# Cooperative Planning for Coupled Multi-Agent Systems under Timed Temporal Specifications

Alexandros Nikou, Dimitris Boskos, Jana Tumova and Dimos V. Dimarogonas

*Abstract*— **This paper presents a fully automated procedure for controller synthesis for multi-agent systems under coupled constraints. Each agent has dynamics consisting of two terms: the first one models the coupled constraints and the other one is an additional 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). First, a decentralized abstraction that provides a time and space discretization of the multi-agent system is designed. Second, by utilizing this abstraction and techniques from formal verification, we provide an algorithm that computes the individual runs which provably satisfy the high-level tasks. The overall approach is demonstrated in a simulation example.**

## 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, formation and rendez-vous ([1]–[5]) and at the same time fulfill properties such as network connectivity ([6], [7]). Over the last few years, multi-agent control under complex high-level specifications has been gaining significant attention. In particular, coordination of multi-robot teams under qualitative temporal tasks constitutes an emerging application in this field. In this work, we aim to additionally introduce specific time bounds into these tasks, in order to include specifications such as "Visit region $A$ within 5 time units" or "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 20 time units".

The specification language that has primarily been used to express the tasks is Linear Temporal Logic (LTL) (see, e.g., [8]). LTL has been proven a valuable tool for controller synthesis, because it provides a compact mathematical formalism for specifying desired behaviors of a system. There is a rich body of literature containing algorithms for verification and synthesis of multi-agent systems under temporal logic specifications ([9], [10]). A common approach in multi-agent planning under LTL specifications is the consideration of a centralized, global task for the team, which is then decomposed into local tasks to be accomplished by the individual agents (see [11], [12]). A three-step hierarchical procedure to address this problem is described as follows ([13]): first, the robot dynamics is abstracted into a discrete transition system using sampling or cell decomposition methods based on triangular, rectangular or other partitions. Second, invoking ideas from formal verification, a discrete plan that meets the high-level task is synthesized. Third, the discrete plan is translated into a sequence of continuous controllers for the original system.

Time constraints in the system modeling have been considered e.g., in [14]–[16]. Both the aforementioned, as well as most existing works on multi-agent planning, consider temporal properties which treat time in a qualitative manner. However, for real applications, a multi-agent team might be required to perform a specific task within a certain time bound, rather than at some arbitrary time in the future, i.e. in a quantitative manner. Timed specifications have been considered in [17]–[21]. However, all these works are restricted to single agent planning and they are not extendable to multi-agent systems in a straightforward way.

The multi-agent case has been considered in [22], 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 [23], where Metric Interval Temporal Logic (MITL) formulas were introduced in order to synthesize controllers such that every agent fulfills an individual specification and the team of agents fulfill a global specification.

In [23], the abstraction of the dynamics was given and an upper bound of the time that each agent needs to finish a transition from one region to another was assumed. Furthermore, potential coupled constraints between the agents were not taken into consideration. In this work, we aim to address the aforementioned issues. The dynamics of each agent consists of two parts: the first part is a consensus type term 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, it will be called a free input. A decentralized abstraction procedure is provided, which leads to an individual Transition System (TS) for each agent and provides the basis for high-level planning. Additionally, this abstraction is associated to a time quantization which allows us to assign precise time durations to the transitions of each agent.

There is a rich literature on abstractions for dynamical systems (see e.g., [24]–[27]). Multi-agent abstractions have been addressed in [28]–[32]. Motivated by [32], we start from the dynamics of each agent and we construct a TS for each agent in a decentralized manner. An individual task is assigned to each agent and we aim to design the free inputs so that each

The authors 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, boskos, tumova, dimos}@kth.se. This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation.

agent performs the desired individual task within specific time bounds. To the best of the authors' knowledge, this is the first time that a fully automated framework for multi-agent systems consisting of both constructing an abstraction and conducting high-level timed temporal logic planning is considered. Hence, this works lies in the intersection of the fields of multi-agent systems, abstractions and timed formal verification.

The contribution of this paper is to provide an automatic controller synthesis method of a general framework of coupled multi-agent systems under high-level tasks with timed constraints. Compared to the existing works on multi-agent planning under temporal logic specifications, the proposed approach yields the first solution to the problem of planning of dynamically coupled multi-agent systems under timed temporal specifications in a distributed way.

The remainder of the paper is structured as follows. In Sec. II a description of the necessary mathematical tools, the notations and the definitions are given. Sec. III provides the dynamics of the system and the formal problem statement. Sec. IV discusses the technical details of the solution. Sec. V is devoted to a simulation example. Finally, the conclusions and the future work directions are discussed in Sec. 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. Also, define $\mathbb{T}_\infty = \mathbb{T} \cup \{\infty\}$ for a set $\mathbb{T} \subseteq \mathbb{R}$. Given a set $S$, we denote by $|S|$ its cardinality and by $2^S$ the set of all its subsets. For a subset $S$ of $\mathbb{R}^n$, we denote by $\text{cl}(S), \text{int}(S)$ and $\partial S = \text{cl}(S) \backslash \text{int}(S)$ its closure, interior and boundary, respectively and $\backslash$ is used for set subtraction. The notation $\|x\|$ is used for the Euclidean norm of a vector $x \in \mathbb{R}^n$ and $\|A\| = \max\{\|Ax\| : \|x\| = 1\}$ for the induced norm of a matrix $A \in \mathbb{R}^{m \times n}$. Given a matrix $A$, the spectral radius of $A$ is denoted by $\lambda_{\max}(A) = \max\{|\lambda| : \lambda \in \sigma(A)\}$, where $\sigma(A)$ is the set of all the eigenvalues of $A$.

### B. Multi-Agent Systems

Consider a set of agents $\mathcal{I} = \{1, 2, \dots, N\}$ operating in $\mathbb{R}^n$. The topology of the multi-agent network is modeled through a static undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I}$ is the set of nodes (agents) and $\mathcal{E} \subseteq \{\{i, j\} : i, j \in \mathcal{I}, i \neq j\}$ is the set of edges (denoting the communication capability between neighboring respective agents). For each agent, its neighbors' set $\mathcal{N}(i)$ is defined as $\mathcal{N}(i) = \{j_1, \dots, j_{N_i}\} = \{j \in \mathcal{I} : \{i, j\} \in \mathcal{E}\}$ where $N_i = |\mathcal{N}(i)|$.

Given a vector $x_i = (x_i^1, \dots, x_i^n) \in \mathbb{R}^n$, the component operator $c(x_i, k) = x_i^k \in \mathbb{R}, k = 1, \dots, n$ gives the projection of $x_i$ onto its $k$-th component (see [33]). Similarly, for the stack vector $x = (x_1, \dots, x_N) \in \mathbb{R}^{Nn}$ the component operator is defined as $c(x, k) = (c(x_1, k), \dots, c(x_N, k))$
$\in \mathbb{R}^N, k = 1, \dots, n$. By using the component operator, the norm of a vector $x \in \mathbb{R}^{Nn}$ can be computed as $\|x\| = \left\{\sum_{k=1}^{n} \|c(x, k)\|^2\right\}^{\frac{1}{2}}$.

The Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{N \times N}$ of the graph $\mathcal{G}$ is defined as $L(\mathcal{G}) = D(\mathcal{G}) D(\mathcal{G})^\tau$ where $D(\mathcal{G})$ is the $N \times |\mathcal{E}|$ incidence matrix ([33]). The graph Laplacian $L(\mathcal{G})$ is positive semidefinite and symmetric. By considering an ordering $0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \ldots \leq \lambda_N(\mathcal{G}) = \lambda_{\max}(\mathcal{G})$ of the eigenvalues of $L(\mathcal{G})$ then we have that $\lambda_2(\mathcal{G}) > 0$ iff $\mathcal{G}$ is connected ([33]).

We denote by $\tilde{x} \in \mathbb{R}^{|\mathcal{E}|n}$ the stack column vector of the vectors $x_i - x_j, \{i, j\} \in \mathcal{E}$ with the edges ordered as in the case of the incidence matrix. Thus, the following holds:

$$\tilde{x} = D(\mathcal{G})^\tau x. \tag{1}$$

### C. Cell Decompositions

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

**Definition 1.** A *cell decomposition* $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ of a set $\mathcal{D} \subseteq \mathbb{R}^n$, where $\mathbb{I} \subseteq \mathbb{N}$ is a finite or countable index set, is a family of uniformly bounded convex sets $S_\ell, \ell \in \mathbb{I}$ such that $\text{int}(S_\ell) \cap \text{int}(S_{\hat{\ell}}) = \emptyset$ for all $\ell, \hat{\ell} \in \mathbb{I}$ with $\ell \neq \hat{\ell}$ and $\cup_{\ell \in \mathbb{I}} S_\ell = \mathcal{D}$.

**Example 1.** An example of a cell decomposition with $\mathbb{I} = \{1, 2, 3, 4, 5, 6\}$ and $S = \{S_\ell\}_{\ell \in \mathbb{I}} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$ is depicted in Fig. 1. This cell decomposition will be used as reference for the following examples.

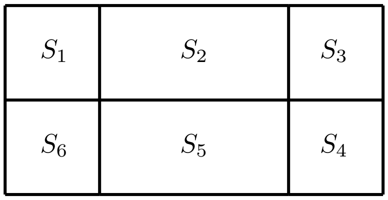


Fig. 1: An example of a cell decomposition with $|\mathbb{I}| = 6$ cells

### D. Time Sequence, Timed Run and Weighted Transition System

In this section we review some basic definitions from computer science that are required in the sequel.

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 $i$-th element of a sequence is denoted by $\chi(i)$.

**Definition 2.** ([34]) A *time sequence* $\tau = \tau(0)\tau(1)\ldots$ is an infinite sequence of time values $\tau(j) \in \mathbb{T} = \mathbb{Q}_+$, satisfying the following properties:

- Monotonicity: $\tau(j) < \tau(j+1)$ for all $j \geq 0$.
- Progress: For every $t \in \mathbb{T}$, there exists $j \geq 1$, such that $\tau(j) > t$.

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

**Definition 3.** ([34]) Let $AP$ be a finite set of atomic propositions. A *timed word* $w$ over the set $AP$ is an infinite sequence $w^t = (w(0), \tau(0))(w(1), \tau(1))\ldots$ where $w(0)w(1)\ldots$ is an infinite word over the set $2^{AP}$ and $\tau(0)\tau(1)\ldots$ is a time sequence with $\tau(j) \in \mathbb{T},\ j \geq 0$.

**Definition 4.** A Weighted Transition System (*WTS*) is a tuple $(S, S_0, Act, \longrightarrow, d, AP, L)$ 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 (time values in this framework) to each transition; $AP$ is a finite set of atomic propositions; and $L : S \to 2^{AP}$ is a labeling function. For simplicity, the notation $s \xrightarrow{\alpha} s'$ is used to denote that $(s, \alpha, s') \in \longrightarrow$ for $s, s' \in S$ and $\alpha \in Act$. Furthermore, for every $s \in S$ and $\alpha \in Act$ the operator $\text{Post}(s, \alpha) = \{s' \in S : (s, \alpha, s') \in \longrightarrow\}$ is defined.

