Decentralized Abstractions and Timed Constrained Planning of a General Class of Coupled Multi-Agent Systems

Alexandros Nikou, Shahab Heshmati-alamdari, Christos Verginis and Dimos V. Dimarogonas

Abstract— **This paper presents a fully automated procedure for controller synthesis for a general class of multi-agent systems under coupling constraints. Each agent is modeled with dynamics consisting of two terms: the first one models the coupling constraints and the other one is an additional bounded control input. We aim to design these inputs so that each agent meets an individual high-level specification given as a Metric Interval Temporal Logic (MITL). Furthermore, the connectivity of the initially connected agents, is required to be maintained. First, assuming a polyhedral partition of the workspace, a novel decentralized abstraction that provides controllers for each agent that guarantee the transition between different regions is designed. The controllers are the solution of a Robust Optimal Control Problem (ROCP) for each agent. Second, by utilizing techniques from formal verification, an algorithm that computes the individual runs which provably satisfy the high-level tasks is provided. Finally, simulation results conducted in MATLAB environment verify the performance of the proposed framework.**

Index Terms— **multi-agent systems, cooperative control, hybrid systems.**

I. Introduction

Cooperative control of multi-agent systems has traditionally focused on designing distributed control laws in order to achieve global tasks such as consensus and formation control, and at the same time fulfill properties such as network connectivity and collision avoidance. Over the last few years, the field of control of multi-agent systems under temporal logic specifications has been gaining attention. In this work, we aim to additionally introduce specific time bounds into these tasks, in order to include specifications such as: "Robot 1 and robot 2 should visit region A and B within 4 time units respectively or "Both robots 1 and 2 should periodically survey regions A_1, A_2, A_3, avoid region X and always keep the longest time between two consecutive visits to A_1 below 8 time units".

The qualitative specification language that has primarily been used to express the high-level tasks is Linear Temporal Logic (LTL) (see, e.g., [1], [2]). There is a rich body of literature containing algorithms for verification and synthesis of multi-agent systems under temporal logic specifications ([3]–[5]). A three-step hierarchical procedure to address the problem of multi-agent systems under LTL specifications is described as follows ([6]–[8]): first the dynamics of each agent is abstracted into a Transition System (TS). Second, by invoking ideas from formal verification, a discrete plan that meets the high-level tasks is synthesized for each agent. Third, the discrete plan is translated into a sequence of continuous time controllers for the original continuous dynamical system of each agent.

Controller synthesis under timed specifications has been considered in [9]–[13]. However, all these works are restricted to single agent planning and are not extendable to multi-agent systems in a straightforward way. The multi-agent case has been considered in [14], where the vehicle routing problem was addressed, under Metric Temporal Logic (MTL) specifications. The corresponding approach does not rely on automata-based verification, as it is based on a construction of linear inequalities and the solution of a Mixed-Integer Linear Programming (MILP) problem.

An automata-based solution was proposed in our previous work [15], where MITL formulas were introduced in order to synthesize controllers such that every agent fulfills an individual specification and the team of agents fulfills a global specification. Specifically, the abstraction of each agent's dynamics was considered to be given and an upper bound of the time that each agent needs to perform a transition from one region to another was assumed. Furthermore, potential coupled constraints between the agents were not taken into consideration. Motivated by this, in this work, we aim to address the aforementioned issues. We assume that the dynamics of each agent consists of two parts: the first part is a nonlinear function representing the coupling between the agent and its neighbors, and the second one is an additional control input which will be exploited for high-level planning. Hereafter, we call it a free input. A decentralized abstraction procedure is provided, which leads to an individual Weighted Transition System (WTS) for each agent and provides a basis for high-level planning.

Abstractions for both single and multi-agent systems have been provided e.g. in [16]–[24]. In this paper, we deal with the complete framework of both abstractions and controller synthesis of multi-agent systems. We start from the dynamics of each agent and we provide controllers that guarantee the transition between the regions of the workspace, while the initially connected agents remain connected for all times. The

Alexandros Nikou, Christos Verginis and Dimos V. Dimarogonas are with the ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden and with the KTH Center for Autonomous Systems. Email: {anikou, cverginis, dimos}@kth.se. Shahab Heshmati-alamdari is with the Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Athens, Greece. Email: {shahab}@mail.ntua.gr. This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the Swedish Research Council (VR), the Swedish Foundation for Strategic Research, the Knut och Alice Wallenberg Foundation, the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 644128 (AEROWORKS) and the EU H2020 Research and Innovation Programme under GA No. 731869 (Co4Robots).

decentralized controllers are the solution of an ROCP. Then, each agent is assigned an individual task given as an MITL formulas. We aim to synthesize controllers, in discrete level, so that each agent performs the desired individual task within specific time bounds as imposed by the MITL formulas. In particular, we provide an automatic controller synthesis method of a general class of coupled multi-agent systems under high-level tasks with timed constraints. Compared to existing works on multi-agent planning under temporal logic specifications, the proposed approach considers dynamically coupled multi-agent systems under timed temporal specifications in a distributed way.

In our previous work [25], we treated a similar problem, but the under consideration dynamics were linear couplings and connectivity maintenance was not guaranteed by the proposed control scheme. Furthermore, the procedure was partially decentralized, due to the fact that a product Wighted Transition System (WTS) was required, which rendered the framework computationally intractable. To the best of the authors' knowledge, this is the first time that a fully automated framework for a general class of multi-agent systems consisting of both constructing purely decentralized abstractions and conducting timed temporal logic planning is considered.

This paper is organized as follows. In Section II a description of the necessary mathematical tools, the notations and the definitions are given. Section III provides the dynamics of the system and the formal problem statement. Section IV discusses the technical details of the solution. Section V is devoted to a simulation example. Finally, conclusions and future work are discussed in Section VI.

II. Notation and Preliminaries

A. Notation

We denote by $\mathbb{R}, \mathbb{Q}_+, \mathbb{N}$ the set of real, nonnegative rational and natural numbers including 0, respectively. $\mathbb{R}^n_{\geq 0}$ and $\mathbb{R}^n_{>0}$ are the sets of real n-vectors with all elements nonnegative and positive, respectively. Define also $\mathbb{T}_\infty = \mathbb{T} \cup \{\infty\}$ for a set $\mathbb{T} \subseteq \mathbb{R}$. Given a set S, denote by $|S|$ its cardinality, by $S^N = S \times \cdots \times S$ its N-fold Cartesian product, and by 2^S the set of all its subsets. Given the sets S_1, S_2, their *Minkowski addition* is defined by $S_1 \oplus S_2 = \{s_1 + s_2 : s_1 \in S_1, s_2 \in S_2\}$. $I_n \in \mathbb{R}^{n \times n}$ stands for the identity matrix. The notation $\|x\|$ is used for the Euclidean norm of a vector $x \in \mathbb{R}^n$. $\|A\| = \max\{\|Ax\| : \|x\| = 1\}$ stands for the induced norm of a matrix $A \in \mathbb{R}^{n \times n}$. $\mathcal{B}(c, \underline{r}) = \{x \in \mathbb{R}^2 : \|x - c\| \leq \underline{r}\}$ is the disk of center $c \in \mathbb{R}^2$ and $\underline{r} \in \mathbb{R}_{>0}$. The absolute value of the maximum singular value and the absolute value of the minimum eigenvalue of a matrix $A \in \mathbb{R}^{n \times n}$ are denoted by $\sigma_{\max}(A), \lambda_{\min}(A)$, respectively. The indexes i and j stand for agent i and its neighbors (see Sec. III for the definition of neighbors), respectively; $\mu, z \in \mathbb{N}$ are indexes used for sequences and sampling times, respectively.

Definition 1. Consider two sets $A, B \subseteq \mathbb{R}^n$. Then, the *Pontryagin difference* is defined by:

$$A \sim B = \{x \in \mathbb{R}^n : x + y \in A, \forall\ y \in B\}.$$

Definition 2. ([26]) A continuous function $\alpha : [0, a) \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{K}$, if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\alpha(r) \to \infty$, as $r \to \infty$.

Definition 3. ([26]) A continuous function $\beta : [0, a) \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{KL}$, if:

- For each fixed s, $\beta(r, s) \in \mathcal{K}$ with respect to r.
- For each fixed r, $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$, at $s \to \infty$.

Definition 4. ([27]) A nonlinear system $\dot{x} = f(x, u)$ with initial condition $x(t_0)$ is said to be *Input to State Stable (ISS)* if there exist functions $\beta \in \mathcal{KL}$ and $\sigma \in \mathcal{K}_\infty$ such that:

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t) + \sigma(\|u\|).$$

Definition 5. ([27]) A Lyapunov function $V(x, u)$ for the nonlinear system $\dot{x} = f(x, u)$ with initial condition $x(t_0)$ is said to be *ISS-Lyapunov function* if there exist functions $\alpha, \sigma \in \mathcal{K}_\infty$ such that:

$$\dot{V}(x, u) \leq -\alpha(\|x\|) + \sigma(\|u\|), \forall x, u. \tag{1}$$

Theorem 1. *A nonlinear system $\dot{x} = f(x, u)$ with initial condition $x(t_0)$ is said to be ISS if and only if it admits a ISS-Lyapunov function.*

Proof. The proof can be found in [28]. □

B. Partitions

In the subsequent analysis a discrete partition of the workspace will be considered which is formalized through the following definition.

Definition 6. Given a set S, we say that a family of sets $\{S_\ell\}_{\ell \in \mathbb{I}}$ forms a partition of S if $S \neq \emptyset$, $\bigcup\limits_{\ell \in \mathbb{I}} S_\ell = S$ and for every $S, S' \in S$ with $S \neq S'$ it holds $S \cap S' = \emptyset$.

Hereafter, every region S_ℓ of a partition S will be called *region*.

C. Time Sequence, Timed Run and Weighted Transition System

In this section we include some definitions that are required to analyze our framework.

An infinite sequence of elements of a set X is called an *infinite word* over this set and it is denoted by $\chi = \chi(0)\chi(1)\ldots$ The z-th element of a sequence is denoted by $\chi(z)$. For certain technical reasons that will be clarified in the sequel, we will assume hereafter that $\mathbb{T} = \mathbb{Q}_+$.

Definition 7. ([29]) A *time sequence* $\tau = \tau(0)\tau(1)\ldots$ is an infinite sequence of time values $\tau(\mu) \in \mathbb{T} = \mathbb{Q}_+$, satisfying the following properties: 1) Monotonicity: $\tau(\mu) < \tau(\mu + 1)$ for all $j \geq 0$; 2) Progress: For every $t \in \mathbb{T}$, there exists $\mu \geq 1$, such that $\tau(\mu) > t$.

An *atomic proposition* σ is a statement that is either True ($\top$) or False ($\bot$).

Definition 8. ([29]) Let Σ be a finite set of atomic propositions. A *timed word* w over the set Σ is an infinite sequence $w^t = (w(0), \tau(0))(w(1), \tau(1))\dots$ where $w(0)w(1)\dots$ is an infinite word over the set 2^Σ and $\tau(0)\tau(1)\dots$ is a time sequence with $\tau(\mu) \in \mathbb{T}, \mu \geq 0$.

Definition 9. A Weighted Transition System (*WTS*) is a tuple $(S, S_0, Act, \longrightarrow, d, \Sigma, L)$ where S is a finite set of states; $S_0 \subseteq S$ is a set of initial states; Act is a set of actions; $\longrightarrow \subseteq S \times Act \times S$ is a transition relation; $d : \longrightarrow \to \mathbb{T}$ is a map that assigns a positive weight to each transition; Σ is a finite set of atomic propositions; and $L : S \to 2^\Sigma$ is a labeling function.

Definition 10. A *timed run* of a WTS is an infinite sequence $r^t = (r(0), \tau(0))(r(1), \tau(1))\dots$, such that $r(0) \in S_0$, and for all $\mu \geq 1$, it holds that $r(\mu) \in S$ and $(r(\mu), \alpha(\mu), r(\mu+1)) \in \longrightarrow$ for a sequence of actions $\alpha(1)\alpha(2)\dots$ with $\alpha(\mu) \in Act, \forall\ \mu \geq 1$. The *time stamps* $\tau(\mu), \mu \geq 0$ are inductively defined as: 1) $\tau(0) = 0$; 2) $\tau(\mu+1) = \tau(\mu) + d(r(\mu), \alpha(\mu), r(\mu+1)), \forall\ \mu \geq 1$. Every timed run r^t generates a *timed word* $w(r^t) = (w(0), \tau(0))\ (w(1), \tau(1))\dots$ over the set 2^Σ where $w(\mu) = L(r(\mu))$, $\forall\ \mu \geq 0$ is the subset of atomic propositions that are true at state $r(\mu)$.

D. Metric Interval Temporal Logic (MITL)

The syntax of *Metric Interval Temporal Logic (MITL)* over a set of atomic propositions Σ is defined by the grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1\ \mathcal{U}_I\ \varphi_2,$$

where $\sigma \in \Sigma$, and $\bigcirc$, $\Diamond$, $\Box$ and $\mathcal{U}$ are the next, eventually, always and until temporal operators, respectively; $I = [a, b] \subseteq \mathbb{T}$ where $a, b \in [0, \infty]$ with $a < b$ is a non-empty timed interval. MITL can be interpreted either in continuous or point-wise semantics [30]. In this paper, the latter approach is utilized, since the consideration of point-wise (event-based) semantics is more suitable for the automata-based specifications considered in a discretized state-space. The MITL formulas are interpreted over timed words like the ones produced by a WTS it is given in Def. 10.

Definition 11. ([30], [31]) Given a timed word $w^t = (w(0), \tau(0))(w(1), \tau(1))\dots$, an MITL formula φ and a position i in the timed word, the satisfaction relation $(w^t, i) \models \varphi$, for $i \geq 0$ (read w^t satisfies φ at position μ) is inductively defined as follows:

$$\begin{aligned}
&(w^t, \mu) \models p \Leftrightarrow p \in w(\mu),\\
&(w^t, \mu) \models \neg\varphi \Leftrightarrow (w^t, i) \not\models \varphi,\\
&(w^t, \mu) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (w^t, \mu) \models \varphi_1 \text{ and } (w^t, \mu) \models \varphi_2,\\
&(w^t, \mu) \models \bigcirc_I \varphi \Leftrightarrow (w^t, \mu+1) \models \varphi\\
&\qquad\qquad\qquad \text{and } \tau(\mu+1) - \tau(i) \in I,\\
&(w^t, \mu) \models \Diamond_I \varphi \Leftrightarrow \exists \mu' \geq \mu, \text{ such that}\\
&\qquad\qquad\qquad (w^t, j) \models \varphi, \tau(\mu') - \tau(\mu) \in I,\\
&(w^t, \mu) \models \Box_I \varphi \Leftrightarrow \forall \mu' \geq \mu,\\
&\qquad\qquad\qquad \tau(\mu') - \tau(\mu) \in I \Rightarrow (w^t, \mu') \models \varphi,\\
&(w^t, \mu) \models \varphi_1\ \mathcal{U}_I\ \varphi_2 \Leftrightarrow \exists \mu' \geq \mu, \text{ s.t. } (w^t, \mu') \models \varphi_2,\\
&\qquad \tau(\mu') - \tau(\mu) \in I \text{ and } (w^t, \mu'') \models \varphi_1, \forall\ \mu \leq \mu'' < \mu'.
\end{aligned}$$

We say that a timed run $r^t = (r(0), \tau(0))(r(1), \tau(1))\dots$ satisfies the MITL formula φ (we write $r^t \models \varphi$) if and only if the corresponding timed word $w(r^t) = (w(0), \tau(0))(w(1), \tau(1))\dots$ with $w(\mu) = L(r(\mu)), \forall \mu \geq 0$, satisfies the MITL formula ($w(r^t) \models \varphi$).

It has been proved that MITL is decidable in infinite words and point-wise semantics, which is the case considered here (see [32], [33] for details). The model checking and satisfiability problems are *EXPSPACE*-complete. It should be noted that in the context of timed systems, EXSPACE complexity is fairly low [34]. An example with a WTS and two runs r_1^t, r_2^t that satisfy two MITL formulas can be found in [35, Section II, page 4].

E. Timed Büchi Automata

Timed Büchi Automata (TBA) were introduced in [29] and in this work, we also partially adopt the notation from [34], [36]. Let $C = \{c_1, \dots, c_{|C|}\}$ be a finite set of *clocks*. The set of *clock constraints* $\Phi(C)$ is defined by the grammar

$$\phi := \top \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid c \bowtie \psi,$$

where $c \in C$ is a clock, $\psi \in \mathbb{T}$ is a clock constant and $\bowtie \in \{<, >, \geq, \leq, =\}$. A clock *valuation* is a function $\nu : C \to \mathbb{T}$ that assigns a value to each clock. A clock c_i has valuation ν_i for $i \in \{1, \dots, |C|\}$, and $\nu = (\nu_1, \dots, \nu_{|C|})$. We denote by $\nu \models \phi$ the fact that the valuation ν satisfies the clock constraint ϕ.

Definition 12. A *Timed Büchi Automaton* is a tuple $\mathcal{A} = (Q, Q^{\text{init}}, C, Inv, E, F, \Sigma, \mathcal{L})$ where Q is a finite set of locations; $Q^{\text{init}} \subseteq Q$ is the set of initial locations; C is a finite set of clocks; $Inv : Q \to \Phi(C)$ is the invariant; $E \subseteq Q \times \Phi(C) \times 2^C \times Q$ gives the set of edges; $F \subseteq Q$ is a set of accepting locations; Σ is a finite set of atomic propositions; and $\mathcal{L} : Q \to 2^\Sigma$ labels every state with a subset of atomic propositions.

For the semantics of TBA we refer the reader to [35, Section II, page 4]. The problem of deciding the emptiness of the language of a given TBA $\mathcal{A}$ is PSPACE-complete [29]. Any MITL formula φ over Σ can be algorithmically translated to a TBA with the alphabet 2^Σ, such that the language of timed words that satisfy φ is the language of timed words produced by the TBA ([32], [37], [38]). An example of a TBA and accepting runs of it can be found in [35, Section II, page 4].

