2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Model Predictive Control for Multi-Agent Systems under Limited Communication and Time-Varying Network Topology

Danilo Saccani, Lorenzo Fagiano, Melanie N. Zeilinger and Andrea Carron

Abstract— **In control system networks, reconfiguration of the controller when agents are leaving or joining the network is still an open challenge, in particular when operation constraints that depend on each agent's behavior must be met. Drawing our motivation from mobile robot swarms, in this paper, we address this problem by optimizing individual agent performance while guaranteeing persistent constraint satisfaction in presence of bounded communication range and time-varying network topology. The approach we propose is a model predictive control (MPC) formulation, building on multi-trajectory MPC (mt-MPC) concepts. To enable plug and play operations when the system is in closed-loop without the need of a request, the proposed MPC scheme predicts two different state trajectories in the same finite horizon optimal control problem. One trajectory drives the system to the desired target, assuming that the network topology will not change in the prediction horizon, while the second one ensures constraint satisfaction assuming a worst-case scenario in terms of new agents joining the network in the planning horizon. Recursive feasibility and stability of the closed-loop system during plug and play operations are shown. The approach effectiveness is illustrated with a numerical simulation.**

I. Introduction

The interest in autonomous mobile robots is ever increasing for applications ranging from military technology to self-driving vehicles [1]. In particular, multi-agent motion planning has proven successful due to its relevance for numerous real-life applications, see for example [2]. Among the different approaches for dynamic path planning, optimization-based ones, such as Model Predictive Control (MPC), see [3], have received broad attention thanks to their ability to manage state and input constraints while minimizing multi-objective cost functions. When the communication between multiple agents depends on the agent's state, the generated communication network is time-varying, and at each time step, subsystems can leave or join the network. The problem of efficiently treating agents or nodes joining or leaving a network has been referred to as a Plug and Play (PnP) problem in the literature [4]. In order to give guarantees on stability and constraint satisfaction for the new network topology, available results resort on offline redesign of the local controllers to accept a plug-in request. The problem of automatic PnP is still an open challenge [4] when a subsystem is added or removed without a request. In this paper, we address the problem of autonomously navigating a group of robots to a target while guaranteeing collision avoidance despite plug-in plug-out operations. Each agent is able to communicate with neighbouring robots. Neighbouring subsystems are defined based on the agent's current state, and, during the navigation, the agent's state evolves and shifts its communication capabilities. Thus, the network topology can not be enforced to remain the same, but it is intrinsically time-varying and evolves during navigation. This time-varying nature of the network topology calls for an approach that must be able to tolerate plug-in/out operations without a-priori requests. The presented solution is based on the multi-trajectory MPC concept firstly introduced in [5] and nonlinear tracking MPC proposed in [6], [7]. Specifically, the multi-trajectory formulation is used to balance two potentially conflicting requirements: tracking of the target and safe behaviour in case of network topology changes.

Related work: Numerous applications necessitate enhancing performance without compromising safety. This problem has been addressed in different works, often exploiting optimization in order to satisfy safety constraints. In [8], the authors derived a predictive safety filter to ensure the system's safety while an external, potentially unsafe, learning-based control action optimizes the system's performance. The same concept has been applied to distributed networked systems in [9]. Also control barrier function theory has been investigated to guarantee the system's safety [10]. In [11], a soft-constrained predictive control problem has been used as a recovery mechanism for a safety filter to guarantee the feasibility of the problem. These approaches guarantee the system's safety but rely on an external controller to maximize the performance. Moreover, predictive safety filters are designed to handle uncertainty in system dynamics, but not changing network topologies. To combine performance and safety in a single approach, in [12], the authors proposed the use of multiple trajectories in a trajectory planner where a back-up trajectory is used to ensure safety. In [13] and [14], the approach has been considered in an MPC framework providing theoretical guarantees on robust constraint satisfaction and convergence of the approach. In [15], a similar concept

This research has been supported by the Italian Ministry of University and Research (MIUR) under the PRIN 2017 grant n. 201732RS94 “Systems of Tethered Multicopters”, by the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40_180545) and by Fondazione Cariplo under grant n. 2022-2005, project "NextWind - Advanced control solutions for large scale Airborne Wind Energy Systems" and by the Italian Ministry of Enterprises and Made in Italy under grant "4DDS - 4D Drone Swarms".

D. Saccani is with the Institute of Mechanical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. (email: danilo.saccani@epfl.ch)

L. Fagiano is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy. (email: lorenzo.fagiano@polimi.it)

M. N. Zeilinger and A. Carron are with the Institute for Dynamic Systems and Control, ETH Zurich, Switzerland. (email { carrona, mzeilinger}@ethz.ch)

has been exploited to trade-off the behaviour of a nominal with that of a contingency model to control a self-driving car.

When a network setup is considered, ensuring safety while accounting for network topology changes is important. To this aim, a significant effort has been made to address the PnP problem. In [16], the authors present a transition scheme that prepares the system for the new network topology. The plug-in plug-out requests are elaborated by the network, and if the request can be accepted, a re-design of local controllers is performed. These results have been exploited in [17] to derive a safety filter able to provide safety verification during plug-in plug-out operations when a distributed learning-based control action is applied to the system. Similarly to the approach presented in [16], in [18], an offline re-design of local controllers has been proposed for the plugging-in plugging-out of a subsystem when the network accepts the request. In contrast, the approach proposed in this work designs a safe, feasible trajectory online that can always tolerate possible plug-in plug-out operations deriving from the time-varying network topology.