**Definition 5.** 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 $j \geq 1$, it holds that $r(j) \in S$ and $(r(j), \alpha(j), r(j+1)) \in \longrightarrow$ for a sequence of actions $\alpha(1)\alpha(2)\dots$ with $\alpha(j) \in Act, \forall\ j \geq 1$. The *time stamps* $\tau(j), j \geq 0$ are inductively defined as

1) $\tau(0) = 0$.
2) $\tau(j+1) = \tau(j) + d(r(j), r(j+1)),\ \forall\ j \geq 1$.

Every timed run $r^t$ generates a *timed word* $w(r^t) = (w(0), \tau(0))\ (w(1), \tau(1)) \dots$ over the set $2^{AP}$ where $w(j) = L(r(j)),\ \forall\ j \geq 0$ is the subset of atomic propositions that are true at state $r(j)$.

### *E. Metric Interval Temporal Logic*

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

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1\ \mathcal{U}_I\ \varphi_2 \tag{2}$$

where $p \in AP$, and $\bigcirc$, $\Diamond$, $\Box$ and $\mathcal{U}$ are the next, eventually, always and until temporal operator, respectively. $I \subseteq \mathbb{T}$ is a non-empty time interval in one of the following forms: $[i_1, i_2], [i_1, i_2), (i_1, i_2], (i_1, i_2), [i_1, \infty], (i_1, \infty)$ where $i_1, i_2 \in \mathbb{T}$ with $i_1 < i_2$. MITL can be interpreted either in continuous or point-wise semantics [35]. The latter approach is utilized, since the consideration of point-wise (event-base) type semantics renders a framework that includes Transition Systems and automata construction, namely our current approach, more natural. The MITL formulas are interpreted over timed runs such as the ones produced by a WTS (Def. 5).

**Definition 6.** ([35], [36]) Given a timed word $w^t = (w(0), \tau(0))(w(1), \tau(1)) \dots$ and an MITL formula $\varphi$, we define $(w^t, i) \models \varphi$, for $i \geq 0$ (read $w^t$ satisfies $\varphi$ at position $i$) as follows:

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

It has been proved that MITL is decidable in both finite and infinite words [37] and in both pointwise and continuous semantics [38]. The model checking and satisfiability problems are *EXPSPACE*-complete.

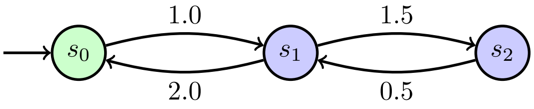


Fig. 2: An example of a WTS

**Example 2.** Consider the WTS $\mathcal{T}$ with $S = \{s_0, s_1, s_2\}, S_0 = \{s_0\}, Act = \emptyset, \longrightarrow = \{(s_0, \emptyset, s_1), (s_1, \emptyset, s_2), (s_1, \emptyset, s_0), (s_2, \emptyset, s_1)\}, d((s_0, \emptyset, s_1)) = 1.0, d((s_1, \emptyset, s_2)) = 1.5, d((s_1, \emptyset, s_0)) = 2.0, d((s_2, \emptyset, s_1)) = 0.5, AP = \{green\}, L(s_0) = \{green\}, L(s_1) = L(s_2) = \emptyset$ depicted in Fig. 2.

Let two timed runs of the system: $r_1^t = (s_0, 0.0)(s_1, 1.0)(s_0, 3.0)(s_1, 4.0)\dots, r_2^t = (s_0, 0.0)(s_1, 1.0)(s_2, 2.5)(s_1, 3.0)\dots$ and two MITL formulas $\varphi_1 = \Diamond_{[2,5]}\{green\}, \varphi_2 = \Box_{[0,5]}\{green\}$. According to the MITL semantics, it can be seen that the timed run $r_1^t$ satisfies the formula $\varphi_1$ (we formally write $r_1^t \models \varphi_1$), since at the time stamp $3.0 \in [2, 5]$ we have that $L(s_0) = \{green\}$ so the atomic proposition $green$ occurs at least once in the given interval. On the other hand, the timed run $r_2^t$ does not satisfy the formula $\varphi_2$ (we formally write $r_2^t \not\models \varphi_2$) since the atomic proposition $green$ does not always hold at every time stamp of the runs (it holds only at the time stamp $0.0$).

### *F. Timed Büchi Automata*

*Timed Büchi Automata (TBA)* were introduced in [34]. In this work, the notation from [39], [40] is partially adopted. 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

$$\phi := \top \mid\ \neg\phi \mid \phi_1 \wedge \phi_2 \mid c \bowtie \psi \tag{3}$$

where $c \in C$ is a clock, $\psi \in \mathbb{T}$ is a clock constant and $\bowtie \in \{<, >, \geq, \leq, =\}$. An example of clock constraints for a set of clocks $C = \{c_1, c_2\}$ can be $\Phi(C) = \{c < c_1 \vee c > c_2\}$. A clock *valuation* is a function $\nu : C \to \mathbb{T}$ that assigns a value to each clock. A clock $c_i$ has valuation $\nu_i$ for $i \in \{1, \dots, |C|\}$, and $\nu = (\nu_1, \dots, \nu_{|C|})$. By $\nu \models \phi$ is denoted the fact that the valuation $\nu$ satisfies the clock constraint $\phi$.

**Definition 7.** A *Timed Büchi Automaton* is a tuple $\mathcal{A} = (Q, Q^{\text{init}}, C, Inv, E, F, AP, \mathcal{L})$ 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; $AP$ is a finite set of atomic propositions; and $\mathcal{L} : Q \to 2^{AP}$ labels every state with a subset of atomic propositions.

A state of $\mathcal{A}$ is a pair $(q, \nu)$ where $q \in Q$ and $\nu$ satisfies the *invariant* $Inv(q)$, i.e., $\nu \models Inv(q)$. The initial state of $\mathcal{A}$ is $(q(0), (0, \dots, 0))$, where $q(0) \in Q^{\text{init}}$. Given two states $(q,\nu)$ and $(q',\nu')$ and an edge $e=(q,\gamma,R,q')$, there exists a *discrete transition* $(q,\nu)\xrightarrow{e}(q',\nu')$ iff $\nu\models\gamma$, $\nu'\models Inv(q')$, and $R$ is the *reset set*, i.e., $\nu'_i=0$ for $c_i\in R$ and $\nu'_i=\nu_i$ for $c_i\notin R$. Given a $\delta\in\mathbb{T}$, there exists a *time transition* $(q,\nu)\xrightarrow{\delta}(q',\nu')$ iff $q=q',\nu'=\nu+\delta$ ($\delta$ is summed component-wise) and $\nu'\models Inv(q)$. We write $(q,\nu)\xrightarrow{\delta}\xrightarrow{e}(q',\nu')$ if there exists $q'',\nu''$ such that $(q,\nu)\xrightarrow{\delta}(q'',\nu'')$ and $(q'',\nu'')\xrightarrow{e}(q',\nu')$ with $q''=q$.

An infinite run of $\mathcal{A}$ starting at state $(q(0),\nu)$ is an infinite sequence of time and discrete transitions $(q(0),\nu(0))\xrightarrow{\delta_0}(q(0)',\nu(0)')\xrightarrow{e_0}(q(1),\nu(1))\xrightarrow{\delta_1}(q(1)',\nu(1)')\ldots$, where $(q(0),\nu(0))$ is an initial state. This run produces the timed word $w=(\mathcal{L}(q(0)),\tau(0))(\mathcal{L}(q(1)),\tau(1))\ldots$ with $\tau(0)=0$ and $\tau(i+1)=\tau(i)+\delta_i, \forall\, i\geq 1$. The run is called *accepting* if $q(i)\in F$ for infinitely many times. A timed word is *accepted* if there exists an accepting run that produces it. The problem of deciding the emptiness of the language of a given TBA $\mathcal{A}$ is PSPACE-complete [34]. In other words, an accepting run of a given TBA $\mathcal{A}$ can be synthesized, if one exists.

Any MITL formula $\varphi$ over $AP$ can be algorithmically translated to a TBA with the alphabet $2^{AP}$, such that the language of timed words that satisfy $\varphi$ is the language of timed words produced by the TBA ([37], [41], [42]).

**Example 3.** The TBA $\mathcal{A}$ with $Q=\{q_0,q_1,q_2\},Q^{\text{init}}=\{q_0\},C=\{c\},Inv(q_0)=Inv(q_1)=Inv(q_2)=\emptyset,E=\{(q_0,\{c\leq c_2\},\emptyset,q_0),(q_0,\{c\leq c_1\vee c>c_2\},c,q_2),(q_0,\{c\geq c_1\wedge c\leq c_2\},c,q_1),(q_1,\top,c,q_1),(q_2,\top,c,q_2)\},F=\{q_1\},AP=\{green\},\mathcal{L}(q_0)=\mathcal{L}(q_2)=\emptyset,\mathcal{L}(q_1)=\{green\}$ that accepts all the timed words that satisfy the formula $\varphi=\Diamond_{[c_1,c_2]}\{green\}$ is depicted in Fig. 3. This formula will be used as reference for the following examples and simulations.

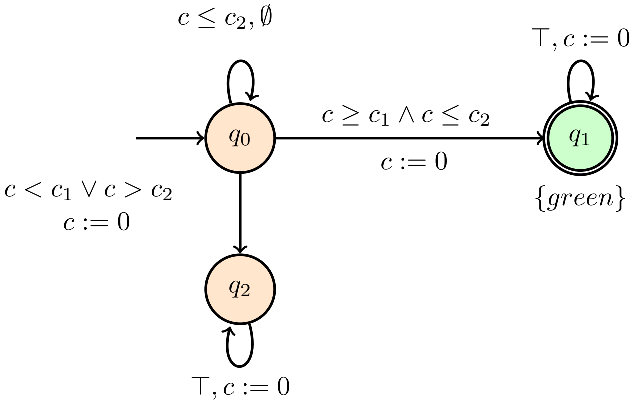


Fig. 3: A TBA $\mathcal{A}$ that accepts the runs that satisfy formula $\varphi=\Diamond_{[c_1,c_2]}\{green\}$.

An example of a timed run of this TBA is $(q_0,0)\xrightarrow{\delta=\alpha_1}(q_0,\alpha_1)\xrightarrow{e=(q_0,\{c\geq c_1\wedge c\leq c_2\},c,q_1)}(q_1,0)\ldots$ with $c_1\leq\alpha_1\leq c_2$, which generates the timed word $w^t=(\mathcal{L}(q_0),0)(\mathcal{L}(q_0),\alpha_1)(\mathcal{L}(q_1),\alpha_1)\ldots=(\emptyset,0)(\emptyset,\alpha_1)(\{green\},\alpha_1)\ldots$ that satisfies the formula $\varphi$. The timed run $(q_0,0)\xrightarrow{\delta=\alpha_2}(q_0,\alpha_2)\xrightarrow{e=(q_0,\{c\leq c_1\vee c>c_2\},c,q_2)}(q_2,0)\ldots$ with $\alpha_2<c_1$, generates the timed word $w^t=(\mathcal{L}(q_0),0)(\mathcal{L}(q_0),\alpha_2)(\mathcal{L}(q_2),\alpha_2)\ldots=(\emptyset,0)(\emptyset,\alpha_2)(\emptyset,\alpha_2)\ldots$ that does not satisfy the formula $\varphi$.

