2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Dynamic Trajectory Planning for Emergency Vehicle Clearance at Traffic Intersections Using Model Predictive Control

Mohammad Al Khatib and Naim Bajcinca
Department of Mechanical and Process Engineering
RPTU Kaiserslautern
{mohammad.alkhatib, naim.bajcinca}@rptu.de

***Abstract*— This paper presents an innovative approach for optimizing the clearance of emergency vehicles at traffic intersections by employing Model Predictive Control (MPC) in conjunction with reference trajectory generation. The algorithm operates in two distinct phases: offline static map generation and online dynamic trajectory planning. In the offline phase, the algorithm constructs a static map of the intersection, approximating the drivable area with a polytope covering. In the online phase, the algorithm continuously gathers real-time data on the positions of all vehicles present at the intersection. Based on mixed-integer programming techniques, our algorithm dynamically generates reference trajectories for each vehicle, including the emergency vehicle to facilitate the fastest possible passage for the emergency vehicle to its target location while ensuring the safe clearance of the path ahead. We demonstrate the feasibility and effectiveness of our model predictive control-based algorithm in enhancing the response time of emergency vehicles and minimizing intersection congestion, ultimately contributing to the improvement of urban safety and emergency response services.**

***Index Terms*—Centralized MPC, Traffic control, Autonomous driving, Trajectory planning.**

I. Introduction

In recent years, the rapid advancements in autonomous mobile systems, particularly autonomous vehicles (AV), increased the challenges that accompany this progress [1]. In fact AV can dramatically reduce the frequency of accidents caused by conventional driving, where more than 5.3 million automobile crashes in the United States in 2011, resulting in more than 2.2 million injuries and $32,000$ fatalities, as well as billions of dollars in private and social costs [2].

One of the significant challenges pertains to the management of road scenarios involving emergency vehicles. Unlike regular driving situations, emergency vehicle scenarios require vehicles to deviate from standard traffic rules while ensuring the safety of all individuals involved. In traffic accidents and disasters, every second counts. A quickly and correctly formed emergency lane can have a life-saving effect. Rescue service associations estimate that if emergency services arrive four minutes earlier, the chances of survival are increased by up to 40 percent. Further more, according to statistics, an accident during an emergency response is 17 times more likely to occur compared to regular driving without special privileges (8 times higher with injuries, 4 times higher with fatalities). In Germany alone [3], for example, police vehicles cause up to $13,000$ accidents with significant personal and property damage in a single year. However, a correctly and timely formed emergency lane is rare and difficult to implement without the foresight and prudent actions of all road users. Many drivers lack an overview of the situation of all the traffic around them, which is why they often fail to react correctly. Addressing such challenges and developing autonomous methods to navigate these traffic situations is imperative. The urgency of these situations demands swift computation of solutions, especially in scenarios involving a substantial number of vehicles, such as traffic jams.

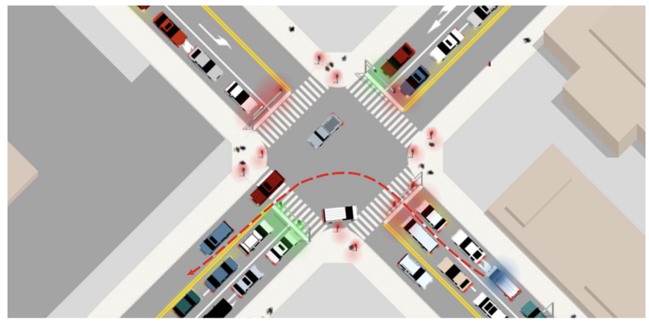


Fig. 1: To ensure a prompt and safe emergency response at such an urban intersection, the traffic control center must implement automated driving maneuvers.

In this paper, we present our innovative approach, the Model Predictive Dynamic Trajectory Planner (MPDTP), designed to efficiently tackle the emergency vehicle clearance at traffic intersections problem as shown in Figure 1 and purposefully manipulate the collective behavior of the traffic in order to facilitate safe and fast interventions. In this problem a set of cars is given together with the initial and target position of an emergency vehicle at a given intersection. The objective is to find a set of collision-free (valid) paths for all agents and for the emergency vehicle that adhere to real-world constraints and lead the emergency vehicle to its respective goal. The proposed MPDTP algorithm addresses the challenges associated with the problem of emergency vehicle clearance at traffic intersections, offering a novel approach that yields optimal solutions, ensuring the

This work was completed in the scope of the research project Aorta funded by the Federal Ministry of Transport and Digital Infrastructure (BMDV)

safety of all individuals involved in emergency scenarios. Our approach is based on the centralized model-predictive control (MPC) strategy that can generate optimal trajectories online for all cars and the emergency vehicle while guaranteeing the satisfaction of state/input constraints and coupled collision-free constraints. At each discrete time the MPDTP algorithm solves an optimization control problem (OCP) and sends paths to each vehicle that are safe over a given time horizon. The vehicles have then to follow the given paths until the algorithm sends them updated paths in the next time instant.