Contributions: The main contributions of this paper are twofold: the first is a safe control scheme for multi-agent systems ensuring collision avoidance with the current neighbouring agents using multi-trajectory MPC. The second contribution is to enable automatic plug-and-play operations in a time-varying network topology of agents with limited communication capabilities, i.e., only with spatially close neighboring agents, which, unlike other works in the literature, cannot be denied.

II. Problem description

In this section, we first introduce the system setup, discuss the communication model among agents, and the resulting communication network. Finally, we will state the problem we aim to solve.

A. System setup

We consider a group of mobile agents where each agent is identified by an integer $i \in \mathcal{M} = \{1, \dots, N_a\}$ and behaves according to the following discrete-time nonlinear dynamics:

$$\begin{aligned} x_i(k+1) &= f_i(x_i(k), u_i(k)) \\ p_i(k) &= C_i x_i(k), \end{aligned} \tag{1}$$

where $x_i(k) \in \mathbb{R}^{n_i}$ is the state vector, $u_i(k) \in \mathbb{R}^{m_i}$ the input vector, $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i}$, and $C_i \in \mathbb{R}^{3 \times n_i}$ is a matrix that extracts the position $p_i(k) \in \mathbb{R}^3$ of the robot. We assume that each vehicle is able to measure its whole state x_i. We denote with $(\bar{x}_i,\ \bar{u}_i)$ an equilibrium of system (1) and we consider a state reference tracking problem where $r_i = f_i(\bar{x}_{r,i}, \bar{u}_{r,i})$ is the constant state reference of the i-th agent. Furthermore, we assume that $f_i(x_i(k), u_i(k))$ is differentiable at every equilibrium point and the linearized model is controllable. Let us consider, without loss of generality, that $p_i(k)$ is at the top of the state vector $x_i(k)$ and introduce the operator:

$$\phi(p_i) = [p_i^T, 0, \dots, 0]^T \in \mathbb{R}^{n_i}.$$

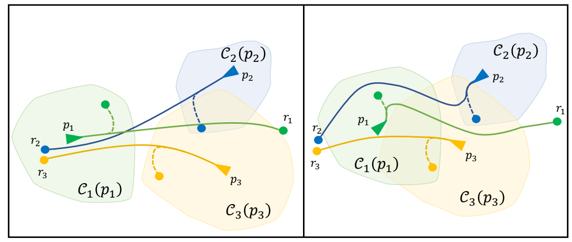

Fig. 1: Plug and play multi-trajectory MPC for three agents whose position is represented by coloured triangles '$\triangle$'. In the left figure, the system is at time instant k, while in the right one, the system is at time $k+1$. $\mathcal{C}_i(p_i)$ are the communication sets and r_i the desired position references. Coloured lines: tracking trajectories; dashed lines: safe trajectories. Coloured dots are the final positions of by the various trajectories.

Assumption 1: The vehicles' dynamics (1) are position invariant, i.e. $\forall p_j \in \mathbb{R}^3$, $x_i(k)$, $u_i(k)$, $f_i(x_i(k) + \phi(p_j), u_i(k)) = x_i(k+1) + \phi(p_j)$.

Remark 1: As an example, Assumption 1 is satisfied when the position is the output of an integrator, which is a typical condition in autonomous vehicles, see e.g., [14], [19].

Finally, each agent is subject to convex time-invariant state and input constraints of the form:

$$(x_i, u_i) \in \mathcal{X}_i \times \mathcal{U}_i,\ \forall i \in \mathcal{M}. \tag{2}$$

B. Communication and network topology

Each agent is equipped with a communication system e.g., an antenna, characterized by a communication set $\mathcal{C}_i(p_i(k)) = [p_i(k) \oplus \mathcal{D}_i]$, where $\mathcal{D}_i \subseteq \mathbb{R}^3$ is a constant compact convex set centered at the origin and $\oplus$ is the Minkowski sum. In some practical applications, the communication set may be originally non-convex, in these cases, one can still take a convex under-approximation of the communication region. Let us consider the following assumption.

Assumption 2: Agent i is able to communicate in a bidirectional way with agent j when $\mathcal{C}_i(p_i(k)) \cap \mathcal{C}_j(p_j(k)) \neq \emptyset$.