## III. Problem Formulation

### A. System Model

We focus on multi-agent systems with coupled dynamics of the form

$$\dot{x}_i=-\sum_{j\in\mathcal{N}(i)}(x_i-x_j)+v_i,\ x_i\in\mathbb{R}^n,\ i\in\mathcal{I}. \tag{4}$$

where $v_i\in\mathbb{R}^n,\ i\in\mathcal{I}$. The dynamics (4) consists of two parts; the first part is a consensus protocol representing the coupling between the agent and its neighbors, and the second one is a control input which will be exploited for high-level planning and is called free input. In this work, it is assumed that the free inputs are bounded by a positive constant $v_{\text{max}}$. Namely, $\|v_i(t)\|\leq v_{\text{max}},\ \forall\, i\in\mathcal{I},t\geq 0$.

**Assumption 1.** We assume that the communication graph $\mathcal{G}=(\mathcal{I},\mathcal{E})$ of the system is undirected and static i.e., every agent preserves the same neighbors for all times.

Notice that the system (4) can be also expressed in the form $c(\dot{x},k)=-L(\mathcal{G})\ c(x,k)+c(v,k),k\in\{1,\ldots,n\}$ where $x,v\in\mathbb{R}^{Nn}$ are obtained by invoking the definition of the component operator from Sec. II-B.

### B. Specification

Our goal is to control the multi-agent system (4) 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* $\mathbb{R}^n$ within certain time limits and provide certain atomic tasks there. Atomic tasks are captured through a finite set of *services* $\Sigma_i, i\in\mathcal{I}$. Hence, it is desired to relate the position $x_i$ of each agent $i\in\mathcal{I}$ in the workspace with the services that are offered at $x_i$. Initially, a labeling function

$$\Lambda_i:\mathbb{R}^n\to 2^{\Sigma_i} \tag{5}$$

is introduced for each agent $i\in\mathcal{I}$ which maps each state $x_i\in\mathbb{R}^n$ to the subset of services $\Lambda_i(x_i)$ which hold true at $x_i$ i.e., the subset of services that agent $i$ *can* provide in position $x_i$. It should be noted that although the term labeling function it is used, these functions should not be confused with the labeling functions of a WTS as in Definition 4. The union of all the labeling functions as $\Lambda(x)=\bigcup_{i\in\mathcal{I}}\Lambda_i(x)$ is also defined.

Without loss of generality, we assume that $\Sigma_i\cap\Sigma_j=\emptyset$, for all $i,j\in\mathcal{I},i\neq j$ which means that the agents do not share any services. Let us now introduce the following assumption which is important for defining the problem properly.

**Assumption 2.** There exists a partition $S=\{S_\ell\}_{\ell\in\mathbb{I}}$ of the workspace which forms a cell decomposition according to Definition 1 and respects the labeling function $\Lambda$ i.e., for all $S_\ell\in S$ it holds that $\Lambda(x)=\Lambda(x'),\forall\ x,x'\in S_\ell$. This assumption intuitively and again without loss of generality, means that the same services hold at all the points that belong to the same cell of the partition.

Define now for each agent $i \in \mathcal{I}$ a labeling function

$$\mathcal{L}_i : S \to 2^{\Sigma_i} \tag{6}$$

which denotes the fact that when agent $i$ visits a region $S_\ell \in S$, it chooses to *provide* a subset of services that are being offered there i.e., it chooses to satisfy a subset of $\mathcal{L}_i(S_\ell)$.

The trajectory of each agent $i$ is denoted by $x_i(t), t \geq 0, i \in \mathcal{I}$. The trajectory $x_i(t), i \in \mathcal{I}$ is associated with a unique sequence $r^t_{x_i} = (r_i(0), \tau_i(0))(r_i(1), \tau_1(1))(r_i(2), \tau_i(2))\ldots$ of regions that the agent $i$ crosses, where for all $j \geq 0$, $r_i(j) \in S_\ell$ for some $\ell \in \mathbb{I}$, $\Lambda_i(x_i(t)) = \mathcal{L}_i(r_i(j)), \forall\ t \in [\tau_i(j), \tau_i(j+1))$ and $r_i(j) \neq r_i(j+1)$. The equality $\Lambda_i(\cdot) = \mathcal{L}_i(\cdot), i \in \mathcal{I}$ is feasible due to assumption 2. The timed word $w^t_{x_i} = (w_i(0), \tau_i(0))(w_i(1), \tau_i(1))(w_i(2), \tau_i(2))\ldots$, where $w_i(j) = \mathcal{L}_i(r_i(j)), j \geq 0, i \in \mathcal{I}$, is associated uniquely with the trajectory $x_i(t)$ and represents the sequence of services that *can be provided* by the agent $i$ following the trajectory $x_i(t), t \geq 0$.

We define the *timed service word* as

$$\tilde{w}^t_{x_i} = (\beta_i(z_0), \tilde{\tau}_i(z_0))(\beta_i(z_1), \tilde{\tau}_i(z_1))(\beta_i(z_2), \tilde{\tau}_i(z_2))\ldots \tag{7}$$

where $z_0 = 0 < z_1 < z_2 < \ldots$ is a sequence of integers, and for all $j \geq 0$ it holds that $\beta_i(z_j) \subseteq \mathcal{L}_i(r_i(z_j))$ and $\tilde{\tau}(z_j) \in [\tau_i(z_j), \tau_i(z_j+1))$. The timed service word is a sequence of services that are *actually provided* by agent $i$ and it is compliant with the trajectory $x_i(t), t \geq 0$.

The specification task $\varphi_i$ given in MITL formulas over the set of services $\Sigma_i$ as in Definition 6, captures requirements on the services to be provided by agent $i$ for each $i \in \mathcal{I}$. We say that a trajectory $x_i(t)$ satisfies a given formula $\varphi_i$ in MITL over the set of atomic propositions $\Sigma_i$ if and only if there exits a *timed service word*, as defined in (7), that complies with $x_i(t)$ and satisfies $\varphi_i$ according to Definition 6.

**Example 4.** We consider here an example in order to understand the notation and the technical terms that have been introduced until here. Let $N = 2$ agents performing in the partitioned environment of Fig. 4. The agents have the ability to pick up, deliver and throw a different ball each. Let the services of each agent be $\Sigma_1 = \{\mathrm{pickUp1}, \mathrm{deliver1}, \mathrm{throw1}\}$ and $\Sigma_2 = \{\mathrm{pickUp2}, \mathrm{deliver2}, \mathrm{throw2}\}$. Note that $\Sigma_1 \cap \Sigma_2 \neq \emptyset$. We capture 3 points of the trajectories of the agents that belong to different cells where different atomic propositions hold. Let $t_0 = t'_0 = 0$ and $t_1 < t'_1 < t_2 < t_2 < t'_2 < t_3 < t'_3$. The trajectories $x_1(t), x_2(t), t \geq 0$ are depicted with red lines. According to Assumption 2 we have that $S = \{S_\ell\}_{\ell \in \mathbb{I}} = \{S_1, \ldots, S_6\}$ where $\mathbb{I} = \{1, \ldots, 6\}$. We also have

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

where generally $t_0 = t'_0, t_1 \neq t'_1$ and $t_2 \neq t'_2$, $t_3 \neq t'_3$. By computing $w_i(j) = \mathcal{L}(r_i(j)), \forall\ i \in \{1, 2\}, j \in \{1, 2, 3\}$ the corresponding individual timed words are given as:

$$\begin{aligned}
w^t_{x_1} &= (\{\mathrm{pickUp1}\}, \mathrm{t}_0)(\{\mathrm{throw1}, \mathrm{t}_1\})(\{\mathrm{deliver1}\}, \mathrm{t}_2)(\emptyset, \mathrm{t}_3)\\
w^t_{x_2} &= (\{\mathrm{pickUp2}\}, \mathrm{t}'_0)(\{\mathrm{deliver2}\}, \mathrm{t}'_1)(\{\mathrm{throw2}\}, \mathrm{t}'_2)(\emptyset, \mathrm{t}'_3).
\end{aligned}$$

According to (7), two time service words (depicted with in Fig. 4) are given as:

$$\begin{aligned}
\tilde{w}^t_1 &= (\beta_1(z_0), \tilde{\tau}_1(z_0))(\beta_1(z_1), \tilde{\tau}_1(z_1))\\
\tilde{w}^t_2 &= (\beta_2(z'_0), \tilde{\tau}_2(z'_0))(\beta_2(z'_1), \tilde{\tau}_2(z'_1))
\end{aligned}$$

where for agent 1 we have: $z_0 = 0, z_1 = 2, \beta_1(z_0) = \{\mathrm{pickUp1}\} \subseteq \mathcal{L}_1(\mathrm{r}_1(\mathrm{z}_0)), \beta_1(\mathrm{z}_1) = \{\mathrm{deliver1}\} \subseteq \mathcal{L}_1(\mathrm{r}_1(\mathrm{z}_1))$ and $\tilde{\tau}_1(z_0) = \tilde{t}'_0 \in [\tau_1(z_0), \tau_1(z_0+1)) = [t_0, t_1), \tilde{\tau}_1(z_1) = \tilde{t}'_1 \in [\tau_1(z_1), \tau_1(z_1+1)) = [t_2, t_3)$. The corresponding elements for agent 2 are $z'_0 = 0, z'_1 = 2, \beta_2(z'_0) = \{\mathrm{pickUp2}\} \subseteq \mathcal{L}_2(\mathrm{r}_2(\mathrm{z}'_0)), \beta_2(\mathrm{z}'_1) = \{\mathrm{deliver2}\} \subseteq \mathcal{L}_2(\mathrm{r}_2(\mathrm{z}'_1))$ and $\tilde{\tau}_2(z'_0) = \tilde{t}''_0 \in [\tau_2(z'_0), \tau_2(z'_1)) = [t'_0, t'_1), \tilde{\tau}_2(z'_1) = \tilde{t}''_1 \in [\tau_1(z'_1), \tau_1(z'_1+1)) = [t'_2, t'_3)$.

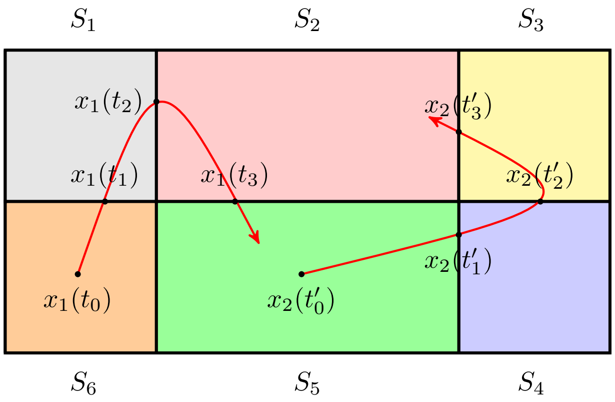


Fig. 4: An example of two agents performing in a partitioned workspace.