Existing literature can solve the aforementioned problem in the context of motion planning for autonomous driving. A recent survey [4] discusses such search-based methods which rely on abstracting the continuous state-space in the OCP into a graph and search for a solution there. Core search algorithms include A* search [4] and dynamic programming (DP) [5]. Variants of the A* algorithm are also developed in this context such as Hybrid A* [6] and Semi-optimization A* [7]. Thus, in comparison to existing methods, the main contribution of this paper is the development of an MPC based planner to solve the emergency vehicle clearance at traffic intersections problem in the continuous state-space, resulting in less conservative movement and faster transition times.

Various related works to solve our problem in the continuous state-space using MPC exists as well. In [8] the authors present a model predictive control algorithm to generate trajectories in real-time for multiple robots. However non-convex state-space constraints are not tackled and collision avoidance constraints are rather quadratic. In [9] the authors use nonlinear constraints to exactly reformulate collision avoidance constraints using strong duality of convex optimization. In [10] the authors propose tube-based nonlinear model predictive control for a class of nonlinear multiagent systems in the context of navigation in a multiagent setting. The same problem is solved for linear systems in [10]. Nonetheless, [8] and [10] do not consider coupled nonconvex constraints to ensure collision avoidance , as tackled in this paper, and the nonlinear optimization problem formulated in [9] requires large computational burden. In addition, unlike the approach in [11] which solves the multi-agent motion planning problem using an optimization-based method, the collision avoidance approach therein is conservative when no geometry of the agents is taken into consideration. In our approach we consider the polytopic nature of the vehicles formulate the problem into a mixed integer quadratic program.

The rest of the paper is organized as follows: In Section II, the emergency vehicle clearance at traffic intersections problem is formulated. In section III, we reformulate the problem as a centralized moving horizon optimization problem and explain the terms introduced by the cost function and constraints. Finally, Section IV evaluates the proposed algorithms before concluding our work.

Notations: We let $\mathbb{R}$, $\mathbb{R}_{0+}$, $\mathbb{R}_{+}$, $\mathbb{N}$ and $\mathbb{N}_{+}$ denote the sets of reals, non-negative reals, positive reals, non-negative integers, and positive integers, respectively. For $I \subseteq \mathbb{R}_{0+}$, let $\mathbb{N}_I = \mathbb{N} \cap I$. Given a set S that is a subset of $\mathbb{R}^n$ and a real matrix A of size $n \times n$, we define the set AS as the set of all vectors $x \in \mathbb{R}^n$ such that there exists a vector y in S satisfying $x = Ay$. Furthermore, for a scalar a in the set of real numbers ($\mathbb{R}$), we define aS as the set obtained by scaling all elements of S by a. Here, aI_n represents the scalar multiplication of the $n \times n$ identity matrix I_n by a. The interior of a set S is denoted by $\mathrm{int}(S)$, while the convex hull of S is denoted by $\mathrm{ch}(S)$. We define the Minkowski sum of sets S and S' as the set of all vectors $x + x'$, where x belongs to S and x' belongs to S'. Symbolically, $S \oplus S' = \{x + x' : x \in S, x' \in S'\}$. We represent the set of compact subsets of $\mathbb{R}^n$ as $\mathcal{K}(\mathbb{R}^n)$ and the set of compact subsets of $\mathbb{R}^n$ containing the point 0 in their interior as $\mathcal{K}_0(\mathbb{R}^n)$. For a vector $K \in \mathbb{R}^l$ and matrix $H \in \mathbb{R}^{l\times n}$ having row vectors $\mathrm{r}_1(H), \ldots, \mathrm{r}_l(H)$ and satisfying $0 \in \mathrm{int}(\mathrm{ch}(\{\mathrm{r}_1(H), \ldots, \mathrm{r}_l(H)\}))$ we denote by $\mathcal{P}(H, K)$ the polytope $\{x \in \mathbb{R}^n : Hx \leq K\}$ and by $\mathrm{vert}(\mathcal{P}(H, K))$ its set of vertices. Finally for a given vector $x \in \mathbb{R}^n$ we denote by $\max(x)$ the maximum element of x and by x_i its i-th element.

II. Problem formulation

In this section we formulate the problem of emergency vehicle clearance at traffic intersection which is similar to the well known multi-agent motion planning problem in the literature. The latter is a large and active research area which mainly relies on decoupling the problem into two sub-problems. One which handles the complex nonlinear dynamics of separate vehicles by a low-level tracking controller. The other problem deals with the complexity of the overall specification that the vehicles must satisfy together through a global planner. In our setting we assume that a tracking controller is given for each agent such that it can track the reference path generated by our planner.