Remark 1. Traditionally, the clock constraints and the TBAs are defined with $\mathbb{T} = \mathbb{N}$. However, they can be extended to accommodate $\mathbb{T} = \mathbb{Q}_+$, by multiplying all the rational numbers that are appearing in the state invariants and the edge constraints with their least common multiple.

III. Problem Formulation

A. System Model

Consider a system of N agents, with $\mathcal{V} = \{1, \ldots, N\}, N \geq 2$, operating in a workspace $W \subseteq \mathbb{R}^2$. The workspace is assumed to be closed, bounded and connected. Let $x_i : \mathbb{R}_{\geq 0} \to D$ denotes the position of each agent in the workspace at time $t \in \mathbb{R}_{\geq 0}$. Each agent is equipped with a sensor device that can sense omnidirectionally. Let the disk $\mathcal{B}(x_i(t), \underline{r})$ model the sensing zone of agent i at time $t \in \mathbb{R}_{\geq 0}$, where $\underline{r} \in \mathbb{R}_{\geq 0}$ is the sensing radius. The sensing radius is the same for all the agents. Let also $h > 0$ denote the constant sampling period of the system. We make the following assumption:

Assumption 1. (Measurements Assumption) It is assumed that each agent i, is able to measure its own position and all agents' positions that are located within agent's i sensing zone without any delays.

According to Assumption 1, the agent's i neighboring set at time t_0 is defined by $\mathcal{N}_i = \{j \in \mathcal{V} : x_j(t_0) \in \mathcal{B}(x_i(t_0), \underline{r})\}$. For the neighboring set $\mathcal{N}_i$ define also $N_i = |\mathcal{N}_i|$. Note that $i \in \mathcal{N}_j \Leftrightarrow j \in \mathcal{N}_i, \forall\ i, j \in \mathcal{V}, i \neq j$. The control design for every agent i should guarantee that it remains connected with all its neighbors $j \in \mathcal{N}_i$, for all times.

Consider the neighboring set $\mathcal{N}_i$. The coupled dynamics of each agent are given in the form:

$$\dot{x}_i = f(x_i, \bar{x}_i) + u_i,\ x_i \in W,\ i \in \mathcal{V}, \tag{2}$$

where $f : W \times W^{N_i} \to W$, is a nonlinear function representing the coupling between agent i and its neighbors $i_1, \ldots, i_{N_i}$. The notation $\bar{x}_i = [x_{i_1}^\top, \ldots, x_{i_{N_i}}^\top]^\top \in W^{N_i}$ is used for the vector of the neighbors of agent i, and $u_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^2,\ i \in \mathcal{V}$ is the control input of each agent. For the dynamics of each agent the following assumption are taken.

Assumption 2. There exist constants $u_{\max}, \bar{M}$ with $0 < u_{\max} < \bar{M} < \infty$ such that the following holds $\forall i \in \mathcal{V}, (x_i, \bar{x}_i) \in W \times W^{N_i}$:

$$\|f_i(x_i, \bar{x}_i)\| \leq \bar{M}, \tag{3a}$$

$$u_i \in \mathcal{U}_i \triangleq \{u_i \in \mathbb{R}^2 : \|u_i\| \leq u_{\max}\}. \tag{3b}$$

Assumption 3. The functions $f_i(x_i, \bar{x}_i), i \in \mathcal{V}$ are *Lipschitz continuous* in $W \times W^{N_i}$. Thus, there exists constants $L_i, \bar{L}_i > 0$ such that the following inequalities hold:

$$\|f_i(x_i, \bar{x}_i) - f_i(y_i, \bar{x}_i)\| \leq L_i \|x_i - y_i\|, \tag{4a}$$

$$\|f_i(x_i, \bar{x}_i) - f_i(x_i, \bar{y}_i)\| \leq \bar{L}_i \|\bar{x}_i - \bar{y}_i\|, \tag{4b}$$

for all $x_i, y_i \in W, \bar{x}_i, \bar{y}_i \in W^{N_i}, i \in \mathcal{V}$.

Remark 2. The coupling terms $f_i(x_i, \bar{x}_i), i \in \mathcal{V}$ are encountered in a large set of multi-agent protocols [39], including consensus, connectivity maintenance, collision avoidance and formation control. In addition, (2) may represent internal dynamics of the system as for instance in the case of smart buildings (see e.g., [40]) where the temperature $T_i, i \in \mathcal{V}$ of each room evolves according to the law $\dot{T}_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij}(T_j - T_i) + u_i$, with α_{ij} representing the heat conductivity between rooms i and j and u_i the heating/cooling capabilities of the room.

B. Specification

Our goal is to control the multi-agent system (2) so that each agent obeys a given individual specification. In particular, it is required to drive each agent to a sequence of desired subsets of the *workspace* W within certain time limits and provide certain atomic tasks there. Atomic tasks are captured through a finite set of atomic propositions $\Sigma_i, i \in \mathcal{V}$, with $\Sigma_i \cap \Sigma_j = \emptyset$, for all $i, j \in \mathcal{V}, i \neq j$, which means that the agents do not share any atomic propositions. Each position x_i of each agent $i \in \mathcal{V}$ is labeled with atomic propositions that hold there. Initially, a labeling function

$$\Lambda_i : W \to 2^{\Sigma_i}, \tag{5}$$

is introduced for each agent $i \in \mathcal{V}$ which maps each state $x_i \in \mathbb{R}^2$ with the atomic propositions $\Lambda_i(x_i)$ which hold true at x_i i.e., the subset of atomic propositions that hold for agent i in position x_i. Define also by $\Lambda(x) = \bigcup\limits_{i \in \mathcal{V}} \Lambda_i(x)$ the union of all the labeling functions. Let us now introduce the following assumption which is important for defining the problem properly.

Assumption 4. There exists a partition $D = \{D_\ell\}_{\ell \in \mathbb{I}}$ of the workspace W which respects the labeling function Λ i.e., for all $D_\ell \in D$ it holds that $\Lambda(x) = \Lambda(x'), \forall\ x, x' \in D_\ell$. This assumption, intuitively, and without loss of generality, means that the same atomic propositions hold at all the points that belong to the same region of the partition.

Although the regions $D_\ell, \ell \in \mathbb{I}$ of the partition D may have different geometric shape, without loss of generality, we assume that they are hexagons with side length R. Define also for each agent i a labeling function:

$$L_i : D \to 2^{\Sigma_i}, \tag{6}$$

which maps every region of the partition D to the subset of the atomic propositions which hold true there. Furthermore, we assume that a time step $T > h > 0$ is given. This time step models the required time in which each agent should transit from a region to a neighboring region and is the same for all the agents.

The trajectory of each agent i is denoted by $x_i(t), t \geq 0, i \in \mathcal{V}$. The trajectory $x_i(t)$ is associated with a unique sequence:

$$r_{x_i}^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \ldots,$$

of regions that the agent i crosses, where for all $\mu \geq 0$ it holds that: $x_i(\tau_i(\mu) \in r_i(\mu)$ and $\Lambda_i(x_i(t)) = L_i(r_i(\mu)), \forall\ t \in [\tau_i(\mu), \tau_i(\mu+1))$ for some $r_i(\mu) \in D$ and $r_i(\mu) \neq r_i(\mu+1)$. The timed word:

$$w_{x_i}^t = (L_i(r_i(0)), \tau_i(0))(L_i(r_i(1)), \tau_i(1))$$
$$(L_i(r_i(2)), \tau_i(2)) \ldots,$$

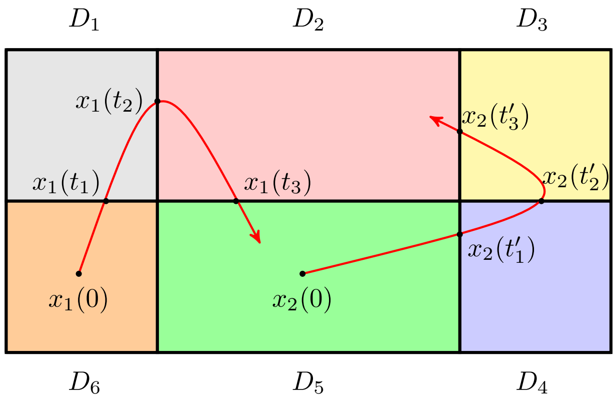

Fig. 1: An example of two agents performing in a partitioned workspace.

where $w_i(\mu) = L_i(r_i(\mu)), \mu \geq 0, i \in \mathcal{V}$, is associated uniquely with the trajectory $x_i(t)$.

Definition 13. For each agent $i \in \mathcal{V}$ we define the *relaxed timed word* as:

$$\widetilde{w}_i^t = (w_i(0), \widetilde{\tau}_i(0))(w_i(1), \widetilde{\tau}_i(1))(w_i(2), \widetilde{\tau}_i(2)) \ldots, \quad (7)$$

where $w_i(\mu) = L_i(r_i(\mu)), \widetilde{\tau}_i(\mu) \in [\tau_i(\mu), \tau_i(\mu+1)), \forall\ \mu \geq 0$.

The time stamp $\tau_i(0) = \widetilde{\tau}_i(0) = t_0, i \in \mathcal{V}$ models the initial starting time of the agents. The time stamps $\tau_i(\mu), \mu \geq 1$ models the exact time in which the agent i crosses the boundary of the regions $r_i(\mu-1)$ and $r_i(\mu)$. The time stamps $\widetilde{\tau}_i(\mu)$ model a time instant in which the agent i is in the region $r_i(\mu)$ of the workspace (see Example 1 below). The specification task φ_i given as an MITL formula over the set of atomic propositions Σ_i, represents desired tasks that are imposed to each agent $i \in \mathcal{I}$. We say that a trajectory $x_i(t)$ *satisfies* a formula φ_i given in MITL over the set Σ_i, and we formally write:

$$x_i(t) \models \varphi_i, \forall t \geq 0,$$

if and only if there exists a *relaxed timed word* $\widetilde{w}_i^t$ that complies with $x_i(t)$ and satisfies φ_i according to the semantics of MITL in 11.

Example 1. Consider $N = 2$ agents performing in the partitioned environment of Fig. 1. Both agents have the ability to pick up, deliver and throw two different balls. Their sets of atomic propositions are $\Sigma_1 = \{\text{pickUp1}, \text{deliver1}, \text{throw1}\}$ and $\Sigma_2 = \{\text{pickUp2}, \text{deliver2}, \text{throw2}\}$, respectively, and satisfy $\Sigma_1 \cap \Sigma_2 = \emptyset$. Three points of the agents' trajectories that belong to different regions with different atomic propositions are captured. Assume that $t_1 < t'_1 < t_2 < t_2 < t'_2 < t_3 < t'_3$. The trajectories $x_1(t), x_2(t), t \geq 0$ are depicted with the red lines. According to Assumption 4, the partition $D = \{D_\ell\}_{\ell \in \mathbb{I}} = \{D_1, \ldots, D_6\}$ is given where $\mathbb{I} = \{1, \ldots, 6\}$ respects the labeling functions $\Lambda_i, L_i, i \in \{1, 2\}$. In particular, it holds that:

$$\begin{aligned}
&\Lambda_1(x_1(t)) = L_1(r_1(0)) = \{\text{pickUp1}\}, \mathrm{t} \in [0, \mathrm{t}_1),\\
&\Lambda_1(x_1(t)) = L_1(r_1(1)) = \{\text{throw1}\}, \mathrm{t} \in [\mathrm{t}_1, \mathrm{t}_2),\\
&\Lambda_1(x_1(t)) = L_1(r_1(2)) = \{\text{deliver1}\}, \mathrm{t} \in [\mathrm{t}_2, \mathrm{t}_3),\\
&\Lambda_1(x_1(t)) = L_1(r_1(3)) = \emptyset, t \geq t_3.\\
&\Lambda_2(x_2(t)) = L_2(r_2(0)) = \{\text{pickUp2}\}, \mathrm{t} \in [0, \mathrm{t}'_1),\\
&\Lambda_2(x_2(t)) = L_2(r_2(1)) = \{\text{deliver2}\}, \mathrm{t} \in [\mathrm{t}'_1, \mathrm{t}'_2),\\
&\Lambda_2(x_2(t)) = L_2(r_2(2)) = \{\text{throw2}\}, \mathrm{t} \in [\mathrm{t}'_2, \mathrm{t}'_3),\\
&\Lambda_2(x_2(t)) = L_2(r_2(3)) = \emptyset, t \geq t'_3.
\end{aligned}$$

By the fact that $w_i(\mu) = L(r_i(\mu)), \forall\ i \in \{1, 2\}, \mu \in \{1, 2, 3\}$, the corresponding individual timed words are given as:

$$\begin{aligned}
w_{x_1}^t &= (\{\text{pickUp1}\}, 0)(\{\text{throw1}\}, \mathrm{t}_1)(\{\text{deliver1}\}, \mathrm{t}_2)(\emptyset, \mathrm{t}_3),\\
w_{x_2}^t &= (\{\text{pickUp2}\}, 0)(\{\text{deliver2}\}, \mathrm{t}'_1)(\{\text{throw2}\}, \mathrm{t}'_2)(\emptyset, \mathrm{t}'_3).
\end{aligned}$$

According to (7), two two relaxed timed words (depicted with red in Fig. 1) are given as:

$$\begin{aligned}
w_1^t = (&\{\text{pickUp1}\}, \widetilde{\tau}_1(0))(\{\text{throw1}\}, \widetilde{\tau}_1(1))\\
&(\{\text{deliver1}\}, \widetilde{\tau}_1(2))(\emptyset, \widetilde{\tau}_1(3)),\\
w_2^t = (&\{\text{pickUp2}\}, \widetilde{\tau}_2(0))(\{\text{deliver2}\}, \widetilde{\tau}_2(1))\\
&(\{\text{throw2}\}, \widetilde{\tau}_2(2))(\emptyset, \widetilde{\tau}_2(3)).
\end{aligned}$$

The time stamps $\widetilde{\tau}_1(\mu), \widetilde{\tau}_2(\mu), \mu \in \{1, 2, 3\}$ should satisfy the following conditions:

$$\begin{aligned}
&\widetilde{\tau}_1(0) \in [\tau_1(0), \tau_1(1)) = [0, t_1),\\
&\widetilde{\tau}_1(1) \in [\tau_1(1), \tau_1(2)) = [t_1, t_2),\\
&\widetilde{\tau}_1(2) \in [\tau_1(2), \tau_1(3)) = [t_2, t_3),\\
&\widetilde{\tau}_1(3) \in [\tau_1(3), \cdot) = [t_3, \cdot),\\
&\widetilde{\tau}_2(0) \in [\tau_2(0), \tau_2(1)) = [0, t_1),\\
&\widetilde{\tau}_2(1) \in [\tau_2(1), \tau_2(2)) = [t_1, t_2),\\
&\widetilde{\tau}_2(2) \in [\tau_2(2), \tau_2(3)) = [t_2, t_3),\\
&\widetilde{\tau}_2(3) \in [\tau_2(3), \cdot) = [t_3, \cdot).
\end{aligned}$$

C. Problem Statement

We can now formulate the problem treated in this paper as follows:

Problem 1. Given N agents operating in the bounded workspace $W \subseteq \mathbb{R}^2$, their initial positions $x_1(t_0), \ldots, x_N(t_0)$, their dynamics as in (2), a time step $T > h > 0$, N task specification formulas $\varphi_1, \ldots, \varphi_N$ expressed in MITL over the sets of services $\Sigma_1, \ldots, \Sigma_N$, respectively, a partition of the workspace W into hexagonal regions $\{D_\ell\}_{\ell \in \mathbb{I}}$ with side length R as in Assumption 4 and the labeling functions $\Lambda_1, \ldots, \Lambda_N, L_1, \ldots, L_N$, as in (5), (6), assign control laws $u_1, \ldots, u_N$ to each agent $1, \ldots, N$, respectively, such that the connectivity between the agents that belong to the neighboring sets $\mathcal{N}_1, \ldots, \mathcal{N}_N$ is maintained, as well as each agent fulfills its individual MITL specification $\varphi_1, \dots, \varphi_N$, respectively, i.e., $x_1(t) \models \varphi_1, \dots, x_N(t) \models \varphi_N, \forall\, t \in \mathbb{R}_{\geq 0}$.

Remark 3. The initial positions $x_1(t_0), \dots, x_N(t_0)$ should be such that the agents which are required to remain connected for all times need to satisfy the inequality $\|x_i(t_0) - x_{i'}(t_0)\| < 2\underline{r}, i, i' \in \mathcal{V}, i \neq i'$.

Remark 4. It should be noted that, in this work, the dependencies between the agents are induced through the coupled dynamics (2) and not in the discrete level, by allowing for couplings between the services (i.e., $\Sigma_i \cap \Sigma_j \neq \emptyset$, for some $i, j \in \mathcal{V}$). Hence, even though the agents do not share atomic propositions, the constraints on their motion due to the dynamic couplings and the connectivity maintenance specifications may restrict them to fulfill the desired high-level tasks. Treating additional couplings through individual atomic propositions in the discrete level is a topic of current work.

Remark 5. In our previous work on the multi-agent controller synthesis framework under MITL specifications [15], the multi-agent system was considered to have fully-actuated dynamics. The only constraints on the system were due to the presence of time constrained MITL formulas. In the current framework, we have two types of constraints: the constraints due to the coupling dynamics of the system (2), which constrain the motion of each agent, and, the timed constraints that are inherently imposed from the time bounds of the MITL formulas. Thus, there exist formulas that cannot be satisfied either due to the coupling constraints or the time constraints of the MITL formulas. These constraints, make the procedure of the controller synthesis in the discrete level substantially different and more elaborate than the corresponding multi-agent LTL frameworks in the literature ([3], [4], [7], [8]).

IV. Proposed Solution

In this section, a systematic solution to Problem 1 is introduced. Our overall approach builds on abstracting the system in (2) through a WTS for each agent and exploiting the fact that the timed runs in the i-th WTS project onto the trajectories of agent i while preserving the satisfaction of the individual MITL formulas $\varphi_i, i \in \mathcal{V}$. In particular, the following analysis is performed:

1) We propose a novel decentralized abstraction technique for the multi-agent system, i.e., discretization of the time into time steps T for the given partition $D = \{D_\ell\}_{\ell \in \mathbb{I}}$, such that the motion of each agent is modeled by a WTS $\mathcal{T}_i,\ i \in \mathcal{I}$ (Section IV-A). We adopt here the technique of designing Nonlinear Model Predictive Controllers (NMPC), for driving the agents between neighboring regions.
2) A three-step automated procedure for controller synthesis which serves as a solution to Problem 1 is provided in Section IV-B.
3) Finally, the computational complexity of the proposed approach is discussed in Section IV-C.