### C. Problem Statement

We are now ready to define our problem formally as follows:

**Problem 1.** Given $N$ agents that are governed by dynamics as in (4) and $N$ task specification formulas $\varphi_1, \ldots, \varphi_N$

expressed in MITL over the sets of services $\Sigma_1, \dots, \Sigma_N$, respectively, the partition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ as in Assumption 2 and the labeling functions $\Lambda_1, \dots, \Lambda_N$ and $\mathcal{L}_1, \dots, \mathcal{L}_N$, as in (5), (6) respectively, assign control laws to the free inputs $v_1, \dots, v_N$ such that each agent fulfills its individual specification, given the bound $v_{\max}$.

**Remark 1.** In our preliminary work on the multi-agent controller synthesis framework under MITL specifications [23], 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. Primarily, due to the coupled dynamics of the system, which constrain the motion of each agent, and, secondly, 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 ([9]–[12]).

**Remark 2.** It should be noted here that, in this work, the couplings and the dependencies between the agents are treated by the dynamics of the form (4) and not in the discrete level by coupling also in the services of each agent (i.e., $\Sigma_i \cap \Sigma_j \neq \emptyset, \ \forall i, j \in \mathcal{I}$). Hence, even though the agents do not share atomic propositions, it is the constraint to their motion due to the dynamic couplings with the neighbors that restrict them to fulfill the desired high-level tasks. Treating the coupling through individual atomic propositions in the discrete level as well, constitutes another problem which is far from trivial and a topic of current work.

**Remark 3.** The motivation for introducing the cell decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ in this Section, is that it is required to know which services hold in each part of the workspace. As will be witnessed through the problem solution, this is necessary since the abstraction of the workspace may not be compliant with the initial given cell decomposition, so new partitions and cell decompositions might be required.

## IV. Proposed Solution

In this section, a systematic solution to Problem 1 is introduced. Our overall approach builds on abstracting the system in (4) into a set of WTSs and further 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 $\varphi_i, i \in \mathcal{I}$. We take the following steps:

1) Initially, the boundedness of the agents' relative positions is proved, in order to guarantee boundedness of the coupling terms $-\sum_{j \in \mathcal{N}(i)} (x_i - x_j)$. This property, is required for the derivation of the symbolic models. (Sec. IV-A).
2) We utilize decentralized abstraction techniques for the multi-agent system, i.e., discretization of both the workspace and the time such that the motion of each agent is modeled by a WTS $\mathcal{T}_i, \ i \in \mathcal{I}$ (Sec. IV-B).
3) In view of the definition of WTS, the run of each agent is defined such as to be consistent in view of the coupling constraints with the neighbors. The computation of the product of each individual WTS is thus also required (Sec. IV-C).
4) A five-step automated procedure for controller synthesis which serves as a solution to Problem 1 is provided in Sec. IV-D.
5) Finally, the computational complexity of the proposed approach is discussed in Sec. IV-E.

The next sections provide the proposed solution in detail.

### A. Boundedness Analysis

**Theorem 1.** *Consider the multi-agent system* (4). *Assume that the network graph is connected (i.e.* $\lambda_2(\mathcal{G}) > 0$*) and let* $v_i, i \in \mathcal{I}$ *satisfy* $\|v_i(t)\| \leq v_{max}, \ \forall \ i \in \mathcal{I}, t \geq 0$. *Furthermore, let a positive constant* $\bar{R} > K_2 v_{max}$ *where* $K_2 = \frac{2\sqrt{N}(N-1)\|D(\mathcal{G})^\tau\|}{\lambda_2^2(\mathcal{G})} > 0$ *and where* $D(\mathcal{G})$ *is the network adjacency matrix. Then, for each initial condition* $x_i(0) \in \mathbb{R}^n$*, there exists a time* $T > 0$ *such that* $\tilde{x}(t) \in \mathcal{X}, \ \forall t \geq T$*, where* $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\tilde{x}\| \leq \bar{R}\}$ *and* $\tilde{x}(t)$ *was was defined in* (1).

*Proof.* Consider the following candidate Lyapunov function $V : \mathbb{R}^{Nn} \to \mathbb{R}$

$$V(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}(i)} \|x_i - x_j\|^2 = \|\tilde{x}\|^2 > 0. \tag{8}$$

The time derivative of $V$ along the trajectories of (4), can be computed as

$$\begin{aligned} \dot{V}(x) &= [\nabla V(x)]^\tau \ \dot{x} \\ &= \sum_{k=1}^{n} \left\{ \frac{\partial V}{\partial x_1^k} \ \dot{x}_1^k \right\} + \ldots + \sum_{k=1}^{n} \left\{ \frac{\partial V}{\partial x_N^k} \ \dot{x}_N^k \right\} \\ &= \sum_{k=1}^{n} \left\{ c\left(\frac{\partial V}{\partial x}, k\right)^\tau \ c(\dot{x}, k) \right\} \\ &= \sum_{k=1}^{n} \left\{ c\left(\frac{\partial V}{\partial x}, k\right)^\tau \ [-L(\mathcal{G}) \ c(x, k) + c(v, k)] \right\} \end{aligned} \tag{9}$$

where $c(\frac{\partial V}{\partial x}, k) = \begin{bmatrix} \frac{\partial V}{\partial x_1^k} & \cdots & \frac{\partial V}{\partial x_N^k} \end{bmatrix}^\tau$. By computing the partial derivative of the Lyapunov function with respect to vector $x_i, \ i \in \mathcal{I}$ we get $\frac{\partial V}{\partial x_i} = \sum_{j \in \mathcal{N}(i)} (x_i - x_j), \ i \in \mathcal{I}$ from which we have that $c\left(\frac{\partial V}{\partial x}, k\right)^\tau = c(x, k)^\tau \ L(\mathcal{G}), \ k =$

$1, ..., n$. Thus, by substituting the last in (9) we get

$$\begin{aligned}\dot{V}(x) &= \sum_{k=1}^{n} \{c(x,k)^{\tau}\ L(\mathcal{G})\ \left[-L(\mathcal{G})\ c(x,k) + c(v,k)\right]\} \\ &= -\sum_{k=1}^{n} \left\{c(x,k)^{\tau}\ \left[L(\mathcal{G})\right]^{2}\ c(x,k)\right\} + \\ &\qquad \sum_{k=1}^{n} \left\{c(x,k)^{\tau}\ \left[L(\mathcal{G})\right]^{2} c(v,k)\right\} \\ &\le -\sum_{k=1}^{n} \left\{c(x,k)^{\tau}\ \left[L(\mathcal{G})\right]^{2}\ c(x,k))\right\} + \\ &\qquad \left\|\sum_{k=1}^{n} \{c(x,k)^{\tau}\ L(\mathcal{G})\ c(v,k)\}\right\|. \end{aligned} \tag{10}$$

For the first term of (10) we have that

$$\sum_{k=1}^{n} \left\{c(x,k)^{\tau}\ L(\mathcal{G})^{2}\ c(x,k))\right\} = \sum_{k=1}^{n} \left\|L(\mathcal{G})\ c(x,k)\right\|^{2}.$$

For the second term of (10) we have that

$$\begin{aligned}&\left\|\sum_{k=1}^{n} \{c(x,k)^{\tau}\ L(\mathcal{G})\ c(\nu,k)\}\right\| \\ &\quad = \left\|\sum_{k=1}^{n} \{c(x,k)^{\tau}\ D(\mathcal{G})\ D(\mathcal{G})^{\tau}\ c(\nu,k)\}\right\| \\ &\le \sum_{k=1}^{n} \{\|D(\mathcal{G})^{\tau}\ c(x,k)\|\ \|D(\mathcal{G})^{\tau}\|\ \|c(\nu,k)\|\} \\ &\quad = \|D(\mathcal{G})^{\tau}\| \sum_{k=1}^{n} \{\|c(\tilde{x},k)\|\ \|c(\nu,k)\|\}. \end{aligned} \tag{11}$$

By using the Cauchy-Schwarz inequality in (11) we get

$$\begin{aligned}&\left\|\sum_{k=1}^{n} \{c(x,k)^{\tau}\ L(\mathcal{G})\ c(v,k)\}\right\| \\ &\quad \le \|D(\mathcal{G})^{\tau}\| \left(\sum_{k=1}^{n} \|c(\tilde{x},k)\|^{2}\right)^{\frac{1}{2}} \left(\sum_{k=1}^{n} \|c(v,k)\|^{2}\right)^{\frac{1}{2}} \\ &\quad = \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| \|v\| \le \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| \sqrt{N} \|v\|_{\infty}\end{aligned}$$

where $\|v\|_{\infty} = \max\left\{\|v_i\| : i = 1, \ldots, N\right\} \le v_{\max}$. Thus, by combining the previous inequalities, (10) is written

$$\dot{V}(x) \le -\sum_{k=1}^{n} \left\{\|L(\mathcal{G}) c(x,k)\|^{2}\right\} + \sqrt{N}\ \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| v_{\max}. \tag{12}$$

In order to proceed the following Lemma is required.

**Lemma 1.** *Let $x^{\perp}$ be the projection of the vector $x \in \mathbb{R}^{Nn}$ to the orthogonal complement of the subspace $H = \{x \in \mathbb{R}^{Nn} : x_1 = \ldots = x_N\}$. Then, the following hold:*

$$\|L(\mathcal{G})\ c(x,k)\| \ge \lambda_2(\mathcal{G})\ \|c(x^{\perp},k)\|,\ \forall\ k \in \mathcal{I} \tag{13}$$

$$\|x^{\perp}\| \ge \frac{1}{\sqrt{2(N-1)}} \|\tilde{x}\|. \tag{14}$$

*Proof.* See the Appendix of [43]. □

By exploiting Lemma 1, (12) is written

$$\begin{aligned}\dot{V}(x) &\le -\lambda_2^{2}(\mathcal{G}) \sum_{k=1}^{n} \left\{\left\|c(x^{\perp},k)\right\|^{2}\right\} + \sqrt{N}\ \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| v_{\max} \\ &= -\lambda_2^{2}(\mathcal{G})\ \|x^{\perp}\|^{2} + \sqrt{N}\ \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| v_{\max} \\ &\le -\frac{\lambda_2^{2}(\mathcal{G})}{2(N-1)} \|\tilde{x}\|^{2} + \sqrt{N}\ \|D(\mathcal{G})^{\tau}\|\ \|\tilde{x}\| v_{\max} \\ &\le -K_1 \|\tilde{x}\| \left(\|\tilde{x}\| - K_2 v_{\max}\right). \end{aligned} \tag{15}$$

where $K_1 = \frac{\lambda_2^2(\mathcal{G})}{2(N-1)} > 0$. By using the following implication $\tilde{x} = D^{\tau}(\mathcal{G}) x \Rightarrow \|\tilde{x}\| = \|D(\mathcal{G})^{\tau} x\| \le \|D(\mathcal{G})^{\tau}\| \|x\|$, apparently, we have that $0 < V(x) = \|\tilde{x}\|^2 \le \|D(\mathcal{G})^{\tau}\|^2 \|x\|^2$ and $\dot{V}(x) < 0$ when $\|\tilde{x}\| \ge \bar{R} > K_2 v_{\max}$. Thus, there exists a finite time $T > 0$ such that the trajectory will enter the compact set $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\tilde{x}\| \le \bar{R}\}$ and remain there for all $t \ge T$ with $\bar{R} > K_2 v_{\max}$. This can be extracted from the following.