Now we consider a traffic intersection $\Phi(\phi_1, \ldots, \phi_M)$ as a covering defined by

$$\Phi(\phi_1, \ldots, \phi_M) = \bigcup_{r=1}^{M} \phi_r \tag{1}$$

for polytopes $\phi_r = \mathcal{P}(H_r, 1)$ and given matrices $H_r \in \mathbb{R}^{l_r \times n}$, $r \in \mathbb{N}_{[1,M]}$. At the intersection we have $N + 1$ vehicles one of which is an emergency vehicle $\mathcal{V}^{[0]} = \mathcal{P}(V^{[0]}, 1)$ and the others are given by $\mathcal{V}^{[i]} = \mathcal{P}(V^{[i]}, 1)$, for given matrices $V^{[i]} \in \mathbb{R}^{p_i \times n}$, $i \in \mathbb{N}_{[0,N]}$. We note that based on the context, we interchangeably use the notation $\mathcal{V}$ to denote the polytope representation of the vehicle or just to refer to the vehicle itself. A simple linear time-invariant model is considered in discrete-time to model the dynamics of each vehicle as:

$$x^{[i]}(t_{k+1}) = A^{[i]}x^{[i]}(t_k) + B^{[i]}u^{[i]}(t_k) \quad k \in \mathbb{N} \tag{2}$$

where $x^{[i]}(t_k) \in \mathbb{R}^n$ is the state of an individual vehicle $\mathcal{V}^{[i]}$, $u^{[i]}(t) \in \mathbb{R}^m$ is its control input, $x^{[i]}(0) = x_0^{[i]}$ is its initial state, and the matrices $A^{[i]}, B^{[i]}$ are of appropriate dimensions. We denote the state and control input of the whole system to be the concatenations $x = [x^{[0]}; \ldots ; x^{[N]}]$ and $u = [u^{[0]}; \ldots ; u^{[N]}]$. Thus the dynamics of all vehicles could be written in compact form as $x(t_{k+1}) = Ax(t_k) + Bu(t_k)$ with appropriate matrices A and B and initial condition $x_0 = [x_0^{[0]}; \ldots ; x_0^{[N]}]$. To this end we can define the planning problem as follows.

Problem 1: Given $\Phi(\phi_1, \ldots, \phi_M)$, vehicles $\mathcal{V}^{[0]}, \ldots, \mathcal{V}^{[N]}$, initial positions $x_0^{[0]}, \ldots, x_0^{[N]}$, set $\mathcal{U} = \mathcal{U}^{[0]} \times \cdots \times \mathcal{U}^{[N]}$, and a terminal position $x_f \in \mathbb{R}^n$. Find a set of trajectories $x^{[0]}(\cdot), \ldots, x^{[N]}(\cdot)$ satisfying the dynamics in (2) such that

- $x^{[i]}(0) = x_0^{[i]}$, $i \in \mathbb{N}_{[0,N]}$,
- $u^{[i]}(t) \in \mathcal{U}^{[i]}$, $\forall t, i \in \mathbb{N}_{[0,N]}$,
- vehicle $\mathcal{V}^{[0]}$ reaches x_f in finite time; there exists $t_f < \infty$ with $x^{[0]}(t_f) = x_f$, and
- all vehicles stay in their drivable areas, with
- no inter-vehicle collisions; i.e. $\forall t, \forall i, j \in \mathbb{N}_{[0,N]}, i \neq j$, $\left(\{x^{[i]}(t)\} \oplus \mathcal{V}^{[i]}\right) \bigcap \left(\{x^{[j]}(t)\} \oplus \mathcal{V}^{[j]}\right) = \emptyset$.

Without loss of generality we assume that all vehicles $\mathcal{V}^{[i]}$, $i \in \mathbb{N}_{[1,N]}$, could interfere with the emergency vehicle and thus the drivable area, which is assumed to be covered by M_p sets $\Phi(\phi_1, \ldots, \phi_{M_p})$, for all considered vehicles is the same. In the general case, the set of vehicles at an intersection is partitioned and only those that can interfere with the emergency vehicle are considered in our algorithm. In the next section, we reformulate Problem 1 as a mixed integer quadratic optimization problem. Then within the model-predictive control paradigm we solve the planning problem online and generate in real-time collision free optimal trajectories for each vehicle so that the optimal path of the emergency vehicle to the given target location is safely cleared.

III. MPC FOR TRAJECTORY PLANNING

In this section, we propose a solution for Problem 1. Given $N+1$ vehicles on an intersection $\Phi(\cdot)$ we propose an MPC which is responsible to generate safe trajectories for all vehicles so that the emergency vehicle reaches as soon as possible a given target location while the other cars clear the way quickly while staying in the allowed drivable area. Figure 2 shows the structure of the proposed algorithm.