The dependence of the communication sets on the system position leads to a time-varying communication topology, including the case where there is no communication among the agents. Such a topology could be described as a time-varying graph composed by connected sub-graphs and null sub-graphs. We denote each of the connected sub-graph as a cluster $\mathcal{G}_m(k) = (\mathcal{V}_m(k), \mathcal{E}_m(k))$ with $m = \{1, \dots, N_c(k)\}$, where the set of nodes $\mathcal{V}_m(k) \in \{1, \dots, N_a\}$ represents the agents in the cluster, and the set of edges $\mathcal{E}_m(k) \subset \mathcal{V}_m(k) \times \mathcal{V}_m(k)$ contains the pairs of agents $\{i, j\}$, which can communicate with each other at time k. At each time step, the agents are grouped in a time-varying number of non-empty clusters $N_c(k)$ and $N_a - N_c(k)$ null sub-graphs. Hence, the sum of cardinalities of the set of nodes is equal to the total number of agents N_a and the number of non-empty clusters is equal to the number of connected sub-graphs at time k. For each cluster $\mathcal{G}_m(k)$, by combining the local system dynamics in (1), the nonlinear dynamics of the *cluster* system is $x(k+1) = f(x(k), u(k))$, where

$x(k) = \text{col}_{i\in\mathcal{V}_m(k)}(x_i(k))$, $u(k) = \text{col}_{i\in\mathcal{V}_m(k)}(u_i(k))$ and $p(k) = \text{col}_{i\in\mathcal{V}_m(k)}(p_i(k))$, where $\text{col}(v_j)$ denote a vector which consists of the stacked subvectors v_j. Fig. 1 shows an example with three agents at two subsequent time steps. At time step k (on the left), the agents on the right are able to communicate generating the cluster $\mathcal{G}_1(k)$ with cardinality of the of nodes' set $|\mathcal{V}_1(k)| = 2$. Instead, the agent on the left cannot communicate with the others, representing a cluster $\mathcal{G}_2(k)$ with nodes' cardinality $|\mathcal{V}_2(k)| = 1$ and the cluster $\mathcal{G}_3(k)$ is empty. At the subsequent time step (on the right), agents 1 and 2 can communicate each with agent 3, thus defining a new communication topology with only one non-empty cluster $\mathcal{G}_1(k+1)$ with cardinality of the of nodes' set $|\mathcal{V}_1(k+1)| = 3$. Thus, for a cluster m, we have a plug-in operation when $|\mathcal{V}_m(k)| < |\mathcal{V}_m(k+1)|$ and a plug-out one when $|\mathcal{V}_m(k)| > |\mathcal{V}_m(k+1)|$. We finally define a position dependent set of neighbouring systems for each agent in the considered cluster.

Definition 1 (Neighboring systems): For each non-empty cluster m with $m = \{1, \dots, N_c(k)\}$, let us denote the set of all neighbors of $i \in \mathcal{V}_m(k)$, including i itself as $\mathcal{N}_i(k) = \{i\} \cup \{j : \{i,j\} \in \mathcal{E}_m(k)\}$. The states of all vehicles $j \in \mathcal{N}_i(k)$ are denoted as $x_{\mathcal{N}_i(k)} = \text{col}_{j\in\mathcal{N}_i(k)}(x_j) \in \mathbb{R}^{n_{\mathcal{N}_i(k)}}$.

C. Collision avoidance

To model the requirements of collision avoidance among agents, let us define the obstacle avoidance non-convex coupling constraint between neighboring agents as:

$$h_i(x_{\mathcal{N}_i(k)}) \leq 0, \ \forall i \in \mathcal{V}_m(k), \ \forall m \in \mathbb{N}_1^{N_c(k)}, \tag{3}$$

where $\mathbb{N}_a^b = \{n \in \mathbb{N} \mid a \leq n \leq b\}$. Possible choices for this constraint will be shown in Section IV-B. Each agent has to avoid collisions with neighbouring agents satisfying constraint (3), $\forall k \geq 0$. Thus, due to the time-varying nature of the communication topology, each agent must be able to tolerate a possible variation of the neighboring systems set $\mathcal{N}_i(k)$ guaranteeing the satisfaction of constraint (3).

D. Problem formulation

We are now in position to state the following problem.

Problem 1: Consider N_a mobile robots with dynamics (1), subject to local state and input constraints $(x_i(k), u_i(k)) \in \mathcal{X}_i \times \mathcal{U}_i$, $\forall i \in \mathcal{M}$. Each agent can communicate with neighbouring agents according to Assumption 2 defining a time-varying number of non-empty clusters ranging in the time-varying set $\{1, \dots, N_c(k)\}$. Each cluster presents a time-varying network topology as described in Definition 1. We aim to design a state feedback control law that drives every agent to their reference r_i or to the closest reachable steady state, while avoiding collision with neighbouring agents satisfying constraint (3), $\forall k \geq 0$.