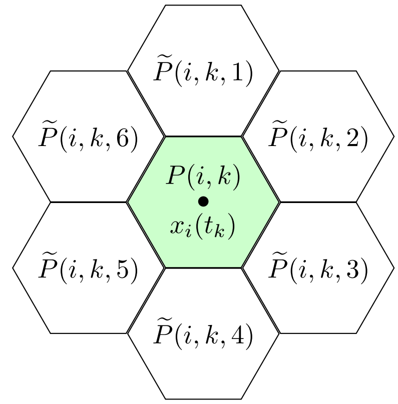

Fig. 2: Illustration of agent i occupying region $P(i,k)$, depicted by green, at time $t_k = t_0 + kT$ with $\bar{P}(i,k) = \bigcup_{\widetilde{\ell} \in \mathbb{L}} \widetilde{P}(i,k,\widetilde{\ell})$ being the set of regions that the agent can transit at exactly time T.

The next sections provide the proposed solution in detail.

A. Discrete System Abstraction

In this section we provide the abstraction technique that is designed in order to capture the dynamics of each agent into WTSs. Thereafter, we work completely at discrete level, which is necessary in order to solve Problem 1.

1) Workspace Geometry: Consider an enumeration $\mathbb{I}$ of the regions of the workspace, the index variable $\ell \in \mathbb{I}$ and the given time step T. The time step T models the time duration that each agent needs to transit between two neighboring regions of the workspace. Consider also a timed sequence:

$$\mathcal{S} = \{t_0, t_1 = t_0 + T, \dots, t_k = t_0 + kT, \dots\}, k \in \mathbb{N}. \quad (8)$$

S models the time stamps in which the agents are required to occupy different neighboring regions. For example, if at time t_k agent i occupies region D_ℓ, at the next time stamp $t_k + T$ is required to occupy a neighboring region of D_ℓ. The agents are always forced to change region for every different time stamp. Let us define the mapping:

$$P : \mathcal{V} \times \mathbb{N} \to D,$$

which denotes the fact that the agent $i \in \mathcal{V}$, at time instant

$$t_k = t_0 + kT, k \in \mathbb{N},$$

occupies the region $D_{\ell_i} \in D$ for an index $\ell_i \in \mathbb{I}$. Define the mapping:

$$\widetilde{P} : \mathcal{V} \times \mathbb{N} \times \mathbb{L} \to D.$$

where $\mathbb{L} = \{1, \dots, 6\}$. By $\widetilde{P}(i,k,\widetilde{\ell}), \widetilde{\ell} \in \mathbb{L}$ we denote one and only one out of the six neighboring regions of region $P(i,k)$ that agent i occupies at time t_k. Define also by $\bar{P}(i,k)$ the union of all the six neighboring regions of region $P(i,k)$, i.e.,

$$\bar{P}(i,k) = \bigcup_{\widetilde{\ell} \in \mathbb{L}} \widetilde{P}(i,k,\widetilde{\ell}),$$

with $|\bar{P}(i,k)| = 6$. An example of agent i being at the region $P(i,k)$ along with its neighboring regions is depicted in Fig. 2.

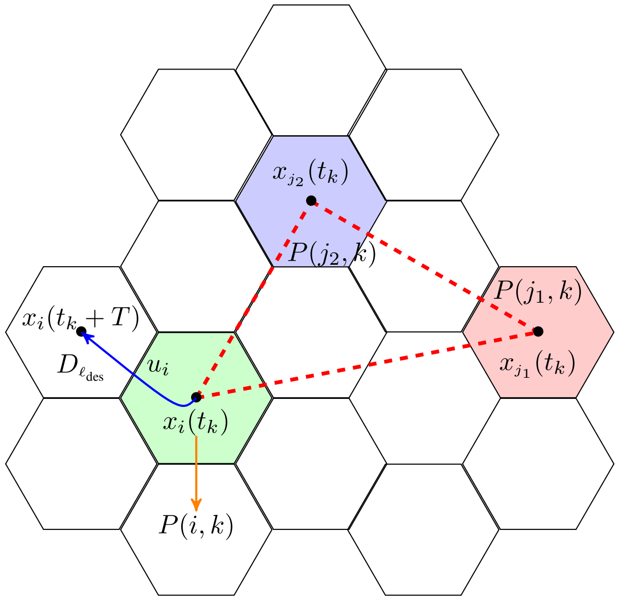

Fig. 3: Illustration of three connected agents i, j_1, j_2. The agents are occupying the regions $P(i,k) = D_{\ell_i}, P(j_1,k) = D_{\ell_{j_1}}$ and $P(j_2,k) = D_{\ell_{j_2}}$ at time $t_k = t_0 + kT$, depicted by green, red and blue color, respectively. Their corresponding neighboring regions $\bar{P}(i,k), \bar{P}(j_1,k)$ and $\widetilde{P}(j_2,k,\widetilde{\ell}), \widetilde{\ell} \in \{4,5,6\}$, respectively, are also depicted. $\widetilde{P}(i,k,6) = D_{\ell_{\rm des}}$ is the desired region in which agent i needs to move at time T by applying a decentralized control law $u_i(x_i, x_{j_1}, x_{j_2})$.

We start by giving a graphical example for the abstraction technique that will be adopted in this work. Consider agent i occupying the green region $P(i,k) = D_{\ell_i}$ at time $t_k = t_0 + kT$ and let its neighbors j_1, j_2 occupying the red and blue regions $P(j_1,k) = D_{\ell_{j_1}}, P(j_2,k) = D_{\ell_{j_2}}$, respectively, as is depicted in Fig. 3. The neighboring regions $\bar{P}(i,k), \bar{P}(j_1,k)$ and $\widetilde{P}(j_2,k,\widetilde{\ell}), \widetilde{\ell} \in \{4,5,6\}$ for agent i, j_1, j_2, respectively, are also depicted. All the agents start their motion at time t_k simultaneously. The goal is to design a decentralized feedback control law $u_i(x_i, x_{j_1}, x_{j_2})$, that drives agent i in the neighboring region $D_{\ell_{\rm des}}$ exactly at time T, *regardless of the transitions of its neighbors to their neighboring regions*. If such controller exists, it is stored in the memory a new search for the next region is performed. This procedure is repeated for all possible neighboring regions i.e., six times, and for all the agents. For the example of Fig. 3, the procedure is performed 6^3 times (six times for each agent). With this procedure, we are able to: 1) synchronize the agents so that each of them knows at every time step T its position in the workspace as well as the region that occupies; 2) know which controller brings each agent in its desired region for any possible choice of controllers of its corresponding neighbors. We will hereafter present a formal approach of this procedure. We will hereafter present a formal approach of this procedure.

2) Decentralized Controller Specification: Consider a time interval $[t_k, t_k + T]$. We state here the specifications that a decentralized feedback controller $u_i(x_i, \bar{x}_i)$ needs to guarantee so as agent i to have a *well-defined transition* between two neighboring regions within the time interval $[t_k, t_k + T]$.

(S1) The controller needs to take into consideration the dynamics (2) and the constraints that are imposed by the bounds of Assumption 1.

(S2) Agent i should move from one region $P(i,k) \in D$ to a neighboring region $\widetilde{P}(i,k,\widetilde{\ell})$, without intersecting other regions, irrespectively of which region its neighbors are moving to. Thus, since the duration of the transition is T, it is required that $x_i(t_k) \in P(i,k)$, $x_i(t_k+T) \in \widetilde{P}(i,k,\widetilde{\ell})$ and $x_i(t) \in P(i,k) \cup \widetilde{P}(i,k,\widetilde{\ell}), t \in (t_k, t_k+T)$. The neighbors of agent i will move also to exactly one of their corresponding neighboring regions.

Remark 6. The reason for imposing the aforementioned constraints is due to the need of imposing safety specifications to the agents. Thus, it is required to be guaranteed that the agents will not cross more than one neighboring region within the duration of a transition T.

3) Error Dynamics: Let us define by $x_{i,k,\widetilde{\ell},\rm des} \in \widetilde{P}(i,k,\widetilde{\ell})$ a reference point of the desired region $\widetilde{P}(i,k,\ell)$ which agent i needs to occupy at time $t_k + T$. Define also by:

$$e_i(t) = x_i(t) - x_{i,k,\widetilde{\ell},\rm des}, t \in [t_k, t_k + T], \tag{9}$$

the error which the controller u_i needs to guarantee to become zero in the time interval $t \in [t_k, t_k + T]$. Then, the *nominal error dynamics* are given by:

$$\dot{e}_i(t) = g_i(e_i(t), \bar{x}_i(t), u_i(t)), t \in [t_k, t_k + T], \tag{10}$$

with initial condition $e_i(t_k) = x_i(t_k) - x_{i,k,\widetilde{\ell},\rm des}$, where:

$$g_i(e_i(t), \bar{x}_i(t), u_i(t)) = f_i(e_i(t) + x_{i,k,\widetilde{\ell},\rm des}, \bar{x}_i(t)) + u_i(t).$$

Property 1. According to Assumption (2), at every time $s \in [t_k, t_k + T]$, with $t_k = t_0 + kT$, the error $e_i(s)$ of the state of agent i is upper bounded by:

$$\|e_i(s)\| \le \|e_i(t_k)\| + (s - t_k)(M + u_{\max}), i \in \mathcal{V}. \tag{11}$$

Proof. The proof can be found in Appendix I. ☐

4) State Constraints: Before defining the ROCP we state here the state constraints that are imposed to the state of each agent. Define the set:

$$\begin{aligned} X_i = \{ & x_i \in W, \bar{x}_i \in W^{N_i} : \\ & \|f_i(x_i, \bar{x}_i)\| \le M, \|x_i - x_j\| < \underline{r}, \forall j \in \mathcal{N}_i(0), \\ & x_i \in P(i,k) \cup \widetilde{P}(i,k,\widetilde{\ell}), \widetilde{\ell} \in \mathbb{L}\}, \end{aligned}$$

as the set that captures the state constraints of agent i. The first constraint in the set X stands for the bound of Assumption 2; the second one stands for the connectivity requirement of agent i with all its neighbors; the last one stands for the requirement each agent to transit from one region to exactly one desired neighboring region. In order to translate the constraints that are dictated for the state $x_i(t)$ into constraints regarding the error state $e_i(t)$ from (10), define the set $E_i = X_i \oplus (-x_{i,k,\widetilde{\ell},\rm des})$. Then, the following implication holds: $x_i \in X_i \Rightarrow e_i \in E_i$.

5) Control Design: This subsection concerns the control design regarding the transition of agent i to one neighboring region $\widetilde{P}(i,k,\widetilde{\ell})$, for some $\widetilde{\ell} \in \mathbb{L}$. The abstraction design, however, concerns all the neighboring regions $\bar{P}(i,k)$, for which we will discuss in the next subsection.

The timed sequence $\mathcal{S}$ consists of intervals of duration T. Within every time interval $[t_k, t_k+T]$, each agent needs to be at time t_k in region $P(i,k)$ and at time t_k+T in a neighboring region $\widetilde{P}(i,k,\widetilde{\ell}), \widetilde{\ell} \in \mathbb{L}$. We assume that T is related to the sampling time h according to: $T = mh, m \in \mathbb{N}$. Therefore, within the time interval $[t_k, t_k+T]$, there exists $m+1$ sampling times. By introducing the notation $t_{k_z} \triangleq t_k + zh, \forall z \in \mathbb{M} \triangleq \{0, \dots, m\}$, we denote by $\{t_{k_z}\}_{z \in \mathbb{M}}$ the sampling sequence within the interval $[t_k, t_k+T]$. Note that $t_{k_0} = t_k$ and $t_{k_m} = t_k + T$. The indexes k, z stands for the interval and for the sampling times within this interval, respectively. As it will be presented hereafter, at every sampling time $t_{k_z}, z \in \mathbb{M}$, each agents solves a ROCP.

Our control design approach is based on Nonlinear Model Predictive Control (NMPC). NMPC has been proven to be efficient for systems with nonlinearities and state/input constraints. For details about NMPC we refer the reader to [41]–[50]. We propose here a sampled-data NMPC with decreasing horizon in order to design a controller that respects the desired specifications and guarantees the transition between regions at time T. In the proposed sampled-data NMPC, an open-loop Robust Optimal Control Problem (ROCP) is solved at every discrete sampling time instant $t_{k_z}, z \in \mathbb{M}$ based on the current error state information $e_i(t_{k_z})$. The solution is an optimal control signal $\hat{u}_i(t)$, for $t \in [t_{k_z}, t_{k_z} + T_z]$, where T_z is defined as follows.

Definition 14. A *decreasing horizon policy* is defined by:

$$T_z = T - zh, z \in \mathbb{M}. \tag{12}$$

This means that at every time sample t_{k_z} in which the ROCP is solved, the horizon is decreased by a sampling time h. The specific policy is adopted in order to enforce the controllers u_i to guarantee that agent i will reach the desired neighboring region at time T. (12) implies also that $t_{k_z} + T_z = t_k + T, \forall z \in \mathbb{M}$ A graphical illustration of the presented time sequences is given in Fig. 4.

The *open-loop input signal* is applied in between the sampling instants and is given by the solution of the following Robust Optimal Control Problem (ROCP): $\mathcal{O}(k, x_i(t), \bar{x}_i(t), P(i,k), \widetilde{\ell}, x_{i,k,\widetilde{\ell},\text{des}}), t \in [t_{k_z}, t_{k_z} + T_z]$, which is defined as:

$$\min_{\hat{u}_i(\cdot)} J_i(e_i(t_{k_z})), \hat{u}_i(\cdot)) =$$

$$\min_{\hat{u}_i(\cdot)} \left\{ V_i(\hat{e}_i(t_{k_z} + T_z)) + \int_{t_{k_z}}^{t_{k_z}+T_z} \Big[F_i(\hat{e}_i(s), \hat{u}_i(s)) \Big] ds \right\} \tag{13a}$$

subject to:

$$\dot{\hat{e}}_i(s) = g_i(\hat{e}_i(s), \hat{\bar{x}}_i(s), \hat{u}_i(s)), \hat{e}_i(t_{k_z}) = e_i(t_{k_z}), \tag{13b}$$

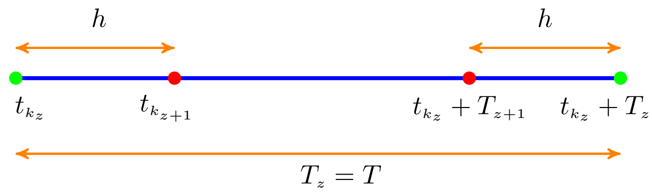

Fig. 4: The prediction horizon of the ROCP along with the times $t_{k_z} < t_{k_{z+1}} < t_{k_z} + T_{z+1} < t_{k_z} + T_z$, with $t_{k_z} = t_{k_z} + zh$ and $T_{k_z} = T - zh, z \in \mathbb{M}$.

$$\hat{e}_i(s) \in E^i_{s-t_{k_z}}, \hat{u}_i(s) \in \mathcal{U}_i, s \in [t_{k_z}, t_{k_z} + T_z], \tag{13c}$$

$$\hat{e}_i(t_{k_z} + T_z) \in \mathcal{E}_i. \tag{13d}$$

The ROCP has as inputs the terms $k, x_i(t), \bar{x}_i(t), P(i,k), \widetilde{\ell}, x_{i,k,\widetilde{\ell},\text{des}}$, for time $t \in [t_{k_z}, t_{k_z} + T_z]$. We will explain hereafter all the terms appearing in the ROCP problem (13a)-(13d). By hat $\hat{(\cdot)}$ we denote the predicted variables (internal to the controller), corresponding to the system (10) i.e., $\hat{e}_i(\cdot)$ is the solution of (13b) driven by the control input $\hat{u}_i(\cdot) : [t_{k_z}, t_{k_z} + T_z] \to \mathcal{U}_i$ with initial condition $\hat{e}_i(t_{k_z}) = e_i(t_{k_z})$. The set $E^i_{s-t_{k_z}}$ is a subset of E_i and will be explicitly defined later.

Remark 7. In sampled-data NMPC bibliography an ROCP is defined over the time interval $s \in \{t_i, t_{i+1} = t_i + h, \dots, t_i + T\}$, where T is the prediction horizon. Due to the fact that we have denoted by i the agents, and the fact that the ROCP is solved for every time interval, we use the notation $s \in \{t_{k_z} = t_k, t_{k_{z+1}} = t_k + h, \dots, t_{k_z} + T_z = t_{k_z} + T\}$, instead. The indexes k, z stands for the interval and for the sampling time, respectively. A graphical illustration of the presented time sequence is given in Fig. 4.

Remark 8. Note that the predicted values are not the same with the actual closed-loop values due to the fact that agent i, can not know the estimation of the trajectories of its neighbors $\hat{\bar{x}}$, within a predicted horizon. Thus, the term $\hat{\bar{x}}$ is treated as a disturbance to the *nominal system* (10).

The term $F_i : E_i \times \mathcal{U}_i \to \mathbb{R}_{\geq 0}$, stands for the *running cost*, and is chosen as:

$$F_i(e_i, u_i) = e_i^\top Q_i e_i + u_i^\top R_i u_i,$$

where $Q_i = \text{diag}\{q_{i_1}, q_{i_2}\}, R_i = \text{diag}\{\xi_{i_1}, \xi_{i_2}\}$, with $q_{i_\zeta} \in \mathbb{R}_{\geq 0}, \xi_{i_\zeta} \in \mathbb{R}_{>0}, \zeta \in \{1, 2\}$. For the running cost, it holds that $F_i(0,0) = 0$, as well as:

$$\underline{m}_i \|e_i\|^2 \leq F_i(e_i, u_i) \leq \bar{m}_i \|e_i\|^2, \tag{14}$$

where $\underline{m}_i, \bar{m}_i$ will be defined later. Note that $\underline{m}_i \|e_i\|^2$ is $\mathcal{K}$ function, according to Definition 2.