The output of our algorithm will be trajectories $x^{[i]}(\cdot)$ defined in discrete time $t_{k+1} = t_k + T$, $k \in \mathbb{N}$, where T is the sampling time of the algorithm. Then in continuous-time a piece-wise linear path will be generated, and sent to the low level tracking controller, defined as

$$S^{[i]}(s) = x^{[i]}(t_k) + \frac{x^{[i]}(t_{k+1}) - x^{[i]}(t_k)}{T}(t - t_k), \qquad (3)$$

for $s \in [t_k, t_{k+1}]$.

We proceed to the next main section where we present useful derivations first before presenting the algorithm afterwards.

A. MPC-based planner design

We reformulate Problem 1 as a constrained mixed-integer quadratic moving horizon optimal control problem which

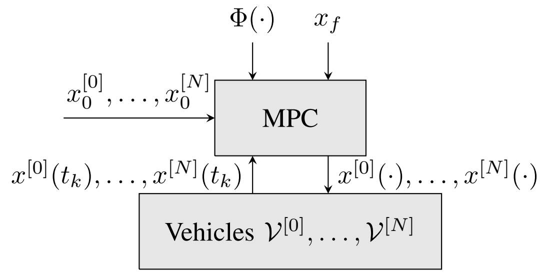


Fig. 2: The proposed MPC planning strategy. Here low level controllers which are out of the scope of this work are assumed to be embedded in the vehicles.

needs to be solved online at each time step $t_k = kT$. The optimization problem has the following form:

$$\min_{x,u} \sum_{k=1}^{L} J_k$$

$$\begin{aligned}
\text{s.t.}\quad & x(t_{k+1}) = Ax(t_k) + Bu(t_k), \qquad \forall k \in \mathbb{N}_{[1,L]} && \text{(4a)}\\
& u^{[i]}(t_k) \in \mathcal{U}^{[i]}, \qquad \forall i \in \mathbb{N}_{[0,N]}, k \in \mathbb{N}_{[1,L]} && \text{(4b)}\\
& \{x^{[i]}(t_k)\} \oplus \mathcal{V}^{[i]} \subseteq \Phi(\phi_1, \ldots, \phi_{Mp}), && \text{(4c)}\\
& \qquad\qquad \forall i \in \mathbb{N}_{[0,N]}, k \in \mathbb{N}_{[1,L]} &&\\
& x(t_0) = x_0, && \text{(4d)}\\
& \mathcal{C}(x(t_k)) = \emptyset, \forall k \in \mathbb{N}_{[1,L]} && \text{(4e)}
\end{aligned}$$

Scalar $L \in \mathbb{N}$ is the prediction horizon. Sets $\mathcal{U}^{[i]} \subset \mathbb{R}^m$, $i \in \mathbb{N}_{[0,N]}$ are assumed to be given closed, bounded, and convex sets. The objective, is to minimize the distance of the emergency vehicle at each time step to the target state. Thus the resulting cost function of the optimization problem is given by:

$$J_k = \left(\|x^{[0]}(t_k) - x_f\|_Q^2 + \sum_{i=0}^{N} \left(\|u^{[i]}(t_k)\|_R^2\right)\right), \qquad (5)$$

where matrices $Q, R > 0$ are the state and input weighting matrices respectively. By such an approach any vehicle crossing the optimal trajectory of the emergency vehicle within the prediction horizon will be forced to move away as needed. But since inter-vehicle collisions are prohibited by constraint (4e) then other cars will be also moved allowing for such clearance. Indeed (4e) is given as well by

$$\mathcal{C}(x(t_k)) = \left(\{x^{[i]}(t_k)\} \oplus \mathcal{V}^{[i]}\right) \bigcap \left(\{x^{[j]}(t_k)\} \oplus \mathcal{V}^{[j]}\right), \qquad (6)$$

for all $j \in \mathbb{N}_{[i+1,N]}$, $i \in \mathbb{N}_{[0,N-1]}$, and guarantees also that the emergency vehicle will not collide with other vehicles in its vicinity. Constraint (4a) embed the dynamics within the generated trajectories. (4b)-(4c) define constraints on the vehicle locations and speeds where vehicles at time t_k, $\mathcal{P}(V^{[i]}, V^{[i]}x^{[i]}(t_k) - 1) = \{x^{[i]}(t_k)\} \oplus \mathcal{V}^{[i]}$ needs to be in the drivable area with acceptable speeds. (4d) defines the initial state for the system.