III. Plug-and-Play Multi-Trajectory MPC

To solve Problem 1, the predicted agent's state trajectory has to be robust to possible network topology changes. Due to the time-varying nature of the constraints, a robust approach ensuring constraint satisfaction assuming a worst-case scenario in terms of new agents joining the network can lead to too conservative behaviour [5]. To guarantee robustness against network changes and, at the same time, exploit the best of the current information about the network, we adopt the multi-trajectory MPC (mt-MPC) concept proposed in [13], [14] and on the nonlinear tracking MPC controller proposed in [6], [7]. The main idea, particularly suitable for time-varying constraints, consists in defining an MPC problem with two trajectories, sharing the first control action, in the same finite-horizon optimal control problem (FHOCP). The first is a safe trajectory towards a polytopic convex safe set $\hat{\mathcal{S}}_i(p_i(k)) = \{q_i \in \mathbb{R}^3 : A_{c,i}(q_i - p_i(k)) \leq b_{c,i}\}$ to guarantee the system's safety, here considered in the form of robustness against network changes. The second one, also called tracking trajectory, aims at minimizing a given tracking cost. Fig. 1 shows a qualitative example where the two trajectories for each agent can be easily distinguished as well as the safe sets. We describe the approach for a single cluster and the same problem is solved by each non-empty cluster $\mathcal{G}_m(k)$ with $m \in \{1, \dots, N_c(k)\}$. We denote with the superscripts "t", "s" the variables pertaining to the tracking and safe trajectory, respectively. Furthermore, let us denote with $x_{i,(j|k)}^{(\cdot)}$ the predicted trajectory at time $k + j$ given the state at time k. Given a finite horizon $N \in \mathbb{N}$, we introduce the two tracking and safe input sequences $U_i^t = \left\{u_{i,(0|k)}^t \ u_{i,(1|k)}^t \ \dots \ u_{i,(N-1|k)}^t\right\}$, $U_i^s = \left\{u_{i,(0|k)}^s \ u_{i,(1|k)}^s \ \dots \ u_{i,(N-1|k)}^s\right\}$, where $u_{i,(0|k)}^t = u_{i,(0|k)}^s$ is the first common control action. Now, given a collection of state references $r = \text{col}_{i\in\mathcal{V}_m(k)}(r_i)$, safe sets $\hat{\mathcal{S}} = \{\hat{\mathcal{S}}_i(p_i(k)), \forall i \in \mathcal{V}_m\}$ and positive scalars $\hat{J}^s(k) = \text{col}_{i\in\mathcal{V}_m(k)}(\hat{J}_i^s(k))$, whose derivation will be clarified in Section IV, the following FHOCP $\mathscr{P}(x, r, \hat{\mathcal{S}}, \hat{J}^s)$ is solved at each time step $k \geq 0$:

$$\min_{U_i^{t,s}, \bar{x}_i^s, \bar{u}_i^s} \sum_{i=1}^{|\mathcal{V}_m(k)|} J_i(x_i, U_i^{t,s}, \bar{x}_i^s, \quad r_i) \tag{4a}$$

subject to:

$$x_{i,(0|k)}^{t,s} = x_i(k), \tag{4b}$$

$$u_{i,(0|k)}^t = u_{i,(0|k)}^s, \tag{4c}$$

$$x_{i,(j+1|k)}^{t,s} = f_i(x_{i,(j|k)}^{t,s}, u_{i,(j|k)}^{t,s}), \quad \forall j \in \mathbb{N}_0^{N-1} \tag{4d}$$

$$p_{i,(j|k)}^{t,s} = C_i x_{i,(j|k)}^{t,s}, \quad \forall j \in \mathbb{N}_0^N \tag{4e}$$

$$\left(x_{i,(j|k)}^{t,s}, u_{i,(j|k)}^{t,s}\right) \in \mathcal{X}_i \times \mathcal{U}_i, \quad \forall j \in \mathbb{N}_0^{N-1} \tag{4f}$$

$$h_i\left(x_{\mathcal{N}_i(k),(j|k)}^{t,s}\right) \leq 0, \quad \forall j \in \mathbb{N}_0^N \tag{4g}$$

$$A_c\left(p_{i,(j+1|k)}^s - p_{i,(j|k)}^s\right) \leq \frac{b_c}{N}, \quad \forall j \in \mathbb{N}_0^N \tag{4h}$$

$$x_{i,(N|k)}^s = \bar{x}_i^s = f_i(\bar{x}_i^s, \bar{u}_i^s), \tag{4i}$$

$$J_i^s(x_i, U_i^s, \bar{x}_i^s, r_i) \leq \hat{J}_i^s(k), \tag{4j}$$

$$\forall i \in \mathcal{V}_m(k).$$

The optimization variables $\bar{x}_i^s, \bar{u}_i^s$ are the artificial reference for the safe trajectory and (4j) is a convergence constraint and will be detailed in Section IV-A. Constraint (4h), instead, forces the positions of the predicted safe trajectory $p_{i,(j|k)}^s$

to lie inside a safe set $\hat{\mathcal{S}}(p_i(k))$, whose definition will be clarified in the next section. Thus, the use of two trajectories allows, on one hand, to avoid being excessively conservative thanks to the tracking trajectory that disregards the safe set, while the second trajectory always ensures the existence of a safe path that can react to possible plug-in plug-out operations.

Problem $\mathscr{P}(x, r, \hat{\mathcal{S}})$ is a nonlinear program (NLP), where the non-convex constraint (4g) is also a coupling constraint between neighbouring subsystems. The MPC problem (4) is amenable to distributed computation, and it can be solved with optimization algorithms for distributed non-convex optimization. Practical solutions to solve this problem are outside the scope of this work. Thus, the MPC control law, computed by each cluster $\mathcal{G}_m$, can be locally computed by each vehicle and applied in a receding horizon fashion.

IV. MPC DESIGN AND THEORETICAL ANALYSIS

In the following subsections we define and analyze the different elements defining problem (4), and conclude with a theoretical analysis of the MPC scheme.