Let us define the compact set $\Omega = \left\{x \in \mathbb{R}^{Nn} : K_2 v_{\max} < \bar{R} \le \|\tilde{x}\| \le \bar{M}\right\}$, where $\bar{M} = V(x(0)) = \|\tilde{x}(0)\|^2$. Without loss of generality it is assumed that it holds $\bar{M} > \bar{R}$. Let us define the compact sets:

$$\begin{aligned}\mathcal{S}_1 &= \left\{x \in \mathbb{R}^{Nn} : \|\tilde{x}\| \le \bar{M}\right\}, \\ \mathcal{S}_2 &= \left\{x \in \mathbb{R}^{Nn} : \|\tilde{x}\| \le K_2 v_{\max}\right\}.\end{aligned}$$

From the equivalences $\forall\ x \in S_1 \Leftrightarrow V(x) = \|\tilde{x}\|^2 \le \bar{M}^2, \forall\ x \in S_2 \Leftrightarrow V(x) = \|x\|^2 \le K_2^2 v_{\max}^2$, we have that the boundaries $\partial S_1, \partial S_2$ of sets $S_1, S_2$ respectively, are two level sets of the Lyapunov function $V$. By taking the above into consideration we have that $\partial S_2 \subset \partial S_1$. Hence, we get from (15) that there exist constant $\gamma > 0$ such that:

$$\dot{V}(x) \le -\gamma < 0, \forall\ x \in \Omega = S_1 \backslash S_2. \tag{16}$$

Consequently, the trajectory has to enter the interior of the set of $S_2$ in finite time $T > 0$ and remain there for all time $t \ge T$. □

It should be noticed that the relative boundedness of the agents' positions guarantees a global bound on the coupling terms $-\sum_{j \in \mathcal{N}(i)} (x_i - x_j)$, as defined in (4). This bound will be later exploited in order to capture the behavior of the system in $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\tilde{x}\| \le \bar{R}\}$, by a discrete state WTS.

### *B. Abstraction*

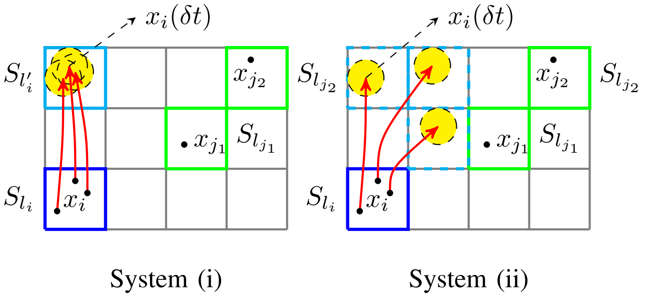
In this section we provide the abstraction technique that is adopted in order to capture the dynamics of each agent into Transition Systems. We work completely in discrete level, which is necessary in order to solve the Problem 1.

Firstly, some additional notation is introduced. Given an index set $\mathbb{I}$ and an agent $i \in \mathcal{I}$ with neighbors $j_1, \ldots, j_{N_i}$, the mappings $\mathrm{pr}_i : \mathbb{I}^N \to \mathbb{I}^{N_i+1}, \bar{\mathrm{pr}}_i : \mathbb{I}^N \to \mathbb{I}$ are defined, where $\mathbb{I}^N = \underbrace{\mathbb{I} \times \ldots \times \mathbb{I}}_{N-products}$. The first one assigns to each $N$-tuple $\mathbf{l} = (l_1, \ldots, l_N) \in \mathbb{I}^N$ the $N_i + 1$ tuple $\mathbf{l}_i = (l_i, l_{j_1}, \ldots, l_{j_{N_i}}) \in \mathbb{I}^{N_i+1}$ which denotes the indices of the cells where the agent $i$ and its neighbors belong. The second one assigns to each

$N$-tuple $l = (l_1, \dots, l_N) \in \mathbb{I}^N$ the position $l_i \in \mathbb{I}$ of the agent $i$, i.e., the cell that the agent $i$ occupies at the moment.

Consider a particular configuration $\bar{S} = \{\bar{S}_l\}_{l \in \mathbb{I}}$, where agent $i$ occupies the cell $\bar{S}_{l_i}$. We denote here with $\bar{S}$ the cell decomposition which is the outcome of the abstraction technique that is adopted for the problem solution that will be presented in this Section. This is not necessarily the same the cell decomposition $S$ from Assumption 2 and Problem 1. Let $\delta t$ be a time step. Through the aforementioned space and time discretization $\bar{S} - \delta t$ we aim to capture the reachability properties of the continuous system (4), in order to create a WTS of each agent. The WTS will later on serve in the synthesis of plans that fulfill the high-level specifications and that map onto the desired free inputs $v_i, i \in \mathcal{I}$.

We proceed by describing the abstraction procedure. If there exists a free input for each state in $\bar{S}_{l_i}$ that navigates the agent $i$ into the cell $\bar{S}_{l'_i}$ precisely in time $\delta t$, regardless of the locations of the agent $i$'s neighbors within their current cells, then a transition from $l_i$ to $l'_i$ is enabled in the WTS. This forms the well-possessedness of transitions which will be explained hereafter. A mathematical derivation can be found in [32].



System (i) System (ii)

Fig. 5: Illustration of a space-time discretization which is well posed for system (i) but non-well posed for system (ii).

We next illustrate the concept of a well-posed abstraction, namely, a discretization which generates for each agent a Transition System in accordance with the discussion above and the Def. 4. Consider a cell decomposition $\bar{S} = \{\bar{S}_l\}_{l \in \mathbb{I} = \{1, \dots, 12\}}$ as depicted in Fig. 5 and a time step $\delta t$. The tails and the tips of the arrows in the figure depict the initial cell and the endpoints of agent's $i$ trajectories at time $\delta t$ respectively. In both cases in the figure we focus on agent $i$ and consider the same cell configuration for $i$ and its neighbors. However, different dynamics are considered for Cases (i) and (ii). In Case (i), it can be observed that for the three distinct initial positions in cell $\bar{S}_{l_i}$, it is possible to drive agent $i$ to cell $\bar{S}_{l'_i}$ at time $\delta t$. We assume that this is possible for all initial conditions in this cell and irrespectively of the initial conditions of $i$'s neighbors in their cells and the inputs they choose. It is also assumed that this property holds for all possible cell configurations of $i$ and for all the agents of the system. Thus we have a well-posed discretization for system (i). On the other hand, for the same cell configuration and system (ii), the following can be observed. For three distinct initial conditions of $i$ the corresponding reachable sets at $\delta t$, which are enclosed in the dashed circles, lie in different cells. Thus, it is not possible given this cell configuration of $i$ to find a cell in the decomposition which is reachable from every point in the initial cell and we conclude that discretization is not well-posed for system (ii).

We present at this point the sufficient conditions that relate the dynamics of the multi-agent system (4), the time step $\delta t$ and the diameter $d_{\max} = \sup\{\|x - y\| : x, y \in \bar{S}_l, l \in \mathbb{I}\}$ of the cell decomposition $\bar{S}$, and guarantee the existence of the aforementioned well-posed transitions from each cell. Based on [32] (Section III, inequality (3), Section IV, inequalities (28, 29)), the sufficient conditions that the dynamics of a general class of system in the form

$$\dot{x}_i = f_i(x_i, \mathbf{x}_j) + v_i, i \in \mathcal{I} \tag{17}$$

where $\mathbf{x}_j = (x_{j_1}, \dots, x_{j_{N_i}}) \in \mathbb{R}^{N_i n}$, should fulfill in order to have well-posed abstractions are the following:

**(C1)** There exists $M > v_{\max} > 0$ such that $\|f_i(x_i, \mathbf{x}_j)\| \leq M$, $\forall i \in \mathcal{I}, \forall \ x \in \mathbb{R}^{Nn} : \mathrm{pr}_i(x) = (x_i, \mathbf{x}_j)$ and $\tilde{x} \in \mathcal{X}$, by applying the projection operator $\mathrm{pr}_i$ for $\mathbb{I} = \mathbb{R}^n$.

**(C2)** There exists a Lipschitz constant $L_1 > 0$ such that

$$\begin{aligned} &\|f_i(x_i, \mathbf{x}_j) - f_i(x_i, \mathbf{y}_j)\| \leq L_1 \|(x_i, \mathbf{x}_j) - (x_i, \mathbf{y}_j)\|, \\ &\qquad \forall \ i \in \mathcal{I}, x_i, y_i \in \mathbb{R}^n, \mathbf{x}_j, \mathbf{y}_j \in \mathbb{R}^{N_i n}. \end{aligned} \tag{18}$$

**(C3)** There exists a Lipschitz constant $L_2 > 0$ such that

$$\begin{aligned} &\|f_i(x_i, \mathbf{x}_j) - f_i(y_i, \mathbf{x}_j)\| \leq L_2 \|(x_i, \mathbf{x}_j) - (y_i, \mathbf{x}_j)\|, \\ &\qquad \forall \ i \in \mathcal{I}, x_i, y_i \in \mathbb{R}^n, \mathbf{x}_j, \mathbf{y}_j \in \mathbb{R}^{N_i n}. \end{aligned} \tag{19}$$

From (4) and (17) we get $f_i(x_i, \mathbf{x}_j) = -\sum_{j \in \mathcal{N}(i)} (x_i - x_j)$. By checking all the conditions one by one for $f_i(x_i, \mathbf{x}_j)$ as in (4), we have:

**(C1)** For every $i \in \mathcal{I}, \forall \ x \in \mathbb{R}^{Nn} : \tilde{x} \in \mathcal{X}$ and $\mathrm{pr}_i(x) = (x_i, \mathbf{x}_j)$ we have that

$$\begin{aligned} \|f_i(x_i, \mathbf{x}_j)\| &= \left\| - \sum_{j \in \mathcal{N}(i)} (x_i - x_j) \right\| \leq \sum_{j \in \mathcal{N}(i)} \|x_i - x_j\| \\ &\leq \sum_{(i,j) \in \mathcal{E}} \|x_i - x_j\| = \Delta x \leq \bar{R}. \end{aligned} \tag{20}$$