Now we simplify further, constraints (4c) and (4e). As for constraint (4c) it requires every vehicle $\mathcal{V}^{[i]}$, $i \in \mathbb{N}_{[0,N]}$ to be within the drivable area

$$\Phi(\phi_1,\ldots,\phi_{M_p}) = \bigcup_{r=1}^{M_p} \phi_r. \tag{7}$$

(4c) follows from the logical statement:

$$\bigvee_{r=1}^{M_p} \Big(\big(\{x^{[i]}(t_k)\} \oplus \mathcal{V}^{[i]} \big) \subseteq \phi_r \Big) = 1. \tag{8}$$

In other words, vehicle $\mathcal{V}^{[i]}$ needs to be at least in one of polytopes ϕ_r, $r \in \mathbb{N}_{[1,M_p]}$ defining the coverage of the drivable area. Consequently, logical statement (8) as well as constraint (4c) could be then rewritten using additional binary variables as in the following result:

Proposition 1: Constraint (4c) is satisfied if there exist binary variables $\delta_1^{[i]}(t_k),\ldots,\delta_{M_p}^{[i]}(t_k)$ such that

$$\sum_{r=1}^{M_p} \delta_r^{[i]}(t_k) = 1 \tag{9}$$

and

$$\sum_{r=1}^{M_p} \delta_r^{[i]}(t_k) H_r x^{[i]}(t_k) \leq 1 - b_r^{[i]} 1. \tag{10}$$

where the offline computed constants $b_r^{[i]} = \max_{x\in \text{vert}(\mathcal{V}^{[i]})}\{\max(H_r x)\}$.

Next, we reformulate in a similar reasoning constraint (4e) to involve binary variables as well. To this end we need to add $\frac{N(N+1)}{2}$ constraints at every discrete-time t_k having the form

$$\mathcal{P}(V^{[i]}, K^{[i]}(t_k))) \cap \mathcal{P}(V^{[j]}, K^{[j]}(t_k))) = \emptyset, \tag{11}$$

with $K^{[i]}(t_k) = V^{[i]}x^{[i]}(t_k) - 1$. (11) is equivalent to the logical statement:

$$\bigvee_{l=1}^{p_j} \Big(\mathrm{r}_l(V^{[j]})(x^{[i]}(t_k) - x^{[j]}(t_k)) \Big) \geq K^{[j]}(t_k) + (a_j^{[i]} + a_i^{[j]})1, \tag{12}$$

where the offline computed constants $a_j^{[i]} = \max_{x\in \text{vert}(\mathcal{V}^{[i]})}\{\max(V^{[j]}x)\}$. In other words, (12) enforces that at least one of the half-space inequalities is violated so that $x^{[i]}(t_k)$ lies with a certain distance outside $\mathcal{P}(V^{[j]}, K^{[j]}(t_k)))$. Consequently logical statement (12) as well as constraint (4e) are enforced using the following result:

Proposition 2: (12) is satisfied if there exist binary variables $\Delta_{l,j}^{[i]}(t_k)$, $l \in \mathbb{N}_{[1,p_j]}$,$j \in \mathbb{N}_{[i+1,N]}$, $i \in \mathbb{N}_{[0,N-1]}$, such that:

$$\sum_{l=1}^{l_j} \Delta_{l,j}^{[i]}(t_k) = 1 \tag{13}$$

and

$$\sum_{l=1}^{l_j} \Delta_{l,j}^{[i]}(t_k) \Big(\mathrm{r}_l(V^{[j]})(x^{[i]}(t_k) - x^{[j]}(t_k)) \Big) \geq K^{[j]}(t_k) + (a_j^{[i]} + a_i^{[j]})1. \tag{14}$$

B. Algorithm

Given $\Phi(\phi_1,\ldots,\phi_M)$, vehicles $\mathcal{V}^{[0]},\ldots,\mathcal{V}^{[N]}$, initial positions $x_0^{[0]},\ldots,x_0^{[N]}$, and a terminal position $x_f \in \mathbb{R}^n$ we follow the derivations in Section III-A, and solve Problem 1 by iteratively solving at discrete times t_k, as in Algorithm 1, the following OCP:

$$\begin{aligned}
\min_{x,u,\delta,\Delta} \; & \sum_{k=1}^{L} \Big(\|x^{[0]}(t_k) - x_f\|_Q^2 + \sum_{i=0}^{N} \big(\|u^{[i]}(t_k)\|_R^2 \big) \Big) \\
\text{s.t.} \quad & x(t_{k+1}) = Ax(t_k) + Bu(t_k), \quad \forall k \in \mathbb{N}_{[1,L]} && \text{(15a)}\\
& u(t_k) \in \mathcal{U}, \qquad \forall k \in \mathbb{N}_{[1,L]} && \text{(15b)}\\
& Eqs.(9)\&(10), \qquad \forall k \in \mathbb{N}_{[1,L]} && \text{(15c)}\\
& x(t_0) = x_0, && \text{(15d)}\\
& Eqs.(13)\&(14), \qquad \forall k \in \mathbb{N}_{[1,L]} && \text{(15e)}
\end{aligned}$$

OCP (15) is a mixed-integer quadratic optimization problem that we solve online using available solvers such as Gurobi [12].

