Leg 2 after 10, 30 and 50 samples.

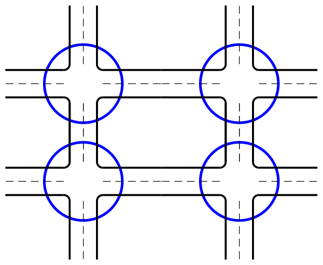


Fig. 10: A grid of four intersections. Each blue circle represents the control boundary of the respective intersection.

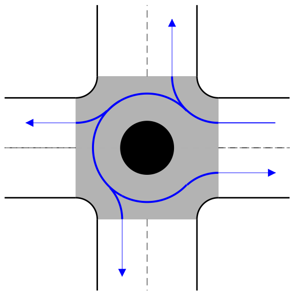


Fig. 11: A roundabout with four legs, each with an entry and exit lane. Paths starting in the first leg shown in blue.

and regarding them as a single large intersection while if the distance between the intersections is large, one might consider using a controller between the intersections that permit overtaking, e.g., the one proposed in [26].

C. Generalized intersections

The algorithm can be applied to other types of intersections besides the four-way intersection. As long as the predefined paths are well defined for all vehicles, the intersection can have any number of legs and lanes. It is also possible to apply the algorithm to merging and splitting of roads. One application that is enabled due to the extended possibility of avoiding rear-end collisions is cooperative driving in roundabouts, Fig. 11. However, including road users with a different movement pattern, such as pedestrians and/or bicyclists is more complex, since it would require predicting their path and speed along that path. Such predictions are often based on a probabilistic model and would require computation of robust bounds to be implemented in the proposed algorithm.

D. Determining the crossing order

One straightforward way of determining the crossing order is to solve the optimization program (8) for all possible crossing orders and choose the solution with the lowest cost (or one of the solutions with the lowest cost). However, the number of possible crossing orders increases rapidly with the number of vehicles. If every vehicle is traveling along its own lane, the number of possible crossing orders is equal to the number of ways to permute the vehicle indices. This means that for N vehicles there are $N!$ possible crossing orders. As an example, if eight vehicles are crossing the intersection, there are $8! = 40320$ possible crossing orders. However, typically there are more vehicles than intersection legs, which lowers the number of possible crossing orders, due to the assumption of no overtaking. Considering the scenario depicted in Fig. 4b, which contains eight vehicles but only four intersection legs, the number of combinations that need to be considered when removing the possibility of overtaking is 2520. Further, some crossing orders might be identical in the sense that they provide the same cost when (8) is solved. This is due to the fact that interchanging two adjacent vehicles in the crossing order, which do not have intersecting or overlapping paths, does not affect the formulation of the optimal control program (8).

By only considering the crossing orders in the Scenario in Fig. 4b which provides a unique solution to (8), the number of orderings are reduced further to 14. While it was possible to reduce the number of crossing orders significantly in this particular case, the reduction is heavily dependent on the predefined paths of the vehicles. For more details on how to find the unique orderings see the paper [13].

E. Soft constraints

In the case study, Section VI, the measurement of states and inputs are assumed to be perfect and instantaneous. In reality, noisy and/or delayed measurements might lead to small violations of constraints and infeasibility. A common solution to this is to use soft constraints, [24]. This introduces slack variables for the state bounds (8c) and collision avoidance constraints (8e), which allows for violation of these constraints. The square of these slack variables are then highly penalized in the cost to discourage constraint violation.

VIII. Conclusion

In this paper, a centralized model predictive controller for optimal control of autonomous vehicles at intersections is

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

presented. The intersection is defined mathematically as a number of known paths along which the vehicles are allowed to travel. The optimal control program is based on the sequential quadratic modeling first introduced in [11], but extended with collision avoidance constraints for vehicles with the same entry lane, the same exit lane and traveling along the same path throughout the intersection. The optimal control program is formulated in the spatial domain where the vehicle speed is replaced by its inverse.

The suggested model predictive controller can handle any type of intersection consisting of intersection legs and a physical area where the vehicles can go straight, turn left and/or turn right. This includes merging and splitting of roads. It can further handle multiple vehicles traveling in the same lane, sudden lane blockages.

The controller is shown to give smooth and collision free solutions in a number of scenarios and with different objective functions. It is also shown that the formulation of the optimal control program only contains collision avoidance constraint between vehicles that are at risk of collision, instead of between all vehicles entering the physical area. It is shown that since the optimal control program only implies collision avoidance constraints between vehicles that are at risk of collision, instead of between all vehicles entering the physical area, the throughput is increased.

References

[1] L. Greer, J. L. Fraser, D. Hicks, M. Mercer, K. Thompson *et al.*, "Intelligent transportation systems benefits, costs, and lessons learned: 2018 update report," United States. Dept. of Transportation. ITS Joint Program Office, Tech. Rep., 2018.

[2] M. R. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.