Lemma 1. *The running cost function $F_i(e_i, u_i)$ is Lipschitz continuous in $E_i \times \mathcal{U}_i$, with Lipschitz constant:*

$$L_{F_i} = 2\bar{\varepsilon}_i \sigma_{\max}(Q_i),$$

where:

$$\bar{\varepsilon}_i = \sup_{e_i \in E_i} \{\|e_i\|\},$$

for all $e_i \in E_i, u_i \in \mathcal{U}_i$.

Proof. The proof can be found in Appendix II. □

Note that, according to (11), the terms $\|e_i\|$ are bounded, for all $i \in \mathcal{V}$. The terms $V_i : E_i \to \mathbb{R}_{>0}$ and $\mathcal{E}_i \subseteq E_i$ are the *terminal penalty cost* and *terminal set*, respectively, and are used to enforce the stability of the system. The terminal cost is given by:

$$V_i(e_i(t)) = e_i(t)^\top P_i e_i(t).$$

where $P_i = \mathrm{diag}\{p_{i_1}, p_{i_2}\}$, with $p_{i_\zeta} \in \mathbb{R}_{>0}, \zeta \in \{1,2\}$. We choose $\underline{m}_i = \{q_{i_1}, q_{i_2}, \xi_{i_1}, \xi_{i_2}\}$ and $\bar{m}_i = \{q_{i_1}, q_{i_2}, \xi_{i_1}, \xi_{i_2}\}$.

The solution of the nominal model (10) at time $s \in [t_{k_z}, t_{k_z} + T_z]$, starting at time t_{k_z} from an initial condition $e_i(t_{k_z})$, applying a control input $u_i : [t_{k_z}, s] \to \mathcal{U}_i$ is denoted by:

$$e_i(s; u_i(\cdot), e_i(t_{k_z})), s \in [t_{k_z}, t_{k_z} + T_z].$$

The predicted state of the system (10) at time $s \in [t_{k_z}, t_{k_z} + T_z]$ is denoted by:

$$\hat{e}_i(s; u_i(\cdot), e_i(t_{k_z})), s \in [t_{k_z}, t_{k_z} + T_z],$$

and it is based on the measurement of the state $e_i(t_{k_z})$ at time t_{k_z}, when a control input $u_i(\cdot; e_i(t_{k_z}))$ is applied to the system (10) for the time period $[t_{k_z}, s]$. Thus, it holds that:

$$e_i(s) = \hat{e}_i(s; u_i(\cdot), e_i(s)), s \in [t_{k_z}, t_{k_z} + T_z]. \tag{15}$$

The state measurement enters the system via the initial condition of (13b) at the sampling instant, i.e. the system model used to predict the future system behavior is initialized by the actual system state. The solution of the ROCP (13a)-(13d) at time t_{k_z} provides an optimal control input denoted by $\hat{u}_i^\star(t; e(t_{k_z}))$, for $t \in [t_{k_z}, t_{k_z} + T_z]$. It defines the open-loop input that is applied to the system until the next sampling instant $t_{k_{z+1}}$:

$$u_i(t; e_i(t_i)) = \hat{u}_i^\star(t_{k_z}; e_i(t_{k_z})), t \in [t_{k_z}, t_{k_{z+1}}). \tag{16}$$

The corresponding *optimal value function* is given by:

$$J_i^\star(e_i(t_{k_z})) \triangleq J_i(e_i(t_{k_z}), \hat{u}_i^\star(\cdot; e_i(t_{k_z}))). \tag{17}$$

with $J_i(\cdot)$ as is given in (13a). The control input $u_i(t; e_i(t_{k_z}))$ is of the feedback form, since it is recalculated at each sampling instant using the new state information. Define an admissible control input as:

Definition 15. A control input $u_i : [t_{k_z}, t_{k_z} + T_z] \to \mathbb{R}^2$ for a state $e(t_{k_z})$ is called *admissible*, if all the following hold:

1) $u_i(\cdot)$ is piecewise continuous;
2) $u_i(s) \in \mathcal{U}_i, \forall\ s \in [t_{k_z}, t_{k_z} + T_z]$;
3) $e_i(s; u_i(\cdot), e(t_{k_z})) \in E_i, \forall\ s \in [t_{k_z}, t_{k_z} + T_z]$;
4) $e_i(T_z; u_i(\cdot), e(t_{k_z})) \in \mathcal{E}_i$;

Property 2. For the given hexagonal regions with side length R, the radius of the inscribed circle is given by $\underline{r}_h = \frac{\sqrt{3}}{2} R$ (two inscribed circles for the given regions are depicted with orange in Fig. 5). Thus, according to Fig. 5, an upper bound of the norm of differences between the actual position x_j and the estimated position $\hat{x}_j$ of the agent's i neighbors states, is given by:

$$\|x_j - \hat{x}_j\| \le 4\underline{r}_h = 2\sqrt{3}R, j \in \mathcal{N}_i, \tag{18}$$

due to the fact that each agent can transit at most to a neighboring region, according to the constraint set X_i.

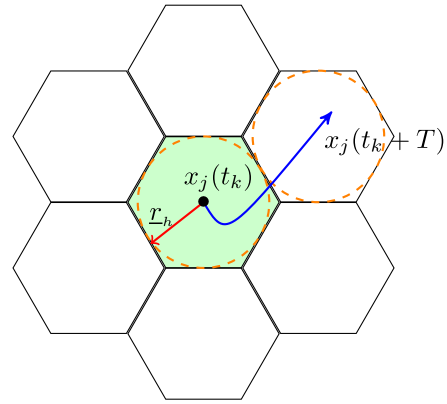

Fig. 5: Illustration of agent j occupying region $P(j,k)$, depicted by green, at time $t_k = t_0 + kT$ along with the regions $\bar{P}(j,k)$. It is desired for agent j to move to region $\widetilde{P}(j,k,2)$ at precise time T. The inscribed circle of regions $P(j,k), \widetilde{P}(j,k,2)$ are depicted with dashed orange color. The radius of the inscribed circle of the depicted hexagons is given by $\underline{r}_h = \frac{\sqrt{3}}{2}R$. By taking into consideration that each agent is moving at most to one neighboring region, according to the constraint set X_j, the following holds: $\sup\{\|x - y\| : x \in P(j,k), y \in \bar{P}(j,k)\} = 4\underline{r}_h = 2\sqrt{3}R$.

Lemma 2. *In view of Assumptions 2, 3, the difference between the actual measurement $e_i(s) = e_i(s; u_i(s; e_i(t_{k_z})), e_i(t_{k_z}))$ at time $s \in [t_{k_z}, t_{k_z} + T_z]$ and the predicted state $\hat{e}_i(s; u_i(s; e_i(t_{k_z})), e_i(t_{k_z}))$ at the same time under the same control law $u_i(s; e_i(t_{k_z}))$, starting at the same initial state $e_i(t_{k_z})$, is upper bounded by:*

$$\|e_i(s) - \hat{e}_i(s; u_i(s; e_i(t_{k_z})), e_i(t_{k_z}))\| \le \rho_i(s - t_{k_z}), \tag{19}$$

where $\rho_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$, with:

$$\rho_i(y) = \min\Big\{\widetilde{\rho}_i\left[e^{L_i y} - 1\right], \\ 2\|e_i(t_{k_z})\| + 2y(M + u_{\max})\Big\}, \tag{20}$$

and

$$\widetilde{\rho}_i = \frac{2\sqrt{3}R\bar{L}_i N_i}{L_i}. \tag{21}$$

Proof. The proof can be found in Appendix III. □

The satisfaction of the constraint on the state along the prediction horizon depends on the future evolution of the neighboring agents trajectories. Under Assumptions (3), (2) of Lipschitz continuity and bounds of the nominal model, respectively, it is possible to compute a bound on the future effect of the disturbance on the system as is given by Lemma 2. Then, by considering this effect on the state constraint on the nominal prediction, it is possible to guarantee that the evolution of the real state of the system will be admissible all the time. In view of latter, the state constraint set E of the standard NMPC formulation, is being replaced by a restricted constrained set $E^i_{s-t_{k_z}} \subseteq E_i$ in (13c). This state constraint's tightening for the nominal system (10) is a key ingredient of the robust NMPC controller and guarantees that the evolution of the real system will be admissible. Authors in [50], [51] has considered such a Robust NMPC formulation. The restricted constrained set is then defined as $E^i_{s-t_{k_z}} = E_i \sim B^i_{s-t_{k_z}}$, where:

$$B^i_{s-t_{k_z}} = \left\{e_i \in \mathbb{R}^2 : \|e_i(s)\| \leq \rho_i(s-t_i)\right\}, s \in [t_{k_z}, t_{k_z}+T_z].$$

Property 3. For every $s \in [t_{k_z}, t_{k_z}+T_z]$, we have that if

$$\hat{e}_i(s; u_i(s; e(t_{k_z})), e_i(t_{k_z})) \in E^i_{s-t_{k_z}} = E_i \sim B^i_{s-t_{k_z}} \subseteq E_i,$$

then the real state satisfies the constraint E_i, i.e., $e_i(s) \in E_i$.

Proof. The proof can be found in Appendix IV. □

For the feasibility and convergence proofs of the ROCP the following assumptions are required.

Assumption 5. Assume that there exists a *local stabilizing controller* $u_{f,i} = \kappa_i(e_i) \in \mathcal{U}_i$ satisfying:

$$\frac{\partial V_i}{\partial e_i}\left[g_i(e_i, \bar{x}_i, \kappa_i(e_i))\right] + F_i(e_i, \kappa_i(e_i)) \leq 0, \forall\, e_i \in \Phi_i, \quad (22)$$

where Φ_i is a set given by:

$$\Phi_i \triangleq \{e_i \in \mathbb{R}^2 : V_i(e_i) \leq \alpha_{1,i}\}, \alpha_{1,i} > 0,$$

such that:

$$\Phi_i \subseteq \mathbb{E}_i \triangleq \{e_i \in E^i_{T_z} : \kappa_i(e_i) \in \mathcal{U}_i\},$$

where $E^i_{T_z} = E_i \sim B^i_{T_z}$.

Lemma 3. *The terminal penalty function $V_i(\cdot)$ is Lipschitz in Φ_i, with Lipschitz constant $L_{V,i} = 2\sigma_{\max}(P_i)\sqrt{\frac{\alpha_{1,i}}{\lambda_{\min}(P_i)}}$, for all $e_i(t) \in \Phi_i$.*

Proof. The proof can be found in Appendix V. □

Once the set Φ_i is computed, the terminal constraint set $\mathcal{E}_i$ is given by the following. Supposing that Assumption 5 holds. Then, by choosing: $\mathcal{E}_i = \{e_i \in \mathbb{R}^2 : \|e_i\| \leq \sqrt{\frac{\alpha_{2,i}}{\lambda_{\min}(P_i)}} < r_h\}$, with $\alpha_{2,i} \in (0, \alpha_{1,i})$, we guarantee the following: 1) $\mathcal{E}_i \subseteq \widetilde{P}(i,k,\widetilde{\ell})$, i.e. the terminal set is a subset of the desired neighboring region; 2) for all $e_i \in \Phi_i$ it holds that $g_i(e_i, \kappa_i(e_i)) \in \mathcal{E}_i$.

The following two lemmas are required in order to prove the basic Theorem or this paper.

Lemma 4. *Let $s \geq t_{k_{z+1}}$, $x \in E^i_{s-t_{k_z}}$ and $y \in \mathbb{R}^2$ such that: $\|x-y\| \leq \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h)$, as ρ_i is given in Lemma 2. Then, it holds that $y \in E^i_{s-t_{k_{z+1}}}$.*

Proof. The proof can be found in Appendix VI. □

Lemma 5. *Let $s \geq t_{k_z}$. The difference between two estimated trajectories $\hat{e}_i(s; u_i(\cdot), e_i(t_{k_{z+1}})), \hat{e}_i(s; u_i(\cdot), e_i(t_{k_z}))$ at time s, starting from from initial points $t_{k_{z+1}}$, t_{k_z}, respectively, under the same control input $u_i(\cdot)$, is upper bounded by:*

$$\|\hat{e}_i(s; u_i(\cdot), e_i(t_{k_{z+1}})) - \hat{e}_i(s; u_i(\cdot), e_i(t_{k_z}))\| \leq \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h). \quad (23)$$

Proof. The proof can be found in Appendix VII. □

Theorem 2. *Suppose that Assumptions 1-5 hold. If the ROCP is feasible at time t_k, then, the closed loop system* (10) *of agent i, under the control input* (16)*, starting its motion at time $t_k = t_0 + kT$ from region $P(i,k)$, is Input to State Stable (ISS) (for ISS see [27]) and its trajectory converges to the admissible positively invariant terminal set $\mathcal{E}_i$ exactly at time $t_k + T$, if it holds that $\rho_i(T_z) \leq \bar{\rho}_i \triangleq \frac{\alpha_{1,1}-\alpha_{2,i}}{L_{V_i}}$.*

Proof. The proof consists of two parts: in the first part it is established that initial feasibility implies feasibility afterwards. Based on this result it is then shown that the error $e_i(t)$ converges to the terminal set $\mathcal{E}_i$. The *feasibility analysis* as well as the *convergence analysis* can be found in Appendix VIII. □

Assumption 5 is common in the NMPC literature. Many methodologies on how to compute Φ_i and controllers $u_{f,i} = \kappa_i(e_i)$, if they exist, have been proposed. We refer the reader to [44], [52]. Regarding the initial feasibility, numerical tools (e.g. [46]) can be utilized in order to solve the ROCP and check if the problem is feasible at time $t_k = t_{k_z}$.

Remark 9. The term $\bar{\rho}_i, i \in \mathcal{V}$ gives an upper bound on the deviation of the trajectories of the neighboring agents of agent i from their real values. If this bound is satisfied, agent i can transit between the corresponding two neighboring regions, provided the ROCP is feasible at t_{k_z}.

Remark 10. It should be noted that, due to the non-linear coupling terms $f_i(x_i, \bar{x}_i)$, the desired connectivity specifications and the bounds of Assumption 2, some of the ROCPs for $k \in N$ might not have a feasible solution. Let $i' \in \mathcal{V}, k' \in \mathbb{N}, \widetilde{\ell}' \in \mathbb{L}$ represent an agent i' that at time step $t_{k'} = t_0 + k'T$ is desired to transit from region $P(i', k')$ to region $\widetilde{P}(i', k', \widetilde{\ell}')$. If the ROCP $\mathcal{O}(k', x_{i'}(t), \bar{x}_{i'}(t), P(i', k'), \widetilde{\ell}', x_{i',k',\widetilde{\ell}',\text{des}}), t \in [t_{k'_z}, t_{k'_z} + T_z]$, has no solution, then there does not exist admissible controller that can drive agent i' from $P(i', k')$ to region $\widetilde{P}(i', k', \widetilde{\ell}')$. Our goal, through the proposed approach, is to seek all the possible solutions of the ROCP, which implies to seek for all possible transitions that will form later the individual WTS $\mathcal{T}_i$ of each agent. In this way, the resulting WTS $\mathcal{T}_i$ will capture the coupling dynamics (2) and the transition possibilities of agent i in the best possible way.

6) Generating the WTSs: Each agent $i \in \mathcal{V}$ solves the ROCP 13a-13d for every time interval $[t_{k_z}, t_{k_z} + T_z], k \in \mathbb{N}$, for all the desired neighboring regions $\widetilde{P}(i,k,\widetilde{\ell}), \widetilde{\ell} \in \mathbb{L}$. This procedure is performed by off-line simulation, i.e., at each sampling time $t_{k_z}, z \in \mathbb{M}$, each agent exchanges information about its new state with its neighbors and simulates the

Algorithm 1 CreateTransitionRelation($\cdot$)

Input: $i, x_i(t_0), \bar{x}_i(t_0)$;
Output: Transit; $\triangleright$ Matrix with regions\control inputs;

Transit $\leftarrow$ zeros($|\mathbb{I}|, 6$); k = 0; Flag = False;
List $\leftarrow$ {Point2Region($x_i(t_0)$)}; $\triangleright$ Initialize
while List $\neq \emptyset$ **do**
  for $p \in$ List **do** $\triangleright$ p is a region of the List;
    for $\widetilde{\ell} \in \mathbb{L}$ **do**
      $t \leftarrow$ Sampling($t_k, t_k + T$);
      for $t_{k_z} \in t, z \in \mathbb{M}$ **do**
        $(u_i^\star)_{k_z} \leftarrow$ OptSolve($k, x_i(t), \bar{x}_i(t), p, \widetilde{\ell}$);
        UpdateStates($x_i, \bar{x}_i$);
        if $(u_i^\star)_{k_z} = \emptyset$ **then** $\triangleright \nexists$ controller;
          Flag = True; $\triangleright$ search next region;
          break;
        end if
      end for
      if Flag = False **then**
        $u_i^\star \leftarrow \{(u_i^\star)_{k_z}\}_{z \in \mathbb{M}}$ $\triangleright$ u_i found
        Transit(p, ℓ) $\leftarrow u_i^\star$;
        List $\leftarrow$ List $\cup\ \widetilde{P}(i, k, \widetilde{\ell})$
      else
        Flag = False;
      end if
    end for
    List $\leftarrow$ List$\backslash p$;
    $k = k + 1$;
  end for
end while

dynamics (10). Between the sampling times the estimation $\hat{\bar{x}}_i$ is considered to be a disturbance, as discussed earlier.