A. Cost function and convergence constraint

To maximize the benefit of the multi-trajectory approach [14], and partially decouple constraint satisfaction (safety) from cost function minimization (tracking), we would ideally minimize the cost only of the tracking trajectory and neglect the safe one. However, to guarantee the convergence of the MPC scheme, also the safe trajectory has to be included in the cost [13]. The local cost functions of problem (4) is:

$$J_i(x_i, U_i^{t,s}, \bar{x}_i^s, r_i) = \sum_{j=0}^{N-1} l_i^t(x_{i,(j|k)}^t - \bar{x}_{r,i}, u_{i,(j|k)}^t - \bar{u}_{r,i}) + \beta V_{O,i}^s(\bar{x}_i^s - r_i), \quad (5)$$

where $l_i^t(\cdot,\cdot) : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$ is the stage cost function for the tracking trajectory, $V_{O,i}^s : \mathbb{R}^{d_i} \to \mathbb{R}$ is the offset safe cost function and $\beta > 0$ is a weight whose meaning will be better clarified later on. Note that for $\beta \to 0$, the cost function tends to the ideal case where only the tracking trajectory is considered in the cost [5]. As shown in [13], the mt-MPC formulation does not ensure convergence without including additional constraints to enforce a decrease in the safe cost function. To guarantee the convergence, as we will show in Section IV-D, we have to properly design the constraint (4j). To this aim let us define the following safe cost function representing a performance index for the safe trajectory:

$$J_i^s(x_i, U_i^s, \bar{x}_i^s, r_i) = \sum_{j=0}^{N-1} l_i^s(x_{i,(j|k)}^s - \bar{x}_i^s, u_{i,(j|k)}^s - \bar{u}_i^s) + V_{O,i}^s(\bar{x}_i^s - r_i), \quad (6)$$

where $l_i^s(\cdot,\cdot) : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$ is a suitable safe stage cost. We can now define, at each time step k, an upper bound on the safe trajectory cost for the current time step, by exploiting the tail of the optimal safe trajectory computed at the previous time instant:

$$\hat{J}_i^s(k) = J_i^s(x_i(k-1), U_i^{s,*}(k-1), \bar{x}_i^{s,*}, r_i) - l_i^s(x_{i,(0|k-1)}^{s,*} - \bar{x}_i^{s,*}, u_{i,(0|k-1)}^{s,*} - \bar{u}_i^{s,*}), \quad (7)$$

where the superscript "*" denotes the optimal quantities computed at time $k-1$ by solving the FHOCP (4). Thus, constraint (4j) imposes that the cost of the safe trajectory must be smaller or equal to the one obtained with the candidate safe trajectory computed at the previous time step.

Let us now define the feasibility set for the FHOCP (4) as $\mathcal{F} \doteq \{x : \mathscr{P}(x, r, \hat{\mathcal{S}}, \hat{J}^s)$ admits a solution$\}$, and let's assume that $\mathcal{F}$ is not empty and bounded. Now, for all the elements $x_i(k)$ in $x(k) \in \mathcal{F}$ let us define the set of reachable steady states starting from $x_i(k)$ as follows.

Definition 2 (Set of reachable steady states): The set $\mathcal{R}_i$ contains all the steady states that can be reached by the system starting from the initial condition $x_i(k)$ in at most N time steps with an admissible input sequence V_i.

$$\begin{aligned}
&\mathcal{R}_i\left(x_i(k), x_{\mathcal{N}_{i(k)}}, N, \hat{\mathcal{S}}_i(p_i(k), \hat{J}_i^s\right) \doteq \{\bar{x}_i \in \mathcal{X}_i : \\
&\exists V_i \in \mathbb{R}^{m_i \times N} : v_{i,(j|k)} \in \mathcal{U}_i, \forall j \in \mathbb{N}_0^{N-1}; \\
&x_{i,(0|k)} = x_i(k), \\
&x_{i,(j|k)} = f_i(x_{i,(j-1|k)}, v_{i,(j-1|k)}), \forall j \in \mathbb{N}_0^N; \\
&x_{i,(N|k)} = \bar{x}_i = f_i(\bar{x}_i, v_{i,(N-1|k)}); \\
&x_{i,(j|k)} \in \mathcal{X}_i,\ C_i x_{i,(j|k)} \in \hat{\mathcal{S}}(p_i(k)), \forall j \in \mathbb{N}_0^N; \\
&h_i\left(x_{\mathcal{N}_{i(k)},(j|k)}\right) \le 0,\ \forall j \in \mathbb{N}_0^N; \\
&J_i^s(x_i, V_i^s, \bar{x}_i, r_i) \le \hat{J}_i^s\}.
\end{aligned} \quad (8)$$

We are now in position to define the optimal reachable steady state belonging to $\mathcal{R}_i$.