[3] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

[4] M. Khayatian, M. Mehrabian, E. Andert, R. Dedinsky, S. Choudhary, Y. Lou, and A. Shirvastava, "A survey on intersection management of connected autonomous vehicles," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, pp. 1–27, 2020.

[5] M. C. Simon, T. Hermitte, and Y. Page, "Intersection road accident causation: A European view," in *International Technical Conference on the Enhanced Safety of Vehicles*, 2009.

[6] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," in *National Traffic Highway Safety Association*, 2010.

[7] H. Ahn and D. D. Vecchio, "Safety verification and control for collision avoidance at road intersections," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 630–642, 2018.

[8] 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," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 408–10 420, 2019.

[9] Y.-T. Lin, H. Hsu, S.-C. Lin, C.-W. Lin, I. H.-R. Jiang, and C. Liu, "Graph-based modeling, scheduling, and verification for intersection management of intelligent vehicles," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–21, 2019.

[10] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, 2011.

[11] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modelling of conflict resolution at traffic intersections," in *Decision and Control (CDC)*, Osaka, Japan, 2015.

[12] L. Riegger, M. Carlander, N. Lidander, N. Murgovski, and J. Sjöberg, "Centralized MPC for autonomous intersection crossing," in *ITSC*, Osaka, Japan, 2016.

[13] J. Karlsson, J. Sjöberg, N. Murgovski, L. Hanning, V. Olsson, S. Luu, and A. Rasch, "Intersection crossing with reduced number of conflicts," in *ITSC*, Hawaii, USA, 2018.

[14] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," in *IFAC*, Toulouse, France, 2017.

[15] G. R. de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, 2017.

[16] X. Zhao, J. Wang, G. Yin, and K. Zhang, "Cooperative driving for connected and automated vehicles at non-signalized intersection based on model predictive control," in *ITSC*, Auckland, New Zealand, 2019.

[17] E. F. Camacho and C. Bordons, *Model predictive control*. Springer Science & Business Media, 2013.

[18] R. Hult, M. Zanon, S. Gros, and P. Falcone, "An MIQP-based heuristic for optimal coordination of vehicles at intersections," in *Decision and Control (CDC)*, Miami Beach, USA, 2018.

[19] S. Huang, A. W. Sadek, and Y. Zhao, "Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1201–1214, 2012.

[20] M. Ma and Z. Li, "A time-independent trajectory optimization approach for connected and autonomous vehicles under reservation-based intersection control," *Transportation Research Interdisciplinary Perspectives*, vol. 9, 2021.

[21] Massachusetts Highway department, "Project development and design guide," 2006.

[22] American Association of State Highway and Transportation officials (AASHTO), *A policy on geometric design of highways and streets*, 6th ed., 2011.

[23] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. Springer, 2000.

[24] S. Boyd and L. Vanderberghe, *Convex optimization*, 1st ed. Cambridge University Press, 2004.

[25] R. Hult, "Optimization-based coordination strategies for connected and autonmous vehicles," Ph.D. dissertation, Chalmers university of technology, 2019.

[26] J. Karlsson, N. Murgovski, and J. Sjöberg, "Computationally efficient autonomous overtaking on highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3169 – 3183, 2020.

Jonas Sjöberg received the degree of master in applied physics from Uppsala University 1989 and the degree of doctor in engineering (PhD) in 1995 from Linköping University, Sweden. After Post-Docs at ETH Zurich, he became Assistant Professor at Chalmers, and after research visits at TU Wien, and at Technion in Haifa, he became a Full Professor at Chalmers University of Technology in 2001. His research interests are in mechatronics, and mechatronic related fields, such as signal processing, and control. Within these fields, interest focus on model based methods, simulations, system identification, and optimization for design and product development of mechatronic systems. Many of the applied projects target the transport area, both electro-mobility, and, active safety and autonomous driving. He was the winner of Volvo Cars technology award 2011. In 2015 he was co-main chair of the FASTzero symposium, and 2016 he is main chair of IEEE Intelligent Vehicles Symposium. Since 2017 he is BOG member of IEEE ITSS.

Nikolce Murgovski is an Associate Professor with the Department of Electrical Engineering, Division of Systems and Control, Chalmers University of Technology. He received the M.S. degree in Software Engineering from University West, Trollhättan, Sweden, in 2007, and the M.S. degree in Applied Physics and the PhD degree in

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3336533

Systems and Control from the Chalmers University of Technology, Gothenburg, Sweden, in 2007 and 2012, respectively. His research interests are in optimization, optimal control, modelling, online learning and estimation. His typical research projects are within electromobility, autonomous driving and automotive active safety.

Johan Karlsson received the M.sc degree in mathematical engineering from Chalmers University of Technology, Gothenburg, Sweden in 2016. He is currently a PhD with the Department of electrical engineering, Chalmers University. His areas of interest include optimization and optimal control in the automotive area.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/