Thus, $M = \bar{R}$. We have also that $\|D(\mathcal{G})^\tau\| = \sqrt{\lambda_{\max}(D(\mathcal{G})D(\mathcal{G})^\tau)} = \sqrt{\lambda_{\max}(\mathcal{G})}$ and $\lambda_2(\mathcal{G}) \leq \frac{N}{N-1} \min\{N_i : i \in \mathcal{I}\}$ from [44]. For $N > 2$ it holds that $\lambda_2(\mathcal{G}) < N$. From Theorem (1) we have that $\bar{R} > K_2 v_{\max} \Leftrightarrow M > K_2 v_{\max}$. It holds that $M > v_{\max}$ since

$$\begin{aligned} K_2 &= \frac{2\sqrt{N}(N-1)\,\|D(\mathcal{G})^\tau\|}{\lambda_2^2(\mathcal{G})} = \frac{2\sqrt{N}(N-1)\sqrt{\lambda_{\max}(\mathcal{G})}}{\sqrt{\lambda_2^3(\mathcal{G})}\sqrt{\lambda_2(\mathcal{G})}} \\ &\geq \frac{2\sqrt{N}(N-1)}{\sqrt{N^3}} \sqrt{\frac{\lambda_{\max}(\mathcal{G})}{\lambda_2(\mathcal{G})}} \geq \frac{2\sqrt{N}(N-1)}{\sqrt{N^3}} > 1. \end{aligned}$$

**(C2)** We have that

$$\begin{aligned}
&\|f_i(x_i,\mathbf{x}_j) - f_i(x_i,\mathbf{y}_j)\| \\
&\quad = \Big\| - \sum_{j\in\mathcal{N}(i)} (x_i - x_j) + \sum_{j\in\mathcal{N}(i)} (x_i - y_j)\Big\| \\
&\quad \le \sqrt{N_i}\ \|(x_i,\mathbf{x}_j) - (x_i,\mathbf{y}_j)\| \\
&\quad \le \max\{\sqrt{N_i} : i = 1,\dots,N\}\ \|(x_i,\mathbf{x}_j) - (x_i,\mathbf{y}_j)\|.
\end{aligned}$$

Thus, the condition **(C2)** holds and the Lipschitz constant is $L_1 = \max\{\sqrt{N_i} : i = 1,\dots,N\} > 0$, where the inequality $\left(\sum_{i=1}^{\rho}\alpha_i\right)^2 \le \rho\ \left(\sum_{i=1}^{\rho}\alpha_i^2\right)$ is used.

**(C3)** By using the same methodology with the proof of **(C2)** we conclude that $L_2 = \max\{N_i : i = 1,\dots,N\} > 0$.

Based on the sufficient condition for well posed abstractions in [32], the acceptable values of $d_{\max}$ and $\delta t$ are given as

$$d_{\max} \in \left(0, \frac{(1-\lambda)^2 v_{\max}^2}{4ML}\right] \tag{21}$$

$$\delta t \in \left[\frac{(1-\lambda)v_{\max} - \sqrt{(1-\lambda)^2 v_{\max}^2 - 4MLd_{\max}}}{2ML},\right. \\ \left.\frac{(1-\lambda)v_{\max} + \sqrt{(1-\lambda)^2 v_{\max}^2 - 4MLd_{\max}}}{2ML}\right] \tag{22}$$

where the parameter $\lambda$ stands for reachability purposes and $L = \max\{3L_2 + 4L_1\sqrt{N_i}, i \in \mathcal{I}\}$ with the dynamics bound $M$ and the Lipschitz constants $L_1, L_2$ as previously deduced.

**Remark 4.** Notice that when $d_{\max}$ in (21) is chosen sufficiently small, it is also possible due to the lower bound on the acceptable $\delta t$ in (22) to select a correspondingly small value of the sampling time and capture with higher accuracy the properties of the continuous trajectories. However, this will result in a finer discretization and increase the complexity of the symbolic models.

**Remark 5.** Assume that a cell-decomposition of diameter $d_{\text{max}}$ and a time step $\delta t$ which guarantee well-posed transitions, namely, which satisfy (21) and (22), have been chosen. It is also possible to chose any other cell-decomposition with diameter $\hat{d}_{\text{max}} \le d_{\text{max}}$ since, by (22), the range of acceptable $\delta t$ increases.

We showed that the dynamics of the system (4) satisfy all the sufficient conditions (C1)-(C3), thus we have a well-posed space-time discretization $\bar{S} - \delta t$. Recall now Assumption 2. It remains to establish the compliance of the cell decomposition $S = \{S_\ell\}_{\ell\in\mathbb{I}}$, which is given in the statement of Problem 1, with the cell decomposition $\bar{S} = \{\bar{S}_l\}_{l\in\bar{\mathbb{I}}}$, which is the outcome of the abstraction. By the term of compliance, we mean that:

$$\bar{S}_l \cap S_\ell \in S \cup \{\emptyset\},\ \text{for each } \bar{S}_l \in \bar{S}, S_\ell \in S \text{ and } l \in \bar{\mathbb{I}}, \ell \in \mathbb{I}. \tag{23}$$

In order to address this problem, we define:

$$\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}} = \{\bar{S}_l \cap S_\ell :\ l \in \bar{\mathbb{I}}, \ell \in \mathbb{I}\}\backslash\{\emptyset\} \tag{24}$$

which forms a cell decomposition which is compliant with the cell decomposition $S$ from Problem 1 and serves as the abstraction solution of this problem. This can be deducted as follows: By taking all the combinations of intersections $\bar{S}_l \cap S_\ell, \forall\ l \in \bar{\mathbb{I}}, \forall\ \ell \in \mathbb{I}$ and enumerating them by indexes of the set $\hat{\mathbb{I}}$, the cells $\{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}}$ are constructed for which the following holds: $\forall\ \ell \in \mathbb{I}, \exists\ l \in \bar{\mathbb{I}}$ such that $\hat{S}_{\hat{l}} = \bar{S}_l \cap S_\ell = \emptyset$ and $\text{int}(\hat{S}_{\hat{\ell}}) \cap \text{int}(\hat{S}_{\hat{\ell}'}) \neq \emptyset$ for all $\hat{\ell}' \in \hat{\mathbb{I}}\backslash\{\hat{\ell}\}$. After all the intersections we have $\cup_{l\in\hat{\mathbb{I}}}\hat{S}_{\hat{l}} = \mathcal{X}$. The diameter of the cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}}$ is defined as $\hat{d}_{\text{max}} = \sup\{\|x-y\| : x, y \in \hat{S}_{\hat{l}}, \hat{l} \in \hat{\mathbb{I}}\} \le d_{\text{max}}$. Hence, according to the discussion above, we have a well-posed abstraction. See Example 5 for an illustration of these deviations.

For the solution to Problem 1, the WTS which corresponds to the cell configuration $\hat{S}$, the diameter $\hat{d}_{\text{max}}$ and the time step $\delta t$ will be exploited. Thus, the WTS of each agent is defined as follows:

**Definition 8.** The motion of each agent $i \in \mathcal{I}$ in the workspace is modeled by a WTS $\mathcal{T}_i = (S_i, S_i^{\text{init}}, Act_i, \longrightarrow_i, d_i, AP_i, \hat{L}_i)$ where

- $S_i = \hat{\mathbb{I}}$, the set of states of each agent is the set of indices of the cell decomposition.
- $S_0^{\text{init}} \subseteq S_i$, is a set of initial states.
- $Act_i = \hat{\mathbb{I}}^{N_i+1}$, the set of actions representing where agent $i$ and its neighbors are located.
- For a pair $(l_i, \mathbf{l_i}, l_i')$ we have that $(l_i, \mathbf{l_i}, l_i') \in \longrightarrow_i$ iff $l_i \xrightarrow{\mathbf{l_i}}_i l_i'$ is well-posed for each $l_i, l_i' \in S_i$ and $\mathbf{l_i} = (l_i, l_{j_1}, \dots, l_{j_{N_i}}) \in Act_i$.
- $d_i : \longrightarrow_i \rightarrow \mathbb{T}$, is a map that assigns a positive weight (duration) to each transition. The duration of each transition is exactly equal to $\delta t > 0$.
- $AP_i = \Sigma_i$, is the set of atomic propositions which are inherent properties of the workspace.
- $\hat{L}_i : S_i \rightarrow 2^{AP_i}$, is the labeling function that maps the every state $s \in S_i$ into the services that can be provided in this state.

The aforementioned Definition is crucial since from the dynamical system (4) we created through the abstraction procedure individual WTSs of each agent $i \in \mathcal{I}$ which capture the motion of each agent and let us work completely in discrete level in order to design the controllers that satisfy the Problem 1.

Every WTS $\mathcal{T}_i, i \in \mathcal{I}$ generates timed runs and timed words of the form $r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2))\dots, w_i^t = (L_i(r_i(0)), \tau_i(0))(L_i(r_i(1)), \tau_i(1))(L_i(r_i(2)), \tau_i(2))\dots$ respectively, over the set $2^{AP_i}$ according to Def. 5 with $\tau_i(j) = j \cdot \delta t, \forall\ j \ge 0$. It is necessary now to provide the relation between the time words that are generated by the WTSs $\mathcal{T}_i, i \in \mathcal{I}$ with the time service words produced by the trajectories $x_i(t), i \in \mathcal{I}, t \ge 0$.

**Remark 6.** By construction, each time word produced by the WTS $\mathcal{T}_i$ is a service time word associated with the trajectory $x_i(t)$ of the system (4). Hence, if we find a timed word of $\mathcal{T}_i$ satisfying a formula $\varphi_i$ given in MITL, we also found for each agent $i$ a desired timed word of the original system, and hence trajectories $x_i(t)$ that are solution to the Problem (1). (i.e., the produced timed words of $\mathcal{T}_i$ are compliant with the service time words of the trajectories $x_i(t)$.)

**Example 5.** Assume that $S = \{S_\ell\}_{\ell \in \{1,\ldots,6\}}$ as given in Example 1 depicted in Fig. 5 by red rectangles, is the cell decomposition of Problem 1. Let also $\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{I}} = \{1,\ldots,6\}}$ depicted in Fig. 7 with light blue cells, be a cell decomposition which serves as potential solution of this Problem satisfying all the abstraction properties that have been mentioned in this Section. It can be observed that the two cell decompositions are not compliant according to (23). However, by using (24), a new cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l} \in \hat{\mathbb{I}} = \{1,\ldots,15\}}$ (depicted in Fig. 7), that is compliant $S$, can be obtained and forms the final cell decomposition solution. Let also $d_{\max}, \hat{d}_{\max}$ be the diameters of the cell decompositions $S, \hat{S}$ respectively. It holds that $\hat{d}_{\max} \leq d_{\max}$ which is in accordance with the Remark 5.

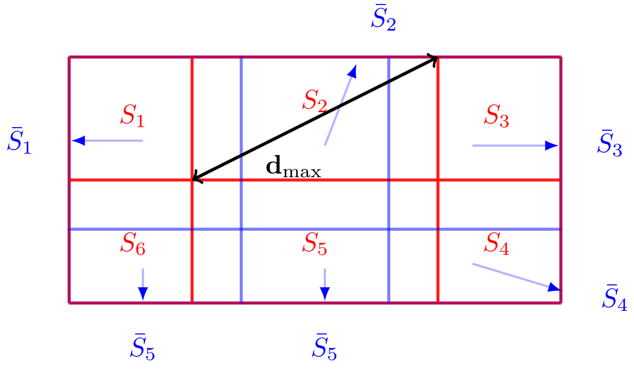


Fig. 6: An example with a given cell decomposition $S = \{S_l\}_{l \in \{1,\ldots,6\}}$ and a non-compliant solution $\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{I}} = \{1,\ldots,6\}}$.