Algorithm 1 MPDTP algorithm

function MPDTP($\Phi(\cdot)$, x_0, x_f, $\mathcal{U}$, $\mathcal{V}^{[0]}$, …, $\mathcal{V}^{[N]}$, L)
1 : Set $k = 0$ and $x(t_k) := x_0$
2 : $x(\cdot) := \mathcal{C}(\Phi(\cdot), x(t_k), x_f, \mathcal{U}, \mathcal{V}^{[0]}, \ldots, \mathcal{V}^{[N]}, L)$
3 : Compute $u_{\text{low}}(\cdot) := \kappa(x(\cdot))$
4 : Apply the control input $u_{\text{low}}(t)$ for $t \in [t_k, t_{k+1}]$
5 : Measure $x(t_{k+1})$,
6 : Increment time index $k := k + 1$ and return to step 2

In the first line of Algorithm 1 we initialize our computations. In line 2, we solve online the optimal control problem (15) for an initial condition $x(t_k)$ to find the vehicles' individual trajectories $(x^{[i]}(t))_{t\in\mathbb{N}_{[t_k,t_{k+L}]}}$, $i \in \mathbb{N}_{[0,N]}$. The control input in continuous-time $t \in [t_k, t_{k+1}]$ is then defined in line 3, using a low level controller (as explained in Section II) for tracking the generated trajectories in the previous step, and applied to the system as in line 4. In line 5-6 we measure the state at time t_{k+1} shift the horizon one step and recompute iteratively the optimal trajectories as in line 2.

Remark 3: We note that OCP (15) could be further generalized to handle signal-temporal logic specifications as in [11], [13]–[15] by translating such specifications into constraints with additional binary variables. Furthermore, in the context of multi-agent motion planning our approach could be easily extended to plan trajectories for multiple agents to reach their individual targets optimally by appropriately changing the cost function to include a final destination for each vehicle.

IV. Case Studies

We demonstrate the functionality of Algorithm 1 at the micromanagement levels, through a simple scenario at an inner-city intersection of a multi-lane road with traffic lights, an emergency vehicle, and two closely spaced vehicles. This scenario takes advantage of the presence of communication and computing infrastructure in the form of Edge Computing Devices (ECD), as well as controllable traffic lights where once the emergency vehicle approaches the traffic intersection it sends a signal to the ECD through vehicle to infrastructure (V2I) communication channels including its position and target location. Then the ECD runs Algorithm 1 iteratively and sends online updated trajectories to all vehicles in the intersection where once low level controllers in the vehicles follow the assigned safe and optimal trajectories the path is cleared in front of the emergency vehicle until it reaches its target location.

All simulations run, using Matlab software, in real-time on machine having a 1.4 GHz i5 processor with 8 GB memory.

The intersection we consider is on the Trippstadt Street at Max Planck Institute in Kaiserslautern (Germany) as depicted in Figure 3. A reconstruction of the intersection in 2D is done and plotted in Figure 4. After that we under-approximate tightly the drivable area there in, as shown in Figure 5, using 14 polytopes $\phi_1, \ldots, \phi_{14}$ and define the set

$$\Phi(\phi_1, \ldots, \phi_{14}) = \bigcup_{r=1}^{14} \phi_r.$$

After that we set the emergency vehicle $\mathcal{V}^{[0]}$ in our controller to be a rectangle of 3 meter length and 1 meter width. The other 2 vehicles $\mathcal{V}^{[1]}$ and $\mathcal{V}^{[2]}$ are identical cars of 0.8 meter width and 1.5 meters length. All matrices in (2) are the same for the vehicles and are given by matrix A and B with:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \tag{16}$$

where the time step is given by $T = 0.1sec$. Initial positions are given by

$$x_0 = [-140; 572; -160; 578; -163; 581].$$

The control inputs for the emergency vehicle are required to be bounded in the set

$$\mathcal{U}^{[0]} = [-17, 17] \times [-17, 17].$$

The other vehicles' speeds are required to be more constrained in emergency situation where we set $\mathcal{U}^{[1]} = \mathcal{U}^{[2]} = 0.6\mathcal{U}^{[0]}$. The target location for the emergency vehicle is set to $x_f = [-195; 520]$. After setting the prediction horizon to be $L = 20$ which corresponds to a duration $LT = 2sec$ we follow the steps of Algorithm 1 and simulate the intersection for a total duration of $5sec$. We note here that just for simulation purposes we set the low level control inputs the same as the control inputs $u^\star(t_k), \ldots, u^\star(t_{k+L})$ generated by controller $\mathcal{C}(\cdot)$ when solving OCP (15). We also feedback these control inputs in line 4 of Algorithm 1 to system (2). And thus in line 5 of the algorithm we have