Definition 3 (Optimal reachable steady state): The optimal reachable steady state $\bar{x}_i^o(k)$ is obtained at each time step by solving the following optimization problem:

$$\bar{x}_i^o(k) \in \arg\min_{\bar{x}_i \in \mathcal{R}_i} V_{O,i}^s(\bar{x}_i - r_i). \quad (9)$$

Thus, two possibilities may arise during the navigation:
i) There exists a time instant $\bar{k} \ge 0$, such that the final target r_i becomes reachable for a long enough horizon N, i.e. $\exists \bar{k} : \bar{x}_i^o(k) = r_i,\ \bar{x}_i^o(k) \in \mathcal{R}_i,\ \forall k \ge \bar{k}$.
ii) There exists a time instant $\bar{k} \ge 0$, such that the final target r_i cannot be reached, e.g. because an agent is between the vehicle and the target, but the position associated to the optimal steady state remains constant, $\forall k \ge \bar{k}$, i.e.
$\exists \bar{k} : \bar{x}_i^o(k) = \bar{x}_i^o \in \mathcal{R}_i,\ C_i \bar{x}_i^o \in \hat{\mathcal{S}}(p_i(k)),\ \forall k \ge \bar{k}$.
Finally, similarly to [6], [7], let us consider the following assumptions on the cost functions.

Assumption 3: f and $l^{(\cdot)}$ are continuous on $\bar{\mathcal{F}} \times \mathcal{U}$, where $\bar{\mathcal{F}}$ is the closure of $\mathcal{F}$, hence $\exists \alpha_f, \alpha_l \in \mathcal{K}_\infty : \|f_i(\bar{x}, \bar{u}) - f_i(\hat{x}, \hat{u})\| \le \alpha_f(\|(\bar{x}, \bar{u}) - (\hat{x}, \hat{u})\|)$, $|l_i^{(\cdot)}(\bar{x}, \bar{u}) - l_i^{(\cdot)}(\hat{x}, \hat{u})| \le \alpha_l(\|(\bar{x}, \bar{u}) - (\hat{x}, \hat{u})\|)$, $\forall (\bar{x}, \bar{u}), (\hat{x}, \hat{u}) \in \mathcal{F} \times \mathcal{U}$, $\forall i$, for some vector norm $\|\cdot\|$. The stage cost $l_i^{(\cdot)}(x, u)$ is designed such that $l_i^{(\cdot)}(x, u) \ge \alpha_l(|x|)$, where α_l is a $\mathcal{K}_\infty$ function and the safe offset cost $V_{O,i}^s(\bar{x}_i - r_i)$ is positive definite, strictly convex and subdifferentiable functions with a unique minimizer that is $\bar{x}_i^o = \arg\min_{\bar{x}_i} V_{O,i}^s(\bar{x}_i - r_i)$.
Where a continuous, strictly increasing function $\alpha : [0, +\infty) \to [0, +\infty)$ is said to belong to class $\mathcal{K}_\infty$ if $\alpha(0) = 0$ and if $\lim_{a \to +\infty} \alpha(a) = +\infty$.

B. Collision avoidance

Let's approximate the shape of the vehicles with a sphere $\mathcal{T}_S(p_i) = \{p_i \in \mathbb{R}^3 : \|p_i\|_2 \leq \sigma\}$. Hence, to enforce collision avoidance between agents, it can be imposed that the euclidean distance between the vehicle i and j is greater or equal than twice the radius σ accounting for the maximum size of the vehicle $\forall j \in \mathcal{N}_i(k)\backslash\{i\}$:

$$h_i(x_{\mathcal{N}_i(k)}) = -\|p_i - p_j\|_2^2 + (2\sigma)^2 + d_{min} \leq 0. \tag{10}$$

An alternative collision avoidance constraints with reduced conservatism in the approximation of the vehicle's shape can be found in [20].

C. Safe set

To avoid collisions among different agents, it is crucial to be able to reach a steady state within the communication set $\mathcal{C}_i(p_i(k))$. It is important that the safe set is located within the communication set in order to ensure the existence of a safe trajectory in an area where we are aware of the presence of other agents, i.e., where we can communicate. To define the safe set $\hat{\mathcal{S}}_i(p_i)$ considered in constraint (4h), where the safe trajectory can remain, let us firstly consider the size of the vehicle, by means of the set $\mathcal{O} = \{R(x_i)\mathcal{T}(p_i),\ \forall x_i \in (\mathcal{X}_i \times \mathcal{U}_i)^{\perp}\}$, where $(\mathcal{X}_i \times \mathcal{U}_i)^{\perp}$ is the projection of $\mathcal{X}_i \times \mathcal{U}_i$ on the state space. Thus, we use the computed set to tighten the communication set $\mathcal{C}_i(p_i)$:

$$\mathcal{S}_i(p_i) = \mathcal{C}_i(p_i) \ominus \mathcal{O}, \tag{11}$$

where $\ominus$ is the Pontryagin difference. The obtained set $\mathcal{S}_i(p_i)$ represents a safe region where the vehicle can safely counteract to a possible change in the topology and the set $\mathcal{O}$ accounts for the size of the vehicle. Finally, we under-approximate the set $\mathcal{S}_i(p_i)$ with the following polytopic set:

$$\hat{\mathcal{S}}_i(p_i) \doteq \{p_i, q \in \mathbb{R}^3 : A_c(q - p_i) \leq b_c\}, \tag{12}$$

that, in practice, can be easily computed by performing the convex hull of some samples of the borders of $\mathcal{S}_i(p_i)$. Constraint (12), is always centered at the vehicle's position $p_i(k)$. This constraint's feature, however, can cause a loss of feasibility. To guarantee that the problem is recursively feasible, we need to ensure that the tail of the predicted safe trajectory lies in the safe sets generated by shifting the set on the predicted trajectory $p^s_{i(j|k)} \in \hat{\mathcal{S}}_i(p^s_{i(l|k)}),\ \forall l \in \mathbb{N}_0^j$. To address this issue, we propose the implementation (4h) to satisfy constraint (12). Specifically, constraint (4h) imposes that the predicted trajectory is able to reach the border of the polytopic set (12) only in N time steps, see [20, Fig. 2].