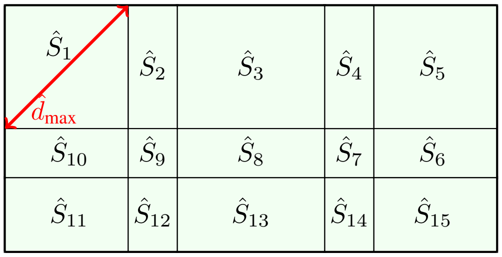


Fig. 7: The resulting compliant cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l} \in \hat{\mathbb{I}} = \{1,\ldots,15\}}$ of Example 5.

### C. Runs Consistency

Due to the fact that the dynamics of the system have couplings between the agents, it is necessary to define timed runs that can be performed from each individual agent. Even though we have the individual WTS of each agent, the runs that the later generates may not be performed by an agent due to the constrained motion that is imposed by the coupling terms. Hence, we need to provide a tool that synchronizes the agents at each time step $\delta t$ and is able to determine which of the generated runs of the individual WTS can be performed by the agent (i.e., they are consistent runs). In order to address the aforementioned issue, we provide a centralized product WTS which captures the behavior of the coupled multi-agent system as a team, and the generated product run (see Def. 10) can later be projected onto consistent individual runs. The following two definitions deal with the product WTS and consistent runs respectively.

**Definition 9.** The product WTS $\mathcal{T}_p = (S_p, S_p^{\text{init}}, \longrightarrow_p)$ is defined as follows:

- $S_p = \hat{\mathbb{I}}^N$.
- $(s_1, \ldots, s_N) \in S^{\text{init}}$ if $s_i \in S_i^{\text{init}}$ for all $i \in \mathcal{I}$.
- $(\mathbf{l}, \mathbf{l}') \in \longrightarrow_{\mathbf{p}}$ iff $l_i' \in \text{Post}_i(l_i, \text{pr}_i(\mathbf{l})), \forall\ \mathbf{i} \in \mathcal{I}, \forall\ \mathbf{l} = (\mathbf{l_1}, \ldots, \mathbf{l_N}), \mathbf{l}' = (\mathbf{l}'_{\mathbf{1}}, \ldots, \mathbf{l}'_{\mathbf{N}})$.
- $d_p : \longrightarrow_p \to \mathbb{T}$ : As in the individual WTS's case, the transition weight is $d_p(\cdot) = \delta t$.

The action labels, the atomic propositions and the labeling function in $\mathcal{T}_p$ are insignificant. Hence, without loss of generality, there were omitted from the tuple.

**Definition 10.** Given the timed run

$$r_p^t = ((r_p^1(0), \ldots, r_p^N(0)), \tau_p(0))((r_p^1(1), \ldots, r_p^N(1)), \tau_p(1)) \ldots$$

of the WTS $\mathcal{T}_p$, the induced set of projected runs

$$\{r_i^t = (r_p^i(0), \tau_p(0))(r_p^i(1), \tau_p(1)) \ldots : i \in \mathcal{I}\}$$

of the WTSs $\mathcal{T}_1, \ldots, \mathcal{T}_N$, respectively will be called *consistent runs*. Since the duration of each agent's transition is $\delta t$ it holds that $\tau_p(j) = j \cdot \delta t, j \geq 0$.

Therefore, through the product WTS $\mathcal{T}_p$, we can always generate individual consistent runs for each agent. It remains to provide a systematic approach of how to determine consistent runs $\widetilde{r}_1, \ldots, \widetilde{r}_N$ which are associated with the corresponding service time words $\widetilde{w}_1^t, \ldots, \widetilde{w}_N^t$ (note that with tilde we denote the outcome of our solution approach) and, according to the Lemma 6, their corresponding compliant trajectories $x_1(t), \ldots, x_N(t)$ will satisfy the corresponding MITL formulas $\varphi_1, \ldots, \varphi_N$. and they are the solution to Problem 1.

**Example 6.** An example that explains the notation that has been introduced until now is the following: Consider an agent (Fig. 8) moving in the workspace with $\mathcal{N}(i) = \{1, 2\}$, $S = \{S_\ell\}_{\ell \in \mathbb{I} = \{1,\ldots,6\}}$ is the given cell decomposition from Problem 1, $\bar{S} = \{\bar{S}_l\}_{l \in \mathbb{I} = \{1,\ldots,28\}}$ is the cell decomposition that is the outcome of the abstraction and atomic propositions $\{p_1, \ldots, p_6\} = \{\text{orange}, green, blue\ yellow, red, grey\}$. The red arrows represent both the transitions of the agent $i$ and its neighbors. The dashed lines indicate the edges in the network graph. For the atomic propositions we have that $L_i(14) = \{p_1\}, L_i(17) = \{p_5\}, L_i(10) = \{p_2\}, L_i(20) = \{p_4\}, L_{j_1}(28) = \{p_6\} = L_{j_1}(27), L_{j_1}(24) = \{p_5\}, L_{j_1}(22) = \{p_4\}, L_{j_2}(2) = \{p_1\}, L_{j_2}(12) = \{p_2\} = L_{j_2}(5), L_{j_2}(9) = \{p_3\}$. Note also the diameter of the cells $\hat{d}_{\max} = d_{\max}$. For the cell configurations we have:

$$Init: \begin{cases} \mathbf{l_i} = (14, 28, 2) \\ \mathbf{l_{j_1}} = (28, 14) \\ \mathbf{l_{j_2}} = (2, 14) \end{cases} \quad Step1: \begin{cases} \mathbf{l_i} = (17, 27, 13) \\ \mathbf{l_{j_1}} = (27, 17) \\ \mathbf{l_{j_2}} = (13, 17) \end{cases}$$

$$Step2: \begin{cases} \mathbf{l_i} = (10, 24, 5) \\ \mathbf{l_{j_1}} = (24, 10) \\ \mathbf{l_{j_2}} = (5, 10) \end{cases} \quad Step3: \begin{cases} \mathbf{l_i} = (20, 22, 9) \\ \mathbf{l_{j_1}} = (22, 20) \\ \mathbf{l_{j_2}} = (9, 20) \end{cases}$$

which are actions to the corresponding transitions. The consistent timed runs are given as

$$\begin{aligned} r_i^t &= (r_i(0) = 14, \tau_i(0) = 0)(r_i(1) = 17, \tau_i(1) = \delta t) \\ &(r_i(2) = 10, \tau_i(2) = 2\delta t)(r_i(3) = 20, \tau_i(3) = 3\delta t) \\ r_{j_1}^t &= (r_{j_1}(0) = 28, \tau_{j_1}(0) = 0)(r_{j_1}(1) = 27, \tau_{j_1}(1) = \delta t) \\ &(r_{j_1}(2) = 24, \tau_{j_1}(2) = 2\delta t)(r_{j_1}(3) = 22, \tau_{j_1}(3) = 3\delta t) \\ r_{j_2}^t &= (r_{j_2}(0) = 2, \tau_{j_2}(0) = 0)(r_{j_2}(1) = 13, \tau_{j_2}(1) = \delta t) \\ &(r_{j_2}(2) = 5, \tau_{j_2}(2) = 2\delta t)(r_{j_2}(3) = 9, \tau_{j_2}(3) = 3\delta t). \end{aligned}$$

It can be observed that $r_i^t \models (\varphi_i = \Diamond_{[0,6]}\{yellow\})$ if $3\delta t \in [0, 6]$, $r_{j_1}^t \models (\varphi_{j1} = \Diamond_{[3,10]}\{red\})$ if $2\delta t \in [3, 10]$ and $r_{j_2}^t \models (\varphi_{j2} = \Diamond_{[3,9]}\{blue\})$ if $3\delta t \in [3, 9]$. For $\delta t = 1$, all the agents satisfy their goals.

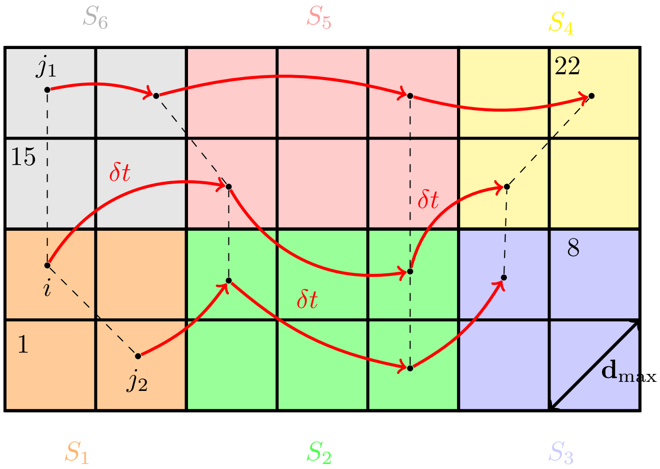


Fig. 8: Timed runs of the agents $i, j_1, j_2$

**Remark 7.** We chose to utilize decentralized abstractions, to generate the individual WTSs $\mathcal{I}, i \in \mathcal{I}$ for each agent and to compute the synchronized-centralized product WTS $\mathcal{T}_p$ for the following reasons:

1) The state space of the centralized system to be abstracted is $\mathcal{X}^N \subseteq \mathbb{R}^{Nn}$, which is i) harder to visualize and handle ii) not naturally related to the individual specifications. Thus, it is more natural to define the specifications through the individual transition system of each agent corresponding to a discretization of $\mathcal{X}$ and then form the product to obtain the possible consistent satisfying plans.
2) Additionally, many centralized abstraction frameworks are based on approximations of the system's reachable set from a given cell over the transition time interval. These require in the general nonlinear case global dynamics properties and may avoid taking into account the finer dynamics properties of the individual entities, which can lead to more conservative estimates for large scale systems.

### D. Controller Synthesis

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

1) $N$ TBAs $\mathcal{A}_i,\ i \in \mathcal{I}$ that accept all the timed runs satisfying the corresponding specification formulas $\varphi_i, i \in \mathcal{I}$ are constructed.
2) A Büchi WTS $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i$ (see Def. 11 below) for every $i \in \mathcal{I}$ is constructed. The accepting runs of $\widetilde{\mathcal{T}}_i$ are the individual runs of the $\mathcal{T}_i$ that satisfy the corresponding MITL formula $\varphi_i,\ i \in \mathcal{I}$.
3) We pick a set of accepting runs $\{\widetilde{r}_1^t, \dots, \widetilde{r}_N^t\}$ from Step 2. We check if they are consistent according to Def. 10. If this is true then we proceed with Step 5. If this is not true then we repeat Step 3 with a different set of accepting runs. At worst case, the number of repetitions that should be performed is finite; if a consistent set of accepting runs is not found, we proceed with the less efficient centralized, yet complete, procedure in Step 4.
4) We create the product $\widetilde{\mathcal{T}}_p = \mathcal{T}_p \otimes \mathcal{A}_p$ where $\mathcal{A}_p$ is the TBA that accepts all the words that satisfy the formula $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_N$. An accepting run $\widetilde{r}_p$ of the product is projected into the accepting runs $\{\widetilde{r}_1, \dots, \widetilde{r}_N\}$. If there is no accepting run found in $\mathcal{T}_p \otimes \mathcal{A}_p$, then Problem 1 has no solution.
5) 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 (4) via the corresponding sequence of feedback laws.