$$x(t_{k+1}) = f(x(t_k), u^\star(t_k)),$$

as the measured state. As Figures 6 and 7 show, the emergency vehicle drives safely toward the target location and reaches destination in $5sec$. As for the other cars we noticed that they started clearing the path, without colliding with each other or going out of the drivable area, from the very beginning so that the emergency vehicle can still drive at maximum allowed speed. After the emergency vehicle passes next to the other vehicles, the latter almost stop moving since they are no more in the former's path.

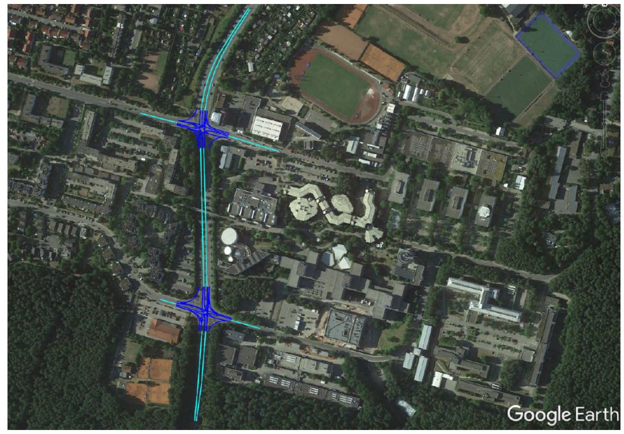


Fig. 3: A bird's-eye view of the Trippstadt street section at the Max-Planck-Institut at Kaiserslautern.

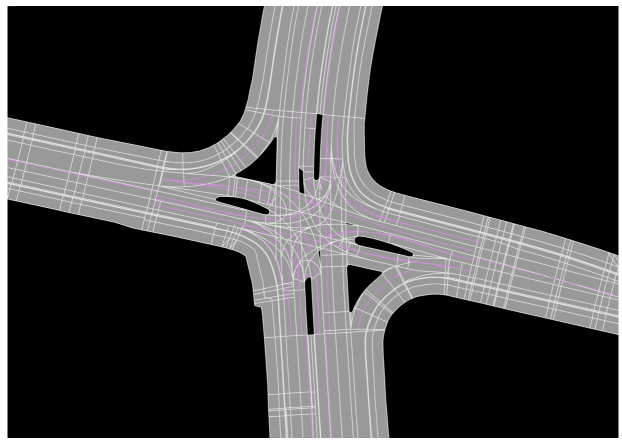


Fig. 4: A 2D reconstruction of the intersection at the Max-Planck-Institut. It is clear that 6 islands are presented which are not drivable, increasing the number of polytopes to cover a tight under-approximation of the drivable area.

V. Conclusion

In this paper, we propose a centralized MPC-based planning algorithm to solve dynamically the problem of emergency vehicle clearance at a given intersection. Using off-the-shelf optimizers, we show that the proposed controller is able to generate online optimal trajectories for individual vehicles that are safe and allow the emergency vehicle to reach its target location in optimal time. Ongoing research related to this topic include the design of distributed/decentralized controllers for the clearance problem. Additionally, we aim to demonstrate the performance of the algorithms on real autonomous cars equipped with low-level real-time controllers that can track the generated reference trajectories by our MPDTP algorithm with high precision and good performance.

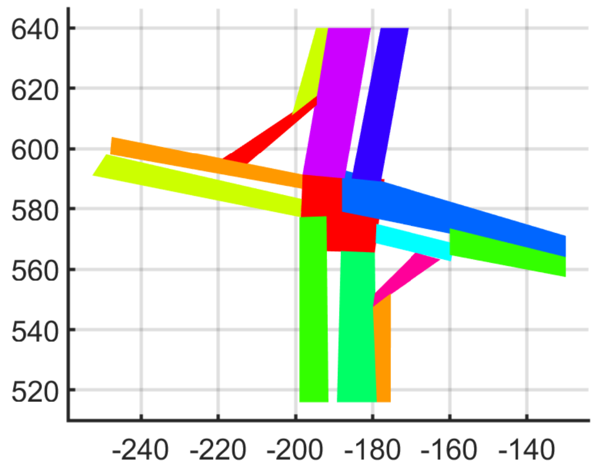


Fig. 5: A polytopic covering $\Phi(\phi_1, \ldots, \phi_{14})$ of the drivable area in the traffic intersection. It is clear that islands are excluded from the cover.