Algorithm 1 provides the off-line procedure in order to generate the transition relation for each agent. At time t_0 each agent i calls the algorithm in order to compute all possible admissible controllers to all possible neighboring regions of the workspace. The term Transit, which is the output of the algorithm, is a matrix of control input sequences for all pairs of neighboring regions in the workspace, initialized at sequences of zeros. The function Point2Region($\cdot$) maps the point $x_i(t_k)$ to the corresponding region of the workspace. The function Sampling($\cdot$) takes as input the interval $[t_k, t_k + T]$ and returns the $m + 1$ samples of this interval. The notation $(u_i^\star)_{k_z}$ stands for the z-th element of the vector $(u_i^\star)$. The function OptSolve($k, x_i(t), \bar{x}_i(t), p, \widetilde{\ell}$) (i) solves the ROCP and the function UpdateStates($x_i, \bar{x}_i$) updates the states of agent i and its neighbors after every sampling time. If the OptSolve function does not return a solution, then there does not exist an admissible control input that can drive agent i to the desired neighboring region. After utilizing Algorithm 1, the WTS of each agent is defined as follows:

Definition 16. The motion of each agent $i \in \mathcal{V}$ in the workspace is modeled by the WTS $\mathcal{T}_i = (S_i,\ S_i^{\text{init}},\ Act_i,\ \longrightarrow_i,\ d_i,\ \Sigma_i,\ L_i)$ where: $S_i = \{D_\ell\}_{\ell \in \mathbb{I}}$ is the set of states of each agent; $S_i^{\text{init}} = P(i, 0) \subseteq S_i$ is a set of initial states defined by the agents' initial positions $x_i(t_0) \in P(i, 0)$ in the workspace; Act_i is the set of actions containing the union of all the admissible control inputs $u_i \in \mathcal{U}_i$ that are a feasible solution to the ROCP and can drive agent i between neighboring regions; $\longrightarrow_i \subseteq S_i \times Act_i \times S_i$ is the transition relation. We say that $(P(i, k), u_i, \widetilde{P}(i, k, \widetilde{\ell})) \in \longrightarrow_i,\ k \in \mathbb{N}, \widetilde{\ell} \in \mathbb{L}$ if there exist an admissible controller $u_i \in Act_i$ which at step k drives the agent i from the region $P(i, k)$ to the desired region $\widetilde{P}(i, k, \widetilde{\ell})$. Algorithm 1 gives the steps how the transition relation can be constructed. $d_i : \longrightarrow_i \rightarrow \mathbb{R}_{\geq 0}$, is a map that assigns a positive weight (duration) to each transition. The duration of each transition is exactly equal to T; Σ_i, is the set of atomic propositions; $L_i : S_i \rightarrow 2^{\Sigma_i}$, is the labeling function.

The individual WTSs of the agents will allow us to work directly in the discrete level and design sequences of controllers that solve Problem 1. Every WTS $\mathcal{T}_i, i \in \mathcal{V}$ generates timed runs and timed words of the form $r_i^t = (r_i(0), \tau_i(0))\ (r_i(1), \tau_i(1)) \ldots$, $w_i^t = (w_i(0), \tau_i(0))\ (w_i(1), \tau_i(1)) \ldots$, respectively, over the set 2^{Σ_i} with $w_i(\mu) = L_i(r_i(\mu)), \tau_i(\mu) = \mu T, \forall\ \mu \geq 0$. The transition relation $\longrightarrow_i$ along with the output of the Algorithm 1, i.e, Transit($\cdot$), allows each agent to have all the necessary information in order to be able to make a decentralized plan in the discrete level that is presented hereafter. The relation between the timed words that are generated by the WTSs $\mathcal{T}_i, i \in \mathcal{V}$ with the timed service words produced by the trajectories $x_i(t), i \in \mathcal{V}, t \geq 0$ is provided through the following remark:

Remark 11. By construction, each timed word produced by the WTS $\mathcal{T}_i$ is a *relaxed timed word* associated with the trajectory $x_i(t)$ of the system (2). Hence, if we find a timed word of $\mathcal{T}_i$ satisfying a formula φ_i given in MITL, we also find for each agent i a desired timed word of the original system, and hence trajectories $x_i(t)$ that are a solution to the Problem 1. Therefore, the produced timed words of $\mathcal{T}_i$ are compliant with the relaxed timed words of the trajectories $x_i(t)$.

B. Controller Synthesis

The proposed controller synthesis procedure is described with the following steps:

1) N TBAs $\mathcal{A}_i,\ i \in \mathcal{V}$ that accept all the timed runs satisfying the corresponding specification formulas $\varphi_i, i \in \mathcal{V}$ are constructed.
2) A Büchi WTS $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i$ (see Def. 17 below) is constructed for every $i \in \mathcal{V}$. The accepting runs of $\widetilde{\mathcal{T}}_i$ are the individual runs of $\mathcal{T}_i$ that satisfy the corresponding MITL formula $\varphi_i,\ i \in \mathcal{V}$.
3) The abstraction procedure allows to find an explicit feedback law for each transition in $\mathcal{T}_i$. Therefore, an accepting run $\widetilde{r}_i^t$ in $\mathcal{T}_i$ that takes the form of a sequence of transitions is realized in the system in (2) via the corresponding sequence of feedback laws.

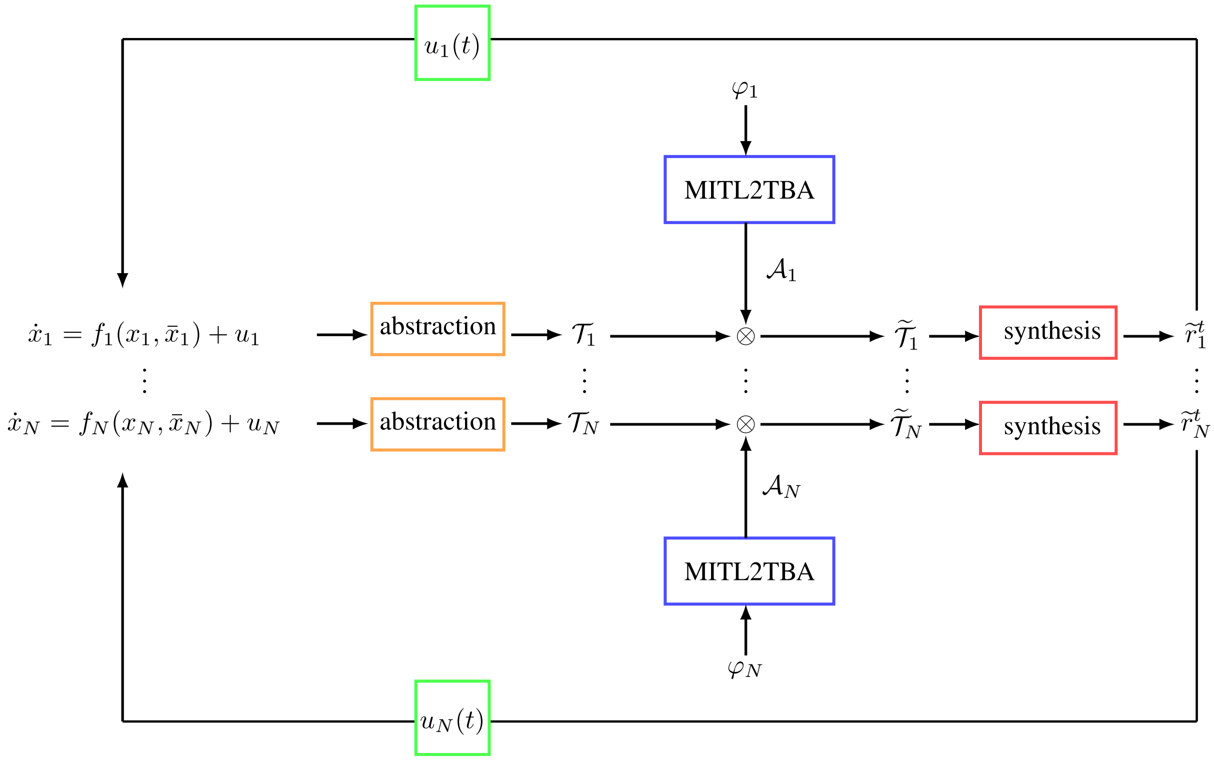

Fig. 6: A graphic illustration of the proposed framework.

Definition 17. Given a WTS $\mathcal{T}_i = (S_i, S_i^{\text{init}}, Act_i, \longrightarrow_i, d_i, \Sigma_i, L_i)$, and a TBA $\mathcal{A}_i = (Q_i, Q_i^{\text{init}}, C_i, Inv_i, E_i, F_i, \Sigma_i, \mathcal{L}_i)$ with $|C_i|$ clocks and let $C_i^{\max}$ be the largest constant appearing in $\mathcal{A}_i$. Then, we define their *Büchi WTS* $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i = (\widetilde{S}_i, \widetilde{S}_i^{\text{init}}, \widetilde{Act}_i, \rightsquigarrow_i, \widetilde{d}_i, \widetilde{F}_i, \Sigma_i, \widetilde{L}_i)$ as follows:

- $\widetilde{S}_i \subseteq \{(s_i, q_i) \in S_i \times Q_i : L_i(s_i) = \mathcal{L}_i(q_i)\} \times \mathbb{T}_\infty^{|C_i|}$.
- $\widetilde{S}_i^{\text{init}} = S_i^{\text{init}} \times Q_i^{\text{init}} \times \{0\}^{|C_i|}$.
- $\widetilde{Act}_i = Act_i$.
- $(\widetilde{q}, act_i, \widetilde{q}') \in \rightsquigarrow_i$ iff
  - $\widetilde{q} = (s, q, \nu_1, \dots, \nu_{|C_i|}) \in \widetilde{S}_i$,
    $\widetilde{q}' = (s', q', \nu_1', \dots, \nu_{|C_i|}') \in \widetilde{S}_i$,
  - $act_i \in Act_i$,
  - $(s, act_i, s') \in \longrightarrow_i$, and
  - there exists γ, R, such that $(q, \gamma, R, q') \in E_i$, $\nu_1, \dots, \nu_{|C_i|} \models \gamma$, $\nu_1', \dots, \nu_{|C_i|}' \models Inv_i(q')$, and for all $i \in \{1, \dots, |C_i|\}$

$$\nu_i' = \begin{cases} 0, & \text{if } c_i \in R \\ \nu_i + d_i(s, s'), & \text{if } c_i \notin R \text{ and} \\ & \nu_i + d_i(s, s') \leq C_i^{max} \\ \infty, & \text{otherwise.} \end{cases}$$

  Then, $\widetilde{d}_i(\widetilde{q}, \widetilde{q}') = d_i(s, s')$.
- $\widetilde{F}_i = \{(s_i, q_i, \nu_1, \dots, \nu_{|C_i|}) \in Q_i : q_i \in F_i\}$.
- $\widetilde{L}_i(s_i, q_i, \nu_1, \dots, \nu_{|C_i|}) = L_i(s_i)$.

Each Büchi WTS $\widetilde{\mathcal{T}}_i, i \in \mathcal{V}$ is in fact a WTS with a Büchi acceptance condition $\widetilde{F}_i$. A timed run of $\widetilde{\mathcal{T}}_i$ can be written as $\widetilde{r}_i^t = (q_i(0), \tau_i(0))(q_i(1), \tau_i(1)) \dots$ using the terminology of Def. 10. It is *accepting* if $q_i(\mu) \in \widetilde{F}_i$ for infinitely many $j \geq 0$. An accepting timed run of $\widetilde{\mathcal{T}}_i$ projects onto a timed run of $\mathcal{T}_i$ that satisfies the local specification formula φ_i by construction. Formally, the following lemma, whose proof follows directly from the construction and and the principles of automata-based LTL model checking (see, e.g., [53]), holds:

Lemma 6. *Consider an accepting timed run $\widetilde{r}_i^t = (q_i(0), \tau_i(0))(q_i(1), \tau_i(1)) \dots$ of the Büchi WTS $\widetilde{T}_i$ defined above, where $q_i(\mu) = (r_i(\mu), s_i(\mu), \nu_{i,1}, \dots, \nu_{i,|C_i|})$ denotes a state of $\widetilde{\mathcal{T}}_i$, for all $\mu \geq 0$. The timed run $\widetilde{r}_i^t$ projects onto the timed run $r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1)) \dots$ of the WTS $\mathcal{T}_i$ that produces the timed word $w(r_i^t) = (L_i(r_i(0)), \tau_i(0))(L_i(r_i(1)), \tau_i(1)) \dots$ accepted by the TBA $\mathcal{A}_i$ via its run $\chi_i = s_i(0)s_i(1) \dots$. Vice versa, if there exists a timed run $r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1)) \dots$ of the WTS T_i that produces a timed word $w(r_i^t) = (L_i(r_i(0)), \tau_i(0))(L_i(r_i(1)), \tau_i(1)) \dots$ accepted by the TBA A_i via its run $\chi_i = s_i(0)s_i(1) \dots$ then there exist the accepting timed run $\widetilde{r}_i^t = (q_i(0), \tau_i(0))(q_i(1), \tau_i(1)) \dots$ of $\widetilde{T}_i$, where $q_i(z)$ denotes $(r_i(z), s_i(z), \nu_{i,1}, \dots, \nu_{i,|C_i|})$ in $\widetilde{T}_i$.*

The proposed framework is depicted in Fig. 6. The dynamics (2) of each agent i is abstracted into a WTS $\mathcal{T}_i$ (orange rectangles). Then the product between each WTS $\mathcal{T}_i$ and the TBA $\mathcal{A}_i$ is computed according to Def. 17. The TBA $\mathcal{A}_i$ accepts all the words that satisfy the formula φ_i (blue rectangles). For every Büchi WTS $\widetilde{\mathcal{T}}_i$ the controller synthesis procedure that was described in this Section (red rectangles) is performed and a sequence of accepted runs $\{\widetilde{r}_1^t, \dots, \widetilde{r}_N^t\}$ is designed. Every accepted run $\widetilde{r}_i^t$ maps into a

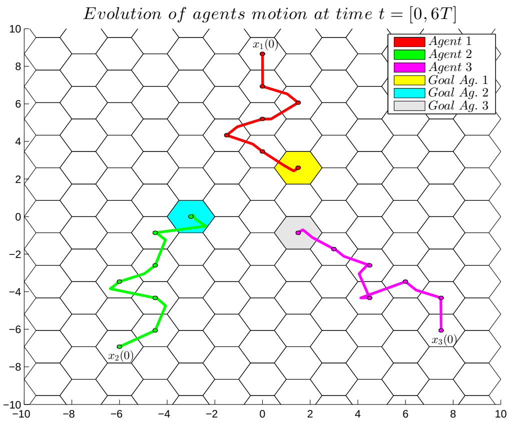

Fig. 7: Evolution of the agent's trajectories up to time $6T$ in the workspace W. Each point-to-point transition has time duration $T = 3$. The depicted timed runs with $\mathrm{red}, \mathrm{green}$ and $\mathrm{magenta}$, of agents $1, 2$ and 3, satisfy the formulas φ_1, φ_2 and φ_3, respectively, while the agents remain connected.

decentralized controller $u_i(t)$ which is a solution to Problem 1.

Proposition 1. The solution that we obtain from Steps 1-5, if one found, gives a sequence of controllers $u_1, \ldots, u_N$ that guarantees the satisfaction of the formulas formulas $\varphi_1, \ldots, \varphi_N$ of the agents $1, \ldots, N$ respectively, governed by dynamics as in (2). Thus, we solved Problem 1.

C. Complexity

In the proposed abstraction technique 6^N MPC optimization problems are solved for every time interval $t \in [t_k, t_k + T]$. Assume that the desired horizon for the system to run is M steps i.e. the timed sequence $\mathcal{S}$ is written as: $\mathcal{S} = \{t_0, t_1 = t_0 + T, \ldots, t_M = t_0 + MT\}$. Then the complexity of the abstraction is $M6^N$. As for the controller synthesis framework now we have the following. Denote by $|\varphi|$ the length of an MITL formula φ. A TBA $\mathcal{A}_i, i \in \mathcal{V}$ can be constructed in space and time $2^{\mathbb{O}(|\varphi_i|)}, i \in \mathcal{V}$ ($\mathbb{O}$ stands for the "big O" from complexity theory). Let $\varphi_{\max} = \max\{|\varphi_i|\}, i \in \mathcal{V}$ be the MITL formula with the longest length. Then, the complexity of Step 1 is $2^{\mathbb{O}(|\varphi_{\max})|}$. The model checking of Step 2 costs $\mathbb{O}(|\mathcal{T}_i|2^{|\varphi_i|}), i \in \mathcal{V}$ where $|\mathcal{T}_i|$ is the length of the WTS $\mathcal{T}_i$ i.e., the number of its states. Thus, $\mathbb{O}(|\mathcal{T}_i|2^{|\varphi_i|}) = \mathbb{O}(|S_i|2^{|\varphi_i|}) = \mathbb{O}(|\mathbb{I}|2^{|\varphi_i|})$, where $|\mathbb{I}|$ is the number of cells of the cell decomposition D. The worst case of Step 2 costs $\mathbb{O}(|\mathbb{I}|2^{|\varphi_{\max}|})$ due to the fact that all WTSs $\mathcal{T}_i, i \in \mathcal{I}$ have the same number of states. Therefore, the complexity of the total framework is $\mathbb{O}(M|\mathbb{I}|6^N 2^{|\varphi_{\max}|})$.

V. Simulation Results

For a simulation example, a system of three agents with $x_i \in \mathbb{R}^2,\ i \in \mathcal{V} = \{1, 2, 3\}$, $\mathcal{N}_1 = \{2, 3\}$ $\mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{1, 1\}$ is considered. The workspace $W = [-10, 10] \times [-10, 10] \subseteq \mathbb{R}^2$ is decomposed into hexagonal regions with $R = 1, r_h = \frac{\sqrt{3}}{2}$, which are depicted in Fig. 7. The initial agents' positions are set to $x_1(0) = (0, 10r_h), x_2(0) = (-6, -8r_h)$ and $x_3(0) = (7.5, -7r_h)$. The sensing radius is $\underline{r} = 18$. The dynamics are set to: $\dot{x}_1 = -2x_1 + x_2 + x_3 - \sin^2(x_1 - x_2) + u_1$, $\dot{x}_2 = -2x_2 + x_1 + x_3 - \sin^2(x_2 - x_1) + u_2$ and $\dot{x}_3 = -2x_3 + x_1 + x_2 + u_3$. The time step is $T = 3$. The specification formulas are set to $\varphi_1 = \Diamond_{[15,27]}\{\mathrm{red}\}, \varphi_2 = \Diamond_{[7.5,22]}\{\mathrm{green}\}, \varphi_3 = \Diamond_{[0,19]}\{\mathrm{grey}\}$ respectively. We set: $Q_i, P_i, R_i = I_2, \forall i \in \mathcal{V}$. Fig. 7 shows a sequence of transitions for agents $1, 2$ and 3 which form the accepting timed words $\widetilde{r}_1^t$, $\widetilde{r}_2^t$ and $\widetilde{r}_3^t$, respectively. Every timed word maps to a sequence of admissible control inputs for each agent, which is the outcome of solving the ROCPs. The agents remain connected for all $t \in [0, 6T]$. The simulations were carried out in MATLAB Environment by using the NMPC toolbox [46], on a desktop with 8 cores, 3.60GHz CPU and 16GB of RAM.

VI. Conclusions and Future Work

A systematic method of both decentralized abstractions and controller synthesis of a general class of coupled multi-agent systems has been proposed in which timed temporal specifications are imposed to the system. The solution involves a repetitive solving of an ROCP for every agent and for every desired region in order to build decentralized Transition Systems that are then used in the derivation of the controllers that satisfy the timed temporal formulas. Future work includes further computational improvement of the proposed decentralized abstraction method.

Appendix I
Proof of Property 1

Proof. By integrating (2) in the time interval $s \in [t_k, t_k + T]$ and taking the norms, we get:

$$\begin{aligned}
\|e_i(t)\| &= \left\| e_i(t_k) + \int_{t_k}^{t} \left[g_i(x_i(s), \bar{x}_i(s), u_i(s)) \right] ds \right\| \\
&\le \|e_i(t_k)\| + \left\| \int_{t_k}^{t} \left[f_i(e_i(s), \bar{x}_i(s)) + u_i(s) \right] ds \right\| \\
&\le \|e_i(t_k)\| + \int_{t_k}^{t} \|f_i(e_i(s), \bar{x}_i(s) + u_i(s)\| \, ds \\
&\le \|e_i(t_k)\| + \int_{t_k}^{t} \Big\{ \|f_i(e_i(s), \bar{x}_i(s)\| + \|u_i(s)\| \Big\} ds \\
&\le \|e_i(t_k)\| + \int_{t_k}^{t} (M + u_{\max}) ds \\
&= \|e_i(t_k)\| + (t - t_k)(M + u_{\max}),
\end{aligned}$$

which concludes the proof. ☐

Appendix II
Proof of Lemma 1

Proof. For every $e_1, e_2 \in E_i, u \in \mathcal{U}_i, i \in \mathcal{V}$, the following holds:

$$\begin{aligned}
&|F_i(e_1, u) - F_i(e_2, u)| = \\
&\qquad \|e_1^\top Q_i e_1 + u^\top R_i u - e_2^\top Q_i e_2 - u^\top R_i u|
\end{aligned}$$

$$
\begin{aligned}
&= |e_1^\top Q_i e_1 - e_2^\top Q_i e_2| \\
&= |e_1^\top Q_i e_1 + e_1^\top Q_i e_2 - e_1^\top Q_i e_2 - e_2^\top Q_i e_2| \\
&= |e_1^\top Q_i (e_1 - e_2) - e_2^\top Q_i (e_1 - e_2)| \\
&\le |e_1^\top Q_i (e_1 - e_2)| + |e_2^\top Q_i (e_1 - e_2)|. \qquad (24)
\end{aligned}
$$

By employing the property that:

$$
|x^\top Ay| \le \sigma_{\max}(A)\|x\|\|y\|, \forall\ x, y \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}, \quad (25)
$$

(24) is written as:

$$
\begin{aligned}
|F_i(e_1, u) - F_i(e_2, u)| &\le \sigma_{\max}(Q_i)\|e_1\|\|e_1 - e_2\| \\
&\quad + \sigma_{\max}(Q_i)\|e_2\|\|e_1 - e_2\| \\
&= \sigma_{\max}(Q_i)(\|e_1\| + \|e_2\|)\|e_1 - e_2\| \\
&= \sigma_{\max}(Q_i)\left[\sup_{e_1,e_2\in E_i}\{\|e_1\| + \|e_2\|\}\right]\|e_1 - e_2\| \\
&= 2\sigma_{\max}(Q_i)\left[\sup_{e_i\in E_i}\{\|e\|\}\right]\|e_1 - e_2\| \\
&= [2\bar{\varepsilon}_i\sigma_{\max}(Q_i)]\,\|e_1 - e_2\|.
\end{aligned}
$$

which completes the proof. □

Appendix III
Proof of Lemma 2

Proof. Let us denote by:

$$
\begin{aligned}
u_i(\cdot) &\triangleq u_i(s; e(t_{k_z})), \\
e_i(s) &\triangleq e_i(s; u_i(\cdot), e_i(t_{k_z})).
\end{aligned}
$$

the control input and real trajectory of the system (2) for $s \in [t_{k_z}, t_{k_z} + T_z]$. Also, denote for sake of simplicity:

$$
\hat{e}_i(s) \triangleq \hat{e}_i(s; u_i(\cdot), e_i(t_{k_z})).
$$

the corresponding estimated trajectory. By integrating (2), (13b) for the time interval $[t_{k_z}, t_{k_z}+s]$ we have the following:

$$
\begin{aligned}
e_i(s) &= e_i(t_{k_z}) + \int_{t_{k_z}}^{s} \left[g_i(e_i(s'), \bar{x}_i(s'), u_i(\cdot))\right] ds', \\
\hat{e}_i(s) &= e_i(t_{k_z}) + \int_{t_{k_z}}^{s} \left[g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot))\right] ds',
\end{aligned}
$$

respectively. Then, we have that:

$$
\begin{aligned}
&\|e_i(s) - \hat{e}_i(s)\| \\
&= \left\| \int_{t_{k_z}}^{s} \left[g(e_i(s'), \bar{x}_i(s'), u_i(\cdot))\right] ds' - \int_{t_{k_z}}^{s} \left[g(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot))\right] ds' \right\| \\
&= \left\| \int_{t_{k_z}}^{s} \left[f(e_i(s'), \bar{x}_i(s')) + u_i(s') - f(\hat{e}_i(s'), \hat{\bar{x}}_i(s')) - u_i(s')\right] ds' \right\| \\
&= \left\| \int_{t_{k_z}}^{s} \left[f(e_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \hat{\bar{x}}_i(s'))\right] ds' \right\| \\
&\le \int_{t_{k_z}}^{s} \left\| f(e_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \hat{\bar{x}}_i(s')) \right\| ds' \\
&= \int_{t_{k_z}}^{s} \Big\| f(e_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \bar{x}_i(s')) \\
&\qquad + f(\hat{e}_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \hat{\bar{x}}_i(s')) \Big\| ds' \\
&\le \int_{t_{k_z}}^{s} \left\| f(e_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \bar{x}_i(s')) \right\| ds' \\
&\qquad + \int_{t_{k_z}}^{s} \left\| f(\hat{e}_i(s'), \bar{x}_i(s')) - f(\hat{e}_i(s'), \hat{\bar{x}}_i(s')) \right\| ds'.
\end{aligned}
$$

By using the bounds of (4a)-(4b) we obtain:

$$
\begin{aligned}
&\|e_i(s) - \hat{e}_i(s)\| \\
&\le \int_{t_{k_z}}^{s} L_i \left\| e_i(s') - \hat{e}_i(s') \right\| ds' \\
&\qquad + \int_{t_{k_z}}^{s} \bar{L}_i \left\| \bar{x}_i(s') - \hat{\bar{x}}_i(s') \right\| ds'. \qquad (26)
\end{aligned}
$$

The following property holds:

$$
\|\bar{x}_i - \hat{\bar{x}}_i\| \le \sum_{j\in\mathcal{N}_i} \|x_j - \hat{x}_j\|, \forall i \in \mathcal{V}, j \in \mathcal{N}_i.
$$

Then, by combining the last inequality with (18) from Property 2, we have that:

$$
\|\bar{x}_i - \hat{\bar{x}}_i\| \le \sum_{j\in\mathcal{N}_i} 2\sqrt{3}R = 2\sqrt{3}RN_i, \forall i \in \mathcal{V}, j \in \mathcal{N}_i.
$$

By combining the last result with (26) we get:

$$
\begin{aligned}
&\|e_i(s) - \hat{e}_i(s)\| \le \\
&\int_{t_{k_z}}^{s} L_i \left\| e_i(s') - \hat{e}_i(s') \right\| ds' + \int_{t_{k_z}}^{s} \bar{L}_i 2\sqrt{3}RN_i ds' \\
&= \int_{t_{k_z}}^{s} L_i \left\| e_i(s') - \hat{e}_i(s') \right\| ds' + 2\sqrt{3}R\bar{L}_i N_i (s - t_{k_z}). \qquad (27)
\end{aligned}
$$

By employing the Gronwall-Bellman inequality from [26], (27) becomes:

$$
\begin{aligned}
&\|e_i(s) - \hat{e}_i(s)\| \\
&\le 2\sqrt{3}RL_i\bar{L}_iN_i \int_{t_{k_z}}^{s} (s' - t_{k_z}) \exp\left[\int_{s'}^{s} L_i ds''\right] ds' \\
&\qquad + 2\sqrt{3}R\bar{L}_iN_i(s - t_{k_z}) \\
&= 2\sqrt{3}RL_i\bar{L}_iN_i \int_{t_{k_z}}^{s} (s' - t_{k_z}) \exp\left[L_i(-s' + s)\right] ds' \\
&\qquad + 2\sqrt{3}R\bar{L}_iN_i(s - t_{k_z}) \\
&= -2\sqrt{3}R\bar{L}_iN_i(s - t_{k_z}) + 2\sqrt{3}R\bar{L}_iN_i(s - t_{k_z}) \\
&\qquad + 2\sqrt{3}R\bar{L}_iN_i \int_{t_{k_z}}^{s} \exp\left[L_i(-s' + s)\right] ds' \\
&= 2\sqrt{3}R\bar{L}_iN_i \int_{t_{k_z}}^{s} \exp\left[L_i(-s' + s)\right] ds' \\
&= -\frac{2\sqrt{3}R\bar{L}_iN_i}{L_i}\left[1 - e^{L_i(s - t_{k_z})}\right] \\
&= \frac{2\sqrt{3}R\bar{L}_iN_i}{L_i}\left[e^{L_i(s - t_{k_z})} - 1\right]. \qquad (28)
\end{aligned}
$$

By employing (11) of Property 1 for the terms $e(s), \hat{e}(s)$ we have that:

$$\begin{aligned}&\|e_i(s)-\hat{e}_i(s)\| \le \|e_i(s)\| + \|\hat{e}_i(s)\| \\ &\le \|e_i(t_{k_z}; u_i(\cdot), e_i(t_{k_z}))\| + (s-t_{k_z})(M+u_{\max}) + \\ &\qquad \|\hat{e}_i(t_{k_z}; u_i(\cdot), e_i(t_{k_z}))\| + (s-t_{k_z})(M+u_{\max}) \\ &\le 2\|e_i(t_{k_z}; u_i(\cdot), e_i(t_{k_z}))\| + 2(s-t_{k_z})(M+u_{\max}) \\ &= 2\|e_i(t_{k_z})\| + 2(s-t_{k_z})(M+u_{\max}). \end{aligned} \tag{29}$$

By combining (28), (29) we get:

$$\begin{aligned}\|e_i(s)-\hat{e}_i(s)\| \le \min\Big\{ &\frac{2\sqrt{3}R\bar{L}_i N_i}{L_i}\left[e^{L_i(s-t_{k_z})}-1\right], \\ &2\|e_i(t_{k_z})\| + 2(s-t_{k_z})(M+u_{\max})\Big\}.\end{aligned}$$

which leads to the conclusion of the proof. □

Appendix IV
Proof of Property 3

Proof. Let $s \in [t_{k_z}, t_{k_z}+T_z]$. Let us also define:

$$z_i(s) \triangleq e_i(s) - \hat{e}_i(s; u_i(s; e(t_{k_z})), e_i(t_{k_z})).$$

Then, according to Lemma 2, for $s \in [t_{k_z}, t_{k_z}+T_z]$, we get:

$$\begin{aligned}\|z_i(s)\| &= \|e_i(s) - \hat{e}_i(s; u_i(s; e_i(t_{k_z})), e_i(t_{k_z}))\| \\ &\le \rho_i(s-t_{k_z}).\end{aligned}$$

Hence, $z_i \in B^i_{s-t_{k_z}}$, which implies that: $-z_i \in B^i_{s-t_{k_z}}$. The following implications hold:

$$\begin{aligned}&\hat{e}_i(s; u_i(s; e_i(t_{k_z})), e_i(t_{k_z})) \in E_i \sim B^i_{s-t_{k_z}} \\ &\qquad \Rightarrow e_i(s) - z_i \in E \sim B^i_{s-t_{k_z}} \\ &\qquad \Rightarrow e_i(s) + (-z_i) \in E \sim B^i_{s-t_{k_z}} \\ &\qquad\qquad \Rightarrow e_i(s) \in E_i, \forall\ s \in [t_{k_z}, t_{k_z}+T_z],\end{aligned}$$

which concludes the proof. □

Appendix V
Proof of Lemma 3

Proof. For every $e_i \in \Phi_i$ we have that:

$$\begin{aligned}V_i(e_i) \le \alpha_{1,i} &\Rightarrow e_i^\top P_i e_i \le \alpha_{1,i} \\ &\Rightarrow \lambda_{\min}(P_i)\|e_i\|^2 \le e_i^\top P_i e_i \le \alpha_{1,i} \\ &\Rightarrow \|e_i\| \le \sqrt{\frac{\alpha_{1,i}}{\lambda_{\min}(P_i)}}.\end{aligned} \tag{30}$$

For every $e_1, e_2 \in \Phi_i$, it also holds:

$$\begin{aligned}|V_i(e_1)-V_i(e_2)| &= |e_1^\top P_i e_1 - e_2^\top P_i e_2| \\ &= |e_1^\top P_i e_1 + e_1^\top P_i e_2 - e_1^\top P_i e_2 - e_2^\top P_i e_2| \\ &= |e_1^\top P_i(e_1-e_2) - e_2^\top P_i(e_1-e_2)| \\ &\le |e_1^\top P_i(e_1-e_2)| + |e_2^\top P_i(e_1-e_2)|,\end{aligned}$$

which by using (25) leads to:

$$\begin{aligned}|V_i(e_1)-V_i(e_2)| &\le \sigma_{\max}(P_i)\|e_1\|\|e_1-e_2\| \\ &\quad + \sigma_{\max}(P_i)\|e_2\|\|e_1-e_2\| \\ &= \sigma_{\max}(P_i)(\|e_1\|+\|e_2\|)\|e_1-e_2\|,\end{aligned}$$

which by employing (30), becomes:

$$\begin{aligned}|V_i(e_1)-V_i(e_2)| &\le \sigma_{\max}(P_i)\|e_1\|\|e_1-e_2\| \\ &\quad + \sigma_{\max}(P_i)\|e_2\|\|e_1-e_2\| \\ &= \sigma_{\max}(P_i)(\|e_1\|+\|e_2\|)\|e_1-e_2\| \\ &\le 2\sigma_{\max}(P_i)\sqrt{\frac{\alpha_1}{\lambda_{\min}(P_i)}}\|e_1-e_2\|,\end{aligned}$$

which completes the proof. □

Appendix VI
Proof of Lemma 4

Proof. For every $s \ge t_{k_{z+1}}, L_g > 0$ the following inequality holds:

$$\left[e^{L_g(t_{k_{z+1}}-t_{k_z})}-1\right] + \left[e^{L_g(s-t_{k_{z+1}})}-1\right] \le \left[e^{L_g(s-t_{k_z})}-1\right],$$

which implies that:

$$\begin{aligned}\widetilde{\rho}_i\left[e^{L_g(t_{k_{z+1}}-t_{k_z})}-1\right] + \widetilde{\rho}_i\left[e^{L_g(s-t_{k_{z+1}})}-1\right] \\ \le \widetilde{\rho}_i\left[e^{L_g(s-t_{k_z})}-1\right].\end{aligned} \tag{31}$$

It holds also that:

$$\begin{aligned}&t_{k_{z+1}} - t_{k_z} + s - t_{k_{z+1}} \le s - t_{k_z} \\ \Leftrightarrow\ &2\|e_i(t_{k_z})\| + 2(t_{k_{z+1}}-t_{k_z})(M+u_{\max}) \\ &\ + 2\|e_i(t_{k_z})\| + 2(s-t_{k_{z+1}})(M+u_{\max}) \le \\ &2\|e_i(t_{k_z})\| + 2(s-t_{k_z})(M+u_{\max}).\end{aligned} \tag{32}$$

By setting:

$$\begin{aligned}A_1 &= \widetilde{\rho}_i\left[e^{L_g(t_{k_{z+1}}-t_{k_z})}-1\right], \\ A_2 &= \widetilde{\rho}_i\left[e^{L_g(s-t_{k_{z+1}})}-1\right], \\ A_3 &= \widetilde{\rho}_i\left[e^{L_g(s-t_{k_z})}-1\right], \\ B_1 &= 2\|e_i(t_{k_z})\| + 2(t_{k_{z+1}}-t_{k_z})(M+u_{\max}), \\ B_2 &= 2\|e_i(t_{k_z})\| + 2(s-t_{k_{z+1}})(M+u_{\max}), \\ B_3 &= 2\|e_i(t_{k_z})\| + 2(s-t_{k_z})(M+u_{\max}),\end{aligned}$$

and taking account (31), (32) we get:

$$\begin{aligned}\rho_i(t_{k_{z+1}}&-t_{k_z}) + \rho_i(s-t_{k_{z+1}}) \\ &\le \min\{A_1, B_1\} + \min\{B_1, B_2\} \\ &\le \min\{A_1+A_2, B_1+B_2\} \\ &\le \min\{A_3, B_3\} \\ &= \rho_i(s-t_{k_z}),\end{aligned}$$

or:

$$\rho_i(t_{k_{z+1}}-t_{k_z}) + \rho_i(s-t_{k_{z+1}}) \le \rho_i(s-t_{k_z}). \tag{33}$$