In order to construct the Buchi WTSs $\widetilde{\mathcal{T}}_p$ and $\widetilde{\mathcal{T}}_i, i \in \mathcal{I}$ that were presented in Steps 2 and 4, we consider the following definition:

**Definition 11.** Given a WTS $\mathcal{T}_i = (S_i, S_i^{\text{init}}, Act_i, \longrightarrow_i, d_i, AP_i, \hat{L}_i)$, and a TBA $\mathcal{A}_i = (Q_i, Q_i^{\text{init}}, C_i, Inv_i, E_i, F_i, AP_i, \mathcal{L}_i)$ with $|C_i|$ clocks and let $C_i^{max}$ be the largest constant appearing in $\mathcal{A}_i$. Then, we define their *Büchi WTS* $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i = (\widetilde{S}_i, \widetilde{S}_i^{init}, \widetilde{Act}_i, \rightsquigarrow_i, \widetilde{d}_i, \widetilde{F}_i, AP_i, \widetilde{L}_i)$ as follows:

- $\widetilde{S}_i \subseteq \{(s_i, q_i) \in S_i \times Q_i : \hat{L}_i(s_i) = \mathcal{L}_i(q_i)\} \times \mathbb{T}_\infty^{|C_i|}$.
- $\widetilde{S}_i^{init} = S_i^{init} \times Q_i^{init} \times \underbrace{\{0\} \times \ldots \times \{0\}}_{|C_i|\ products}$.
- $\widetilde{Act}_i = Act_i$.
- $(\widetilde{q}, \mathbf{I_i}, \widetilde{q}') \in \rightsquigarrow_i$ iff
  - ○ $\widetilde{q} = (s, q, \nu_1, \dots, \nu_{|C_i|}) \in \widetilde{S}_i$, $\widetilde{q}' = (s', q', \nu_1', \dots, \nu_{|C_i|}') \in \widetilde{S}_i$,
  - ○ $\mathbf{I_i} \in Act_i$,
  - ○ $(s, \mathbf{I_i}, s') \in \longrightarrow_i$, and
  - ○ there exists $\gamma, R$, 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 $i \in \{1,\dots,|C_i|\}$

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

Then, $\widetilde{d}_i(\widetilde{q},\widetilde{q}') = d_i(s,s')$.

- $\widetilde{F}_i = \{(s_i, q_i, \nu_1, \dots, \nu_{|C_i|}) \in Q_i : q_i \in F_i\}$.
- $\widetilde{L}_i(s_i, q_i, \nu_1, \dots, \nu_{|C_i|}) = \hat{L}_i(s_i)$.

The Buchi WTS $\widetilde{\mathcal{T}}_p$ is constructed in a similar way. Each Büchi WTS $\widetilde{\mathcal{T}}_i, i \in \mathcal{I}$ 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. 5. It is *accepting* if $q_i(i) \in \widetilde{F}_i$ for infinitely many $i \ge 0$. An accepting timed run of $\widetilde{\mathcal{T}}_i$ projects onto a timed run of $\mathcal{T}_i$ that satisfies the local specification formula $\varphi_i$ 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., [45]), holds:

**Lemma 2.** *Consider an accepting timed run $\widetilde{r}_i^t = (q_k(0),\tau_i(0))(q_i(1),\tau_i(1))\dots$ of the Büchi WTS $\widetilde{\mathcal{T}}_k$ defined above, where $q_i(k) = (r_i(k), s_i(k), \nu_{i,1}, \dots, \nu_{i,M_i})$ denotes a state of $\widetilde{\mathcal{T}}_i$, for all $k \ge 1$. 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 $\rho_i = s_i(0)s_i(1)\dots$. Vice versa, if there exists a timed run $r_k^t = (r_k(0),\tau_k(0))(r_k(1),\tau_k(1))\dots$ of the WTS $\mathcal{T}_k$ that produces a timed word $w(r_k^t) = (L_k(r_k(0)),\tau_i(0))(L_i(r_i(1)),\tau_i(1))\dots$ accepted by the TBA $\mathcal{A}_i$ via its run $\rho_i = s_i(0)s_i(1)\dots$ 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 $\widetilde{\mathcal{T}}_i$, where $q_i(i)$ denotes $(r_i(k), s_i(k), \nu_{i,1}(i), \dots, \nu_{i,M_i}(k))$ in $\widetilde{\mathcal{T}}_i$.*

**Proposition 1.** By following the procedure described in Sec. IV-D a sequence of controllers $v_1, \dots, v_N$ can be designed (if there is a solution according to Steps 1-5) that guarantees the satisfaction of the formulas $\varphi_1, \dots, \varphi_N$ of the agents $1, \dots, N$ respectively, governed by dynamics as in (4).

### *E. Complexity*

Our proposed framework can handle all the expressivity of the MITL formulas according to the semantics of Definition 6. Denote by $|\varphi|$ the length of an MITL formula $\varphi$. A TBA $\mathcal{A}_i, i \in \mathcal{I}$ can be constructed in space and time $2^{\mathcal{O}(|\varphi_i)|}, i \in \mathcal{I}$. So by denoting with $\varphi_{\text{max}} = \max\{|\varphi_i\}, i \in \mathcal{I}$ the MITL formula with the longest length we have that the complexity of Step 1 is $2^{\mathcal{O}(|\varphi_{\text{max}})|}$. The model checking of Step 2 costs $\mathcal{O}(|\mathcal{T}_i| \cdot 2^{|\varphi_i|}), i \in \mathcal{I}$ where $|\mathcal{T}_i|$ is the length of the WTS $\mathcal{T}_i$ i.e., the number of its states. Thus, $\mathcal{O}(|\mathcal{T}_i| \cdot 2^{|\varphi_i|}) = \mathcal{O}(|S_i| \cdot 2^{|\varphi_i|}) = \mathcal{O}(|\hat{\mathbb{I}}| \cdot 2^{|\varphi_i|})$. The worst case of Step 2 costs $\mathcal{O}(|\mathcal{T}_{\text{max}}| \cdot 2^{|\varphi_{\text{max}}|})$ where $|\mathcal{T}_{\text{max}}|$ is the number of the states of the WTS which corresponds to the longest formula $\varphi_{\text{max}}$. Due to the fact that all the WTSs in Step 2 have the same number of states, it holds that the worst case complexity of Step 2 costs $\mathcal{O}(|\hat{\mathbb{I}}| \cdot 2^{|\varphi_{\text{max}}|})$. By denoting with $R_{iter}$ the finite number of repetitions of Step 3, we have the best case complexity as $\mathcal{O}(R_{\text{iter}} \cdot |\hat{\mathbb{I}}| \cdot 2^{|\varphi_{\text{max}}|})$, since the Step 3 is more efficient than Step 4. The worst case complexity of our proposed framework is when Step 4 is followed, which is $\mathcal{O}(|\hat{\mathbb{I}}|^N \cdot 2^{|\varphi_{\text{max}}|})$ where $|\hat{\mathbb{I}}|$ is the number of cells of the cell decomposition $\hat{S}$.

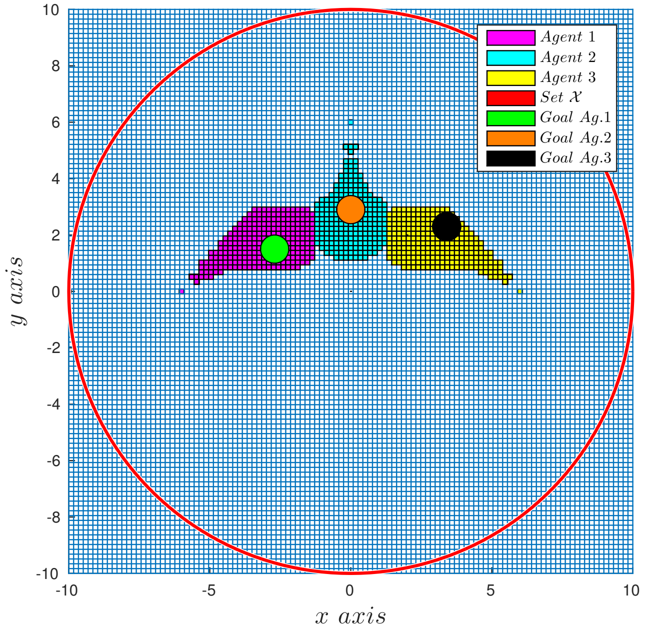


Fig. 9: Space discretization, goal regions and reachable sets for each agent in a time horizon of $11\delta t$ steps

## V. Simulation Results

For a simulation example, a system of three agents with $x_i \in \mathbb{R}^2,\ i \in \mathcal{I} = \{1,2,3\}, \mathcal{E} = \{\{1,2\},\{2,3\}\}, \mathcal{N}(1) = \{2\} = \mathcal{N}(3), \mathcal{N}(2) = \{1,3\}$ is considered. Their dynamics are given as $\dot{x}_1 = x_2 - x_1 + v_1, \dot{x}_2 = x_1 + x_3 - 2x_2 + v_2$ and $\dot{x}_3 = x_2 - x_3 + v_3$. The simulation parameters are set to $\bar{R} = 10, M = 20, v_{\text{max}} = 10, L_1 = \sqrt{2}, L_2 = 2, \delta t = 0.2$. The time step $\delta t$ is chosen during the abstraction process according to the formulas (21), (22) and it is not chosen with reference to satisfaction of the MITL formulas. The workspace $[-10,10] \times [-10,10] \subseteq \mathbb{R}^2$ is partitioned into cells and the initial agents' positions are set to $(-6,0), (0,6)$ and $(6,0)$ respectively. The specification formulas are set to $\varphi_1 = \Diamond_{[0.5,1.7]}\{\text{green}\}, \varphi_2 = \Diamond_{[1.0,1.4]}\{\text{orange}\}, \varphi_3 = \Diamond_{[0.7,1.8]}\{\text{black}\}$ respectively and their corresponding TBAs are given in Fig. 3. The abstraction presented in this paper, the reachable cells of each agent as well as the goal regions are depicted in Fig. 9. It can be observed that not all the individual runs satisfy the desired specification. By applying the five-step controller synthesis procedure that was presented in Sec. IV, the individual run of each agent satisfy the formulas $\varphi_1, \varphi_2$ and $\varphi_3$ in $6\delta t$, $6\delta t$ and $5\delta t$ respectively. The simulation is performed in a horizon of $11\delta t$ steps (as the steps that explained in the Example 6). The product WTS has $45 \times 10^4$ states. The simulations were carried out in MATLAB Environment on a desktop with 8 cores, 3.60GHz CPU and 16GB of RAM.

## VI. Conclusions and Future Work

A systematic method of both abstraction and controller synthesis of dynamically coupled multi-agent path-planning has been proposed, in which timed constraints of fulfilling a high-level specification are imposed to the system. The solution involves initially a boundedness analysis and secondly the abstraction of each agent's motion into WTSs and automata construction. The simulation example demonstrates our solution approach. Future