D. Theoretical analysis

We now analyze the theoretical guarantees of the proposed MPC scheme, by exploiting the different ingredients described in the previous section. Before stating the main result, let us consider the following assumption.

Assumption 4: For any $x \in \mathcal{F}$, there exists a finite value of the optimal safe cost $J^{s,o}$ and the FHOCP $\mathscr{P}(x, r, \hat{\mathcal{S}}, \hat{J}^s)$ has at least one global minimum, which is computed by the solver independently from its initialization.

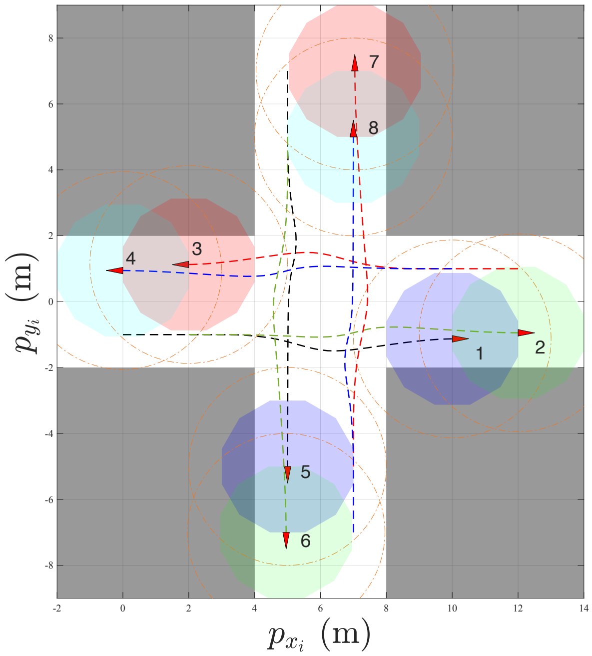

Fig. 2: Simulation results of the intersection problem considered. Dashed colored lines represent vehicle trajectories. Red triangles are the final pose of the vehicles. Orange dashed circles represent the communication set and colored polytopes the safe sets $\hat{\mathcal{S}}_i(p_i)$.

The latter assumption is quite usual and implicitly considered in the context of economic MPC and nonlinear MPC [7] and despite is a rather strong one, it is only needed to show convergence to a global optimum. Moreover, it is satisfied if the FHOCP is convex, which is the important case of linear systems with convex constraints and convex stage cost.

We are now in position to state the following proposition:

Proposition 1: Let Assumptions [1-4] be satisfied and assume that the FHOCP (4) at time $k = 0$ is feasible. Then, problem (4) is recursively feasible and the system controlled by the MPC controller derived from its solution converges arbitrarily close to the optimal reachable steady state according to Definition 3 while satisfying constraints $\forall k \geq 0$.

The proof of Proposition 1 can be found in [20] and it is based on the nonlinear MPC schemes presented in [6], [7].

V. Numerical Example

In this section, we show the effectiveness of the approach via numerical simulations. We employed a Quad-Core Intel Core i7 (2.8 GHz, 16 GB) on MATLAB 2020b, using CasADI [21] to build problem (4) and IPOPT [22] to solve local non-convex problems. We consider eight ground vehicles approaching an intersection and described by the following discrete time kinematic model:

$$x_i(k+1) = \begin{pmatrix} p_{x_i}(k) + T_s \cos(\theta_i) v_i(k) \\ p_{y_i}(k) + T_s \sin(\theta_i) v_i(k) \\ v_i(k) + T_s a_i(k) \\ \theta_i(k) + T_s \frac{v_i(k)}{L} \tan(\gamma_i(k)) \\ \gamma_i(k) + T_s \delta_i(k) \end{pmatrix},$$

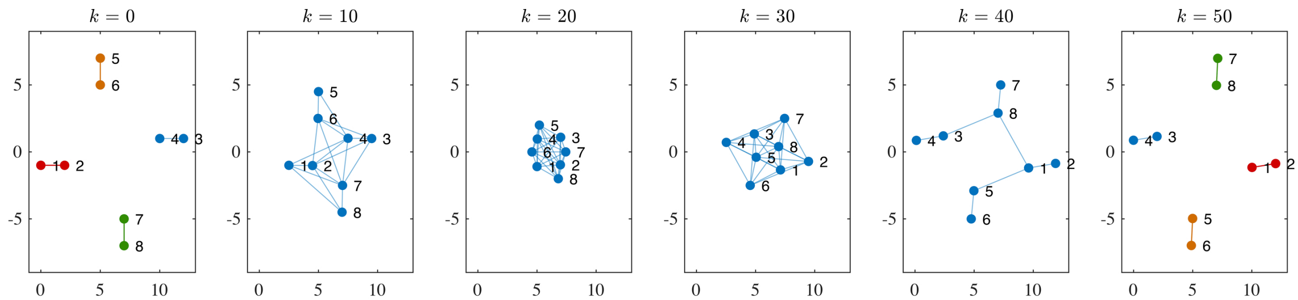

Fig. 3: Evolution of the communication graph at iteration $k = 0, 10, 20, 30, 40, 50$