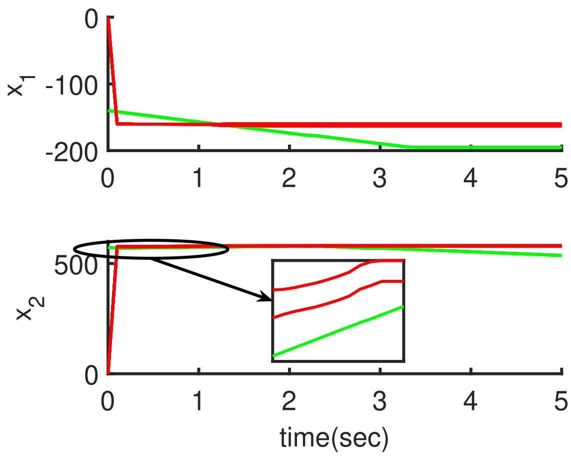
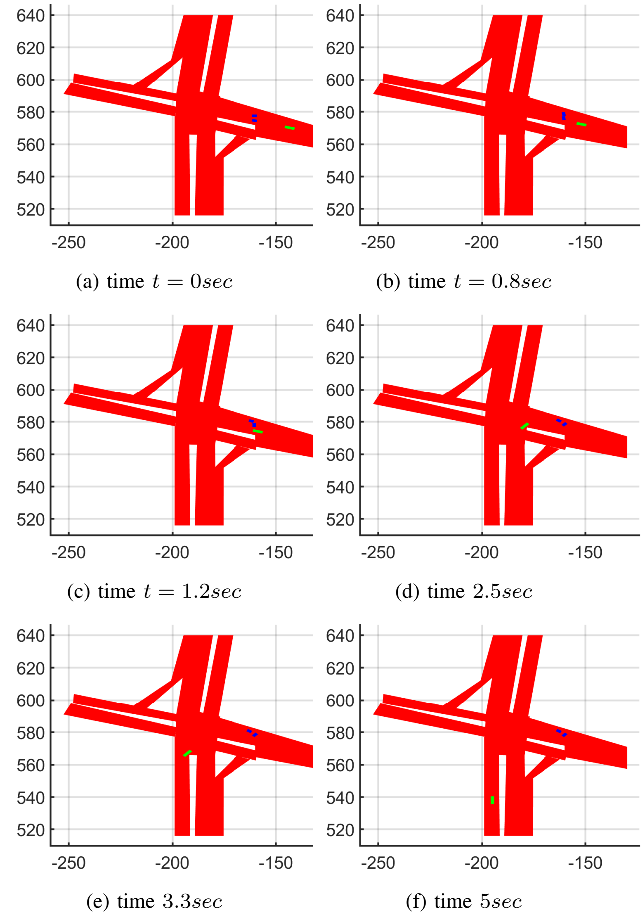


Fig. 6: State trajectories for the emergency vehicle (green) and the two other vehicles (red).



(a) time $t = 0sec$ (b) time $t = 0.8sec$

(c) time $t = 1.2sec$ (d) time $2.5sec$

(e) time $3.3sec$ (f) time $5sec$

Fig. 7: Plotting the emergency vehicle (green) and the other vehicles (blue) at different time instances while running the MPDTP algorithm.

REFERENCES

[1] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of modern transportation*, vol. 24, pp. 284–303, 2016.

[2] J. M. Anderson, K. Nidhi, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014.

[3] "Road safety for emergency drivers," unfallkasse-nrw.de/fileadmin/server/download/verbandszeitschrift/Mit_Blaulicht_und_Martinshorn.pdf.

[4] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu *et al.*, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, 2023.

[5] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[6] ——, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[7] S. M. Bagheri, H. Taghaddos, A. Mousaei, F. Shahnavaz, and U. Hermann, "An a-star algorithm for semi-optimization of crane location and configuration in modular construction," *Automation in Construction*, vol. 121, p. 103447, 2021.

[8] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.

[9] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[10] B. Hernandez and P. Trodden, "Distributed model predictive control using a chain of tubes," in *2016 UKACC 11th International Conference on Control (CONTROL)*. IEEE, 2016, pp. 1–6.

[11] D. Sun, J. Chen, S. Mitra, and C. Fan, "Multi-agent motion planning from signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3451–3458, 2022.

[12] B. Bixby, "The gurobi optimizer," *Transp. Res. Part B*, vol. 41, no. 2, pp. 159–178, 2007.

[13] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 81–87.

[14] A. T. Buyukkocak, D. Aksaray, and Y. Yazıcıoğlu, "Planning of heterogeneous multi-agent systems under signal temporal logic specifications with integral predicates," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1375–1382, 2021.

[15] Y. E. Sahin, P. Nilsson, and N. Ozay, "Multirobot coordination with counting temporal logics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1189–1206, 2019.