Let us consider $\phi \in B^i_{s-t_{k_{z+1}}}$. Then, it holds $\|\phi\| \le \rho_i(s-t_{k_{z+1}})$. Let us denote $z = x - y + \phi$. It is clear that:

$$\begin{aligned}\|z\| &\le \|x-y+\phi\| \\ &\le \|x-y\| + \|\phi\| \\ &\le \rho_i(t_{k_{z+1}}-t_{k_z}) + \rho_i(s-t_{k_{z+1}}).\end{aligned} \tag{34}$$

By employing (33), (34) becomes:

$$\|z\| \leq \rho_i(s - t_{k_z}),$$

which implies that $z \in B^i_{s-t_{k_z}}$. We have that:

$$\begin{aligned} x + (-z) &= y + (-\phi), \\ x &\in E_{s-t_{k_z}} = E \sim B_{s-t_{k_z}}, \\ -z &\in B^i_{s-t_{k_z}}, \\ -\rho &\in B^i_{s-t_{k_{z+1}}}, \end{aligned}$$

which implies that $y \in E_{s-t_{k_{z+1}}} = E \sim B_{s-t_{k_{z+1}}}$. □

APPENDIX VII
PROOF OF LEMMA 5

Proof. Let $s \geq t_{k_z}$. The following equalities hold:

$$\begin{aligned}
&\|\hat{e}_i(s; u_i(\cdot), e_i(t_{k_{z+1}})) - \hat{e}_i(s; u_i(\cdot), e_i(t_{k_z}))\| \\
&= \Bigg\| \hat{e}_i(s; u_i(\cdot), e_i(t_{k_{z+1}})) + \int_{t_{k_{z+1}}}^{s} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot)) ds' \\
&\quad - \hat{e}_i(t_{k_z}; u_i(\cdot), e_i(t_{k_z})) - \int_{t_{k_z}}^{s} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot))) ds \Bigg\| \\
&= \Bigg\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \int_{t_{k_z}}^{s} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot)) ds' \\
&\qquad - \int_{s}^{t_{k_{z+1}}} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot)) ds' \Bigg\| \\
&= \Bigg\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \int_{t_{k_z}}^{t_{k_{z+1}}} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), u_i(\cdot)) ds' \Bigg\| \\
&= \Bigg\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \int_{t_{k_z}}^{t_{k_{z+1}}} \frac{d}{dt} \left[\hat{e}_i(s'; u_i(\cdot), e_i(t_{k_z}) \right] ds' \Bigg\| \\
&= \big\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \hat{e}(t_{k_{z+1}}; u(\cdot), e_i(t_{k_z})) \\
&\qquad + \hat{e}_i(t_{k_z}; u_i(\cdot), e_i(t_{k_z})) \big\| \\
&= \big\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \hat{e}(t_{k_{z+1}}; u(\cdot), e_i(t_{k_z})) + e_i(t_{k_z}) \big\| \\
&= \big\| e_i(t_{k_{z+1}}) - \hat{e}_i(t_{k_{z+1}}; u_i(\cdot), e_i(t_{k_z})) \big\|,
\end{aligned}$$

which, by employing Lemma 2 for $s = t_{k_{z+1}}$, becomes:

$$\begin{aligned} \|\hat{e}_i(s; u(\cdot), e_i(t_{k_{z+1}})) - \hat{e}_i(s; u_i(\cdot), e_i(t_{k_z}))\| &\leq \\ \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h), \end{aligned}$$

since $t_{k_{z+1}} - t_{k_z} = h$, which concludes the proof. □

APPENDIX VIII
FEASIBILITY AND CONVERGENCE

Proof. The proof consists of two parts: in the first part it is established that initial feasibility implies feasibility afterwards. Based on this result it is then shown that the error $e_i(t)$ converges to the terminal set $\mathcal{E}_i$.

Feasibility Analysis: Consider any sampling time instant for which a solution exists, say t_{k_z}. In between t_{k_z} and $t_{k_{z+1}}$, the optimal control input $\hat{u}^\star_i(s; e_i(t_{k_z})), s \in [t_{k_z}, t_{k_{z+1}})$ is implemented. The remaining part of the optimal control input $\hat{u}^\star_i(s; e_i(t_{k_z})), s \in [t_{k_{z+1}}, t_{k_z} + T_z]$, satisfies the state and input constraints $E_i, \mathcal{U}_i$, respectively. Furthermore, since the problem is feasible at time t_{k_z}, it holds that:

$$\hat{e}_i(s; \hat{u}^\star(s; e_i(t_{k_z})), e_i(t_{k_z})) \in E^i_{s-t_{k_z}}, \tag{35a}$$

$$\hat{e}_i(t_{k_z} + T; \hat{u}^\star_i(s; e_i(t_{k_z})), e_i(t_{k_z})) \in \mathcal{E}_i, \tag{35b}$$

for $s \in [t_{k_z}, t_{k_z} + T_z]$. By using Property 1, (35a) implies also that $e_i(s; \hat{u}^\star_i(s; e_i(t_{k_z})), e_i(t_{k_z})) \in E_i$. We know also from Assumption 5 that for all $e_i \in \mathcal{E}_i$, there exists at least one control input $u_{f,i}(\cdot)$ that renders the set $\mathcal{E}_i$ invariant over h. Picking any such input, a feasible control input $\bar{u}_i(\cdot; e_i(t_{k_{z+1}}))$, at time instant $t_{k_{z+1}}$, may be the following:

$$\bar{u}_i(s; e(t_{k_{z+1}})) = \begin{cases} \hat{u}^\star_i(s; e_i(t_{k_z})), & s \in [t_{k_{z+1}}, t_{k_z} + T_{z+1}], \\ u_{f,i}(t_{k_z} + T_{z+1}; \hat{u}^\star(\cdot), e(t_i))), & s \in [t_{k_z} + T_{z+1}, t_{k_z} + T_z]. \end{cases} \tag{36}$$

For the time intervals it holds that (see Fig. 4):

$$t_{k_z} + T_{z+1} = t_{k_z} + T_z - h = t_{k_z} + T - h.$$

For the feasibility of the ROCP, we have to prove the following three statements for every $s \in [t_{k_{z+1}}, t_{k_z} + T_z]$:

1) $\bar{u}_i(s; e(t_{k_{z+1}})) \in \mathcal{U}_i$.
2) $\hat{e}_i(t_{k_z} + T_z; \bar{u}(s; e(t_{k_{z+1}})), e(t_{k_{z+1}})) \in \mathcal{E}_i$.
3) $\hat{e}_i(s; \bar{u}_i(s; e(t_{k_{z+1}})), e(t_{k_{z+1}})) \in E^i_{s-t_{k_{z+1}}}$.

Statement 1: From the feasibility of $\hat{u}^\star_i(s, e(t_{k_z}))$ and the fact that $u_{f,i}(e_i(\cdot)) \in \mathcal{U}_i$, for all $e_i(\cdot) \in \Phi_i$, it follows that:

$$\bar{u}_i(s; e_i(t_{k_{z+1}})) \in \mathcal{U}_i, \forall\ s \in [t_{k_{z+1}}, t_{k_z} + T_z].$$

Statement 2: We need to prove in this step that for every $s \in [t_{k_{z+1}}, t_{k_z} + T_z]$ it holds that $\hat{e}_i(t_{k_z} + T_z; \bar{u}_i(s; e_i(t_{k_{z+1}}))), e_i(t_{k_{z+1}})) \in \mathcal{E}_i$. Since $V_i(\cdot)$ is Lipschitz continuous, we get:

$$\begin{aligned} &V_i(\hat{e}_i(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e_i(t_{k_{z+1}}))) - \\ &\qquad V_i(\hat{e}_i(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e_i(t_{k_z}))) \leq \\ &L_{V_i} \|\hat{e}_i(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) \\ &\qquad - \hat{e}(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e(t_{k_z}))\|. \end{aligned} \tag{37}$$

for the same control input $\bar{u}_i(\cdot) = u^\star_i(s; e_i(t_{k_z}))$. By employing Lemma 5 for $\alpha = t_{k_z} + T_{z+1}$ and $u(\cdot) = \bar{u}_i(\cdot) = u^\star_i(s; e_i(t_{k_z}))$, we have that:

$$\begin{aligned} &\|\hat{e}_i(t_{k_z} + T_z; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) \\ &\quad - \hat{e}_i(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e(t_{k_z}))\| \leq \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h). \end{aligned} \tag{38}$$

Note also that for the function $\rho_i(\cdot)$ the following implication holds:

$$h \leq T_z \Rightarrow \rho_i(h) \leq \rho_i(T_z).$$

By employing the latter result, (38) becomes:

$$\begin{aligned} &\|\hat{e}_i(t_{k_z} + T_z; \bar{u}_i(\cdot), e_i(t_{i+1})) \\ &\quad - \hat{e}_i(t_{k_z} + T_{z+1}; \bar{u}_i(\cdot), e(t_{k_z}))\| \leq \rho_i(h) \leq \rho_i(T_z). \end{aligned} \tag{39}$$

By combining (39) and (37) we get:

$$V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_{z+1}})))- \\ V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_z}))) \leq L_{V_i}\rho_i(T_z),$$

or equivalently:

$$V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_{z+1}}))) \leq \\ V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_z}))) + L_{V_i}\rho_i(T_z). \quad (40)$$

By using (35b), we have that $\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_z})) \in \mathcal{E}_i$. Then, (40) gives:

$$V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_{z+1}}))) \leq \alpha_{2,i} + L_{V_i}\rho_i(T_z) \quad (41)$$

From (**??**) of the Theorem 1, we get equivalently:

$$\rho_i(T_z) \leq \frac{\alpha_{1,1}-\alpha_{2,i}}{L_{V_i}} \\ \Leftrightarrow \alpha_{2,i} + L_{V_i}\rho_i(T_z) \leq \alpha_{1,i}. \quad (42)$$

By combining (41) and (42), we get:

$$V_i(\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_{z+1}}))) \leq \alpha_{1,i},$$

which, from Assumption 5, implies that:

$$\hat{e}_i(t_{k_z}+T_{z+1};\bar{u}_i(\cdot),e_i(t_{k_{z+1}})) \in \Phi_i. \quad (43)$$

But since $\bar{u}_i(\cdot)$ is chosen to be local admissible controller from Assumption 5, according to our choice of terminal set $\mathcal{E}_i$, (43) leads to:

$$\hat{e}_i(t_{k_z}+T_z;\bar{u}_i(\cdot),e_i(t_{k_{z+1}})) \in \mathcal{E}_i.$$

Thus, statement 2 holds.
Statement 3: By employing Lemma 5 for:

$$x = \hat{e}_i(s;\bar{u}_i(s;e(t_{k_z})),e(t_{k_z})) \in E^i_{s-t_i}, \\ y = \hat{e}_i(s;\bar{u}_i(s;e(t_{k_{z+1}})),e(t_{k_{z+1}})),$$

we get that:

$$\|y-x\| = \|\hat{e}(s;\bar{u}(s;e(t_{i+1})),e(t_{i+1})) \\ - \hat{e}(s;\bar{u}(s;e(t_i)),e(t_i)) \in E_{s-t_i}\| \leq \rho_i(h).$$

Furthermore, by employing Lemma 4 for $s \in [t_{k_{z+1}}, t_{k_z}+T_z]$ and the same x, y as previously we get that $y = \hat{e}_i(s;\bar{u}(s;e_i(t_{k_{z+1}})),e(t_{k_{z+1}})) \in E^i_{s-t_{k_{z+1}}}$, which according to Property 1, implies that $e_i(s;\bar{u}_i(s;e_i(t_{k_{z+1}})),e_i(t_{k_{z+1}})) \in E_i$. Thus, Statement 3 holds. Hence, the feasibility at time t_{k_z} implies feasibility at time $t_{k_{z+1}}$. Therefore, if the ROCP (13a) - (13d) is feasible at time t_{k_z}, i.e., it remains feasible for every $t \in [t_k, t_k+T]$.

Convergence Analysis: The second part involves proving convergence of the state e_i to the terminal set $\mathcal{E}_i$. In order to prove this, it must be shown that a proper value function is decreasing along the solution trajectories starting at a sampling time t_i. Consider the optimal value function $J_i^\star(e_i(t_{k_z}))$, as is given in (17), to be a Lyapunov-like function. Consider also the cost of the feasible control input, indicated by:

$$\bar{J}_i(e_i(t_{k_{z+1}})) \triangleq \bar{J}_i(e_i(t_{k_{z+1}}),\bar{u}_i(\cdot;e_i(t_{k_{z+1}}))), \quad (44)$$

where $t_{k_{z+1}} = t_{k_z}+h$. Define:

$$\bar{u}_1(s) \triangleq \bar{u}_i(s;e_i(t_{k_{z+1}})), \quad (45a)$$
$$\bar{e}_1(s) \triangleq \bar{e}_i(s;u_1(s),e_i(t_{k_{z+1}})), s \in [t_{k_{z+1}}, t_{k_z}+T], \quad (45b)$$

where $\bar{e}_1(s)$ stands for the predicted state e_i at time s, based on the measurement of the state e_i at time $t_{k_{z+1}}$, while using the feasible control input $\bar{u}_i(s;e(t_{k_{z+1}}))$ from (36). Let us also define the following terms:

$$\hat{u}_2(s) \triangleq \hat{u}_i^\star(s;e_i(t_{k_z})), \quad (46) \\ \hat{e}_2(s) \triangleq \hat{e}_i(s;\hat{u}_2(s),e_i(t_{k_z})), s \in [t_{k_z}, t_{k_z}+T-h].$$

where $\hat{e}_1(s)$ stands for the predicted state e_i at time s, based on the measurement of the state e_i at time t_{k_z}, while using the control input $\hat{u}_i(s;e(t_{k_z})), s \in [t_{k_z}, t_{k_z}+T-h]$ from (36). By employing (13a), (17) and (44), the difference between the optimal and feasible cost is given by:

$$\begin{aligned}
&\bar{J}(e_i(t_{k_{z+1}})) - J^\star(e_i(t_{k_z})) = \\
&V_i(\bar{e}_1(t_{k_z}+T) + \int_{t_{k_{z+1}}}^{t_{k_z}+T}\Big[F_i(\bar{e}_1(s),\bar{u}_1(s))\Big]ds \\
&- V_i(\hat{e}_2(t_{k_z}+T-h) - \int_{t_{k_z}}^{t_{k_z}+T-h}\Big[F_i(\hat{e}_2(s),\hat{u}_2(s))\Big]ds \\
&= V_i(\bar{e}_1(t_{k_z}+T)) + \int_{t_{k_{z+1}}}^{t_{k_{z+1}}+T-h}\Big[F_i(\bar{e}_1(s),\bar{u}_1(s))\Big]ds \\
&+ \int_{t_{k_z}+T-h}^{t_{k_z}+T}\Big[F_i(\bar{e}_1(s),\bar{u}_1(s))\Big]ds - V_i(\hat{e}_2(t_{k_z}+T-h)) \\
&- \int_{t_{k_z}}^{t_{k_{z+1}}}\Big[F_i(\hat{e}_2(s),\hat{u}_2(s))\Big]ds \\
&- \int_{t_{k_{z+1}}}^{t_{k_z}+T-h}\Big[F_i(\hat{e}_2(s),\hat{u}_2(s))\Big]ds. \quad (47)
\end{aligned}$$

Note that, from (36), the following holds:

$$\bar{u}_i(s;e_i(t_{k_{z+1}})) = \hat{u}_i^\star(s;e_i(t_{k_z})), \forall\ s \in [t_{k_{z+1}}, t_{k_z}+T-h]. \quad (48)$$

By combining (45a), (46) and (48), we have that:

$$\bar{u}_1(s) = \hat{u}_2(s) = \bar{u}_i(s;e_i(t_{k_{z+1}})) = \hat{u}_i^\star(s;e_i(t_{k_z})), \\ \forall\ s \in [t_{k_{z+1}}, t_{k_z}+T-h], \quad (49)$$

By applying the last result and the fact that $F_i(e,u)$ is Lipschitz, the following holds:

$$\begin{aligned}
&\int_{t_{k_{z+1}}}^{t_{k_z}+T-h}\Big[F_i(\bar{e}_1(s),\bar{u}_1(s))\Big]ds \\
&\qquad - \int_{t_{k_{z+1}}}^{t_{k_z}+T-h}\Big[F_i(\hat{e}_2(s),\hat{u}_2(s))\Big]ds \\
&= \int_{t_{k_{z+1}}}^{t_{k_z}+T-h}\Big[F_i(\bar{e}_1(s),\bar{u}_1(s)) - F_i(\hat{e}_2(s),\hat{u}_2(s))\Big]ds
\end{aligned}$$

$$
\begin{aligned}
&= \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \Big[F_i(\bar{e}_1(s), \bar{u}_i(s; e_i(t_{k_{z+1}}))) \\
&\qquad\qquad - F_i(\hat{e}_2(s), \bar{u}_i(s; e_i(t_{k_{z+1}}))) \Big] ds \\
&\leq \left\| \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \Big[F_i(\bar{e}_1(s), \bar{u}_i(\cdot)) - F_i(\hat{e}_2(s), \bar{u}_i(\cdot)) \Big] ds \right\| \\
&\leq \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \Big\| F_i(\bar{e}_1(s), \bar{u}_i(\cdot)) - F_i(\hat{e}_2(s), \bar{u}_i(\cdot)) \Big\| ds \\
&\leq L_{F_i} \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \| \bar{e}_1(s) - \hat{e}_2(s) \| \, ds. \qquad (50)
\end{aligned}
$$

By employing the fact that $\forall s \in [t_{k_{z+1}}, t_{k_z} + T - h]$ the following holds:

$$
\bar{e}_i(s; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) = \hat{e}_i(s; \bar{u}_i(\cdot), e_i(t_{k_z})), \qquad (51)
$$

the term $\|\bar{e}_1(s) - \hat{e}_2(s)\|$ can be written as:

$$
\begin{aligned}
&\|\bar{e}_1(s) - \hat{e}_2(s)\| \\
&= \|\bar{e}_i(s; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) - \hat{e}_i(s; \hat{u}_i(\cdot), e_i(t_{k_z}))\| \\
&= \Bigg\| \bar{e}_i(t_{k_{z+1}}; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) \\
&+ \int_{t_{k_{z+1}}}^{s} g_i(\bar{e}_i(s'), \hat{\bar{x}}_i(s'), \bar{u}_i(\cdot)) ds' \\
&- \hat{e}_i(t_{k_z}; \hat{u}_i(\cdot), e_i(t_{k_z})) - \int_{t_{k_z}}^{t_{k_{z+1}}} g_i(\hat{e}_i(s), \hat{\bar{x}}_i(s'), \hat{u}_i(\cdot)) ds \\
&\qquad - \int_{t_{k_{z+1}}}^{s} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), \bar{u}_i(\cdot)) ds' \Bigg\| \\
&\leq \left\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \int_{t_{k_z}}^{t_{k_{z+1}}} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(s'), \hat{u}_i(\cdot)) ds' \right\| \\
&+ \Bigg\| \int_{t_{k_{z+1}}}^{s} g_i(\bar{e}_i(s'), \hat{\bar{x}}_i(\cdot), \bar{u}_i(\cdot)) ds' \\
&\qquad - \int_{t_{k_{z+1}}}^{s} g_i(\hat{e}_i(s'), \hat{\bar{x}}_i(\cdot), \bar{u}_i(\cdot)) ds' \Bigg\| \\
&= \left\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \int_{t_{k_z}}^{t_{k_{z+1}}} \frac{d}{dt} \left[\hat{e}_i(s; \hat{u}_i(\cdot), e_i(t_{k_z})) \right] ds \right\| \\
&+ \Bigg\| \int_{t_{k_{z+1}}}^{s} \frac{d}{dt} \left[\bar{e}_i(s'; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) \right] ds' \\
&\qquad - \int_{t_{k_{z+1}}}^{s} \frac{d}{dt} \left[\hat{e}_i(s'; \bar{u}_i(\cdot), e_i(t_{k_z})) \right] ds' \Bigg\| \\
&= \big\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \hat{e}_i(t_{k_{z+1}}; \hat{u}_i(\cdot), e_i(t_{k_z})) \\
&\qquad + \hat{e}_i(t_{k_z}; \hat{u}_i(\cdot), e_i(t_{k_z})) \big\| \\
&+ \big\| \bar{e}_i(s; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) - \bar{e}_i(t_{i+1}; \bar{u}_i(\cdot), e_i(t_{k_{z+1}})) \\
&- \hat{e}_i(s; \bar{u}_i(\cdot), e_i(t_{k_z})) + \hat{e}_i(t_{k_{z+1}}; \bar{u}_i(\cdot), e_i(t_{k_z})) \big\| \\
&= \big\| e_i(t_{k_{z+1}}) - e_i(t_{k_z}) - \hat{e}_i(t_{k_{z+1}}; \hat{u}_i(\cdot), e_i(t_{k_z})) + e_i(t_{k_z}) \big\| \\
&= \big\| e_i(t_{k_{z+1}}) - \hat{e}_i(t_{k_{z+1}}; \hat{u}_i(\cdot), e_i(t_{k_z})) \big\| ,
\end{aligned}
$$

which, by employing Lemma 2, leads to:

$$
\|\bar{e}_1(s) - \hat{e}_2(s)\| \leq \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h).
$$

By combining the last result with (50) we get:

$$
\begin{aligned}
&\int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \\
&\qquad\qquad - \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \Big[F_i(\hat{e}_2(s), \hat{u}_2(s)) \Big] ds \\
&\leq L_{F_i} \int_{t_{k_{z+1}}}^{t_{k_z}+T-h} \rho_i(h) ds = (T - 2h)\rho_i(h) L_{F_i}. \qquad (52)
\end{aligned}
$$

By combining the last result with (50), (47) becomes:

$$
\begin{aligned}
&\bar{J}(e_i(t_{k_{z+1}})) - J^\star(e_i(t_{k_z})) \leq (T - 2h)\rho_i(h) L_{F_i} \\
&\qquad + V_i(\bar{e}_1(t_{k_z} + T)) - V_i(\hat{e}_2(t_{k_z} + T - h)) \\
&\qquad + \int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \\
&\qquad - \int_{t_{k_z}}^{t_{k_{z+1}}} \Big[F_i(\hat{e}_2(s), \hat{u}_2(s)) \Big] ds \qquad (53)
\end{aligned}
$$

By integrating inequality (22) from $t_{k_z} + T - h$ to $t_{k_z} + T$ and we get the following:

$$
\begin{aligned}
&\int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[\frac{\partial V}{\partial e} \cdot g_i(\bar{e}_1(s), \hat{\bar{x}}_i(s), \bar{u}_1(s)) \\
&\qquad\qquad + F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \leq 0 \\
&\Leftrightarrow V_i(\bar{e}_1(t_{k_z} + T) - V_i(\bar{e}_1(t_{k_z} + T - h)) \\
&\qquad + \int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \leq 0,
\end{aligned}
$$

which by adding and subtracting the term $V_i(\hat{e}_2(t_{k_z} + T - h))$ becomes:

$$
\begin{aligned}
&V_i(\bar{e}_1(t_{k_z} + T) - V_i(\hat{e}_2(t_{k_z} + T - h)) \\
&\qquad + \int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \leq \\
&\qquad V_i(\bar{e}_1(t_{k_z} + T - h)) - V_i(\hat{e}_2(t_{k_z} + T - h)).
\end{aligned}
$$

By employing the property $y \leq |y|, \forall y \in \mathbb{R}$, we get:

$$
\begin{aligned}
&V_i(\bar{e}_1(t_{k_z} + T) - V_i(\hat{e}_2(t_{k_z} + T - h)) \\
&\qquad + \int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s)) \Big] ds \leq \\
&\qquad |V_i(\bar{e}_1(t_{k_z} + T - h)) - V_i(\hat{e}_2(t_{k_z} + T - h))| . \qquad (54)
\end{aligned}
$$

By employing Lemma 3, we have that:

$$
\begin{aligned}
&|V_i(\bar{e}_1(t_{k_z} + T - h)) - V_i(\hat{e}_2(t_{k_z} + T - h))| \leq \\
&\qquad L_{V_i} \|\bar{e}_1(t_{k_z} + T - h) - \hat{e}_2(t_{k_z} + T - h)\|,
\end{aligned}
$$

which by employing Lemma 5 and (49), becomes:

$$
\begin{aligned}
&|V_i(\bar{e}_1(t_{k_z} + T - h)) - V_i(\hat{e}_2(t_{k_z} + T - h))| \leq \\
&\qquad L_{V_i} \rho_i(t_{k_{z+1}} - t_{k_z}) = \rho_i(h) L_{V_i}.
\end{aligned}
$$

By combining the last result with (54), we get:

$$V_i(\bar{e}_1(t_{k_z}+T) - V_i(\hat{e}_2(t_{k_z}+T-h)) + \int_{t_{k_z}+T-h}^{t_{k_z}+T} \Big[F_i(\bar{e}_1(s), \bar{u}_1(s))\Big] ds \leq \rho_i(h) L_{V_i}.$$

The last inequality along with (53) leads to:

$$\bar{J}(e(t_{k_{z+1}})) - J^\star(e(t_{k_z})) \leq (T-2h)\rho_i(h) L_{F_i} + \rho_i(h) L_{V_i} - \int_{t_{k_z}}^{t_{k_{z+1}}} \Big[F_i(\hat{e}_2(s), \hat{u}_2(s))\Big] ds. \quad (55)$$

By substituting $e_i = \hat{e}_2(s), u_i = \hat{u}_2(s)$ in (14) we get $F_i(\hat{e}_2(s), \hat{u}_2(s)) \geq m_i \|\hat{e}_2(s)\|^2$, or equivalently:

$$\int_{t_{k_z}}^{t_{k_{z+1}}} \Big[F_i(\hat{e}_2(s), \hat{u}_2(s))\Big] ds \geq \underline{m}_i \int_{t_{k_z}}^{t_{k_{z+1}}} \|\hat{e}_2(s)\|^2 ds$$
$$\Leftrightarrow - \int_{t_{k_z}}^{t_{k_{z+1}}} \Big[F(\hat{e}_2(s), \hat{u}_2(s))\Big] ds \leq -\underline{m}_i \int_{t_{k_z}}^{t_{k_{z+1}}} \|\hat{e}_2(s)\|^2 ds.$$

By combining the last result with (55), we get:

$$\bar{J}_i(e(t_{k_{z+1}})) - J_i^\star(e(t_{k_z})) \leq (T-2h)\rho_i(h) L_{F_i} + \rho_i(h) L_{V_i} - \underline{m}_i \int_{t_{k_z}}^{t_{k_{z+1}}} \|\hat{e}_2(s)\|^2 ds \quad (56)$$

It is clear that the optimal solution at time $t_{k_{z+1}}$ i.e., $J^\star(e_i(t_{k_{z+1}}))$ will not be worse than the feasible one at the same time i.e. $\bar{J}(e_i(t_{k_{z+1}}))$. Therefore, (56) implies:

$$J_i^\star(e_i(t_{k_{z+1}})) - J_i^\star(e_i(t_{k_z})) \leq (T-2h)\rho_i(h) L_{F_i} + \rho_i(h) L_{V_i} - \underline{m}_i \int_{t_{k_z}}^{t_{k_{z+1}}} \|\hat{e}_2(s)\|^2 ds,$$

which is equivalent to:

$$J_i^\star(e_i(t_{k_{z+1}})) - J_i^\star(e_i(t_{k_z})) \leq - m_i \int_{t_{k_z}}^{t_{k_{z+1}}} \|\hat{e}_i(s; \hat{u}_i^\star(s; e_i(t_{k_z})), e_i(t_{k_z}))\|^2 ds + (T-2h)\rho_i(h) L_{F_i} + \rho_i(h) L_{V_i}.$$

which, according to (1), is in the form:

$$J_i^\star(e_i(t_{k_{z+1}})) - J_i^\star(e_i(t_{k_z})) \leq -\alpha(\|e_i\|) + \sigma(\|\bar{x}_i\|) \quad (57)$$

Thus, the optimal cost J has been proven to be decreasing, and according to Definition 4 and Theorem 5, the closed loop system is ISS stable. Therefore, the closed loop trajectories converges to the closed set $\mathcal{E}_i$. □

References

[1] S. Loizou and K. Kyriakopoulos, "Automatic Synthesis of Multi-Agent Motion Tasks Based on LTL Specifications," *43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, pp. 153–158, 2004.

[2] T. Wongpiromsarn, U. Topcu, and R. Murray, "Receding Horizon Control for Temporal Logic Specifications," *13th ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 101–110, 2010.

[3] M. Guo and D. Dimarogonas, "Multi-Agent Plan Reconfiguration Under Local LTL Specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015.

[4] S. Karaman and E. Frazzoli, "Linear Temporal Logic Vehicle Routing with Applications to Multi-UAV Mission Planning," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1372–1395, 2011.

[5] Y. Kantaros and M. Zavlanos, "A Distributed LTL-Based Approach for Intermittent Communication in Mobile Robot Networks," *American Control Conference (ACC)*, pp. 5557–5562, 2016.

[6] G. Fainekos, A. Girard, K. Hadas, and G. Pappas, "Temporal logic motion planning for Dynamic Robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[7] M. Kloetzer and C. Belta, "Automatic Deployment of Distributed Teams of Robots From Temporal Motion Specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[8] M. Kloetzer, X. C. Ding, and C. Belta, "Multi-Robot Deployment from LTL Specifications with Reduced Communication," *50th IEEE Conference on Decision and Control (CDC)*, pp. 4867–4872, 2011.

[9] J. Liu and P. Prabhakar, "Switching Control of Dynamical Systems from Metric Temporal Logic Specifications," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5333–5338, 2014.

[10] V. Raman, A. Donzé, D. Sadigh, R. Murray, and S. Seshia, "Reactive Synthesis from Signal Temporal Logic Specifications," *18th International Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 239–248, 2015.

[11] J. Fu and U. Topcu, "Computational Methods for Stochastic Control with Metric Interval Temporal Logic Specifications," *54th IEEE Conference on Decision and Control (CDC)*, pp. 7440–7447, 2015.

[12] B. Hoxha and G. Fainekos, "Planning in Dynamic Environments Through Temporal Logic Monitoring," 2016.

[13] Y. Zhou, D. Maity, and J. S. Baras, "Timed Automata Approach for Motion Planning Using Metric Interval Temporal Logic," *European Control Conference (ECC)*, 2016.

[14] S. Karaman and E. Frazzoli, "Vehicle Routing Problem with Metric Temporal Logic Specifications," *47th IEEE Conference on Decision and Control (CDC 2008)*, pp. 3953–3958, 2008.

[15] A. Nikou, J. Tumova, and D. Dimarogonas, "Cooperative Task Planning of Multi-Agent Systems Under Timed Temporal Specifications," *American Control Conference (ACC)*, pp. 13–19, 2016.

[16] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete Abstractions of Hybrid Systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 971–984, 2000.

[17] C. Belta and V. Kumar, "Abstraction and Control for Groups of Robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[18] M. Helwa and P. Caines, "In-Block Controllability of Affine Systems on Polytopes," *53rd IEEE Conference on Decision and Control (CDC)*, pp. 3936–3942, 2014.

[19] L. Habets, P. Collins, and V. Schuppen, "Reachability and Control Synthesis for Piecewise-Affine Hybrid Systems on Simplices," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 938–948, 2006.

[20] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic Models for Nonlinear Control Systems without Stability Assumptions," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, 2012.

[21] J. Liu and N. Ozay, "Finite Abstractions With Robustness Margins for Temporal Logic-Based Control Synthesis," *Nonlinear Analysis: Hybrid Systems*, vol. 22, pp. 1–15, 2016.

[22] M. Zamani, M. Mazo, and A. Abate, "Finite Abstractions of Networked Control Systems," *53rd IEEE Conference on Decision and Control (CDC)*, pp. 95–100, 2014.

[23] G. Pola, P. Pepe, and M. D. D. Benedetto, "Symbolic Models for Networks of Control Systems," *IEEE Transactions on Automatic Control*, 2016.

[24] D. Boskos and D. Dimarogonas, "Decentralized Abstractions For Multi-Agent Systems Under Coupled Constraints," *54th IEEE Conference on Decision and Control (CDC)*, pp. 282–287, 2015.

[25] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "Cooperative Planning for Coupled Multi-Agent Systems under Timed Temporal Specifications," *Americal Control Conference 2017*.

[26] H. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 1996.

[27] E. D. Sontag, "Input to State Stability: Basic Concepts and Results," pp. 163–220, 2008.

[28] E. D. Sontag and Y. Wang, "On Characterizations of the Input-to-State Stability Property," *Systems and Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[29] R. Alur and D. Dill, "A Theory of Timed Automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[30] D. D. Souza and P. Prabhakar, "On the Expressiveness of MTL in the Pointwise and Continuous Semantics," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 1, pp. 1–4, 2007.
[31] J. Ouaknine and J. Worrell, "On the Decidability of Metric Temporal Logic," *20th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pp. 188–197, 2005.
[32] R. Alur, T. Feder, and T. A. Henzinger, "The Benefits of Relaxing Punctuality," *Journal of the ACM (JACM)*, vol. 43, no. 1, pp. 116–146, 1996.
[33] M. Reynolds, "Metric Temporal Logics and Deterministic Timed Automata," 2010.
[34] P. Bouyer, "From Qualitative to Quantitative Analysis of Timed Systems," *Mémoire dhabilitation, Université Paris*, vol. 7, pp. 135–175, 2009.
[35] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "Cooperative Planning for Coupled Multi-Agent Systems under Timed Temporal Specifications," *http://arxiv.org/pdf/1603.05097v2.pdf*.
[36] S. Tripakis, "Checking Timed Buchi Automata Emptiness on Simulation Graphs," *ACM Transactions on Computational Logic (TOCL)*, vol. 10, no. 3, 2009.
[37] O. Maler, D. Nickovic, and A. Pnueli, "From MITL to Timed Automata," *International Conference on Formal Modeling and Analysis of Timed Systems*, pp. 274–289, 2006.
[38] D. Ničković and N. Piterman, "From MTL to Deterministic Timed Automata," *Formal Modeling and Analysis of Timed Systems*, 2010.
[39] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
[40] M. Andreasson, D. V. Dimarogonas, H. Sandberg, and K. H. Johansson, "Distributed Control of Networked Dynamical Systems: Static Feedback, Integral Action and Consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1750–1764, 2014.
[41] K. S. de Oliveira and M. Morari, "Contractive Model Predictive Control for Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1053–1071, 2000.
[42] B. Kouvaritakis and M. Cannon, *Nonlinear Predictive Control: Theory and Practice*. No. 61, Iet, 2001.
[43] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss, "State and Output Feedback Nonlinear Model Predictive Control: An Overview," *European Journal of Control*, vol. 9, no. 2-3, pp. 190–206, 2003.
[44] H. Chen and F. Allgöwer, "A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
[45] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss, "Towards a Sampled-Data Theory for Nonlinear Model Predictive Control," *New Trends in Nonlinear Dynamics and Control and their Applications*, pp. 295–311, 2003.
[46] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer London, 2011.
[47] E. Camacho and C. Bordons, "Nonlinear Model Predictive Control: An Introductory Review," pp. 1–16, 2007.
[48] J. Frasch, A. Gray, M. Zanon, H. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An Auto-Generated Nonlinear MPC Algorithm for Real-Time Obstacle Cvoidance of Ground Vehicles," *European Control Conference (ECC)*, 2013.
[49] F. Fontes, "A General Framework to Design Stabilizing Nonlinear Model Predictive Controllers," *Systems and Control Letters*, vol. 42, no. 2, pp. 127–143, 2001.
[50] D. Marruedo, T. Alamo, and E. Camacho, "Input-to-State Stable MPC for Constrained Discrete-Time Nonlinear Systems with Bounded Additive Uncertainties," *IEEE Conference on Decision and Control*, vol. 4, pp. 4619–4624, 2002.
[51] A. Eqtami, D. Dimarogonas, and K. Kyriakopoulos, "Novel Event-Triggered Strategies for Model Predictive Controllers," *IEEE Conference on Decision and Control*, pp. 3392–3397, 2011.
[52] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
[53] C. Baier, J. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.