where T_s is the sampling time, $\begin{bmatrix} p_{x_i} \\ p_{y_i} \end{bmatrix} \in \mathbb{R}^2$ is the vehicle's position and θ_i, $\gamma_i \in \mathbb{R}$ are the yaw and the steering angles. The inputs are the commanded acceleration $a_i \in \mathbb{R}$ and the steering rate $\delta_i \in \mathbb{R}$. Thus, each local system has a state $x_i \in \mathbb{R}^5$ and an input $u_i \in \mathbb{R}^2$. The sampling time T_s is $0.1\ s$, and the wheelbase of the vehicle L is $0.8\ m$. We assume a circular communication area $\mathcal{C}_i(p_i) = \{p_i \in \mathbb{R}^2 : \|p_i\|_2 \leq 3m\}$ and we assume a circular shape for the vehicles with a diameter of $1\ m$, leading to the non-convex coupling constraint (10) and a set $\mathcal{O} = \{p_i \in \mathbb{R}^2 :\ \|p_i\|_2 \leq \sigma\}$ where $\sigma = 1\ m$. The acceleration and the steering rate are limited to $|a_i| \leq 3\ m/s^2$ and $|\delta_i| \leq 1$ rad/s. The velocity and the steering angle are limited to $v_i \leq 2\ m/s$ and $\gamma_i \in [-\frac{\pi}{2},\ \frac{\pi}{2}]$ rad and the yaw $\theta_i \in [0,\ 2\pi]$ rad. Each vehicle has to cross the workspace by guaranteeing the obstacle avoidance constraint and solving a problem only with the neighbouring subsystems. Fig. 2 shows the trajectories obtained in closed loop together with the communication areas $\mathcal{C}_i(p_i)$ and the safe sets $\hat{\mathcal{S}}_i(p_i)$. Finally Fig. 3 shows the evolution of the time-varying communication network with plug and play operations at different iterations.

VI. Conclusion

In this work, we propose a multi-trajectory MPC for the trajectory generation of multi-agent systems able to handle changes in the communication network topology in real-time without request. We proved that the approach, based on nonlinear tracking MPC, makes the agents safely converge to their reference or to the closest admissible steady state. Finally, a numerical example demonstrates the effectiveness of the approach in a traffic intersection. The current research activities aim to test the approach's effectiveness on a custom hardware platform with experimental evaluation.

References

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[2] J. Shamma, *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[3] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[4] J. Stoustrup, "Plug & play control: Control technology towards new challenges," *European Journal of Control*, vol. 15, no. 3-4, pp. 311–330, 2009.

[5] D. Saccani and L. Fagiano, "Autonomous UAV navigation in an unknown environment via multi-trajectory model predictive control," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1577–1582.

[6] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo, "Nonlinear MPC for tracking piece-wise constant reference signals," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3735–3750, 2018.

[7] L. Fagiano and A. R. Teel, "Generalized terminal state constraint for model predictive control," *Automatica*, vol. 49, no. 9, pp. 2622–2631, 2013.

[8] K. P. Wabersich and M. N. Zeilinger, "Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning," *arXiv preprint arXiv:1812.05506*, 2018.

[9] S. Muntwiler, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "Distributed model predictive safety certification for learning-based control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5258–5265, 2020.

[10] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[11] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Transactions on Automatic Control*, 2022.

[12] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "FASTER: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Transactions on Robotics*, 2021.

[13] R. Soloperto, A. Mesbah, and F. Allgöwer, "Safe exploration and escape local minima with model predictive control under partially unknown constraints," *arXiv preprint arXiv:2205.03614*, 2022.

[14] D. Saccani, L. Cecchin, and L. Fagiano, "Multitrajectory Model Predictive Control for Safe UAV Navigation in an Unknown Environment," *IEEE Transactions on Control Systems Technology*, 2022.

[15] J. P. Alsterda and J. C. Gerdes, "Contingency model predictive control for linear time-varying systems," *arXiv preprint arXiv:2102.12045*, 2021.

[16] M. N. Zeilinger, Y. Pu, S. Riverso, G. Ferrari-Trecate, and C. N. Jones, "Plug and play distributed model predictive control based on distributed invariance and optimization," in *52nd IEEE conference on decision and control*. IEEE, 2013, pp. 5770–5776.

[17] A. Carron, K. P. Wabersich, and M. N. Zeilinger, "Plug-and-play distributed safety verification for linear control systems with bounded uncertainties," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 3, pp. 1501–1512, 2021.

[18] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

[19] A. Carron and M. N. Zeilinger, "Model predictive coverage control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6107–6112, 2020.

[20] D. Saccani, L. Fagiano, M. N. Zeilinger, and A. Carron, "Model predictive control for multi-agent systems under limited communication and time-varying network topology," *arXiv preprint arXiv:2304.01649*, 2023.

[21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[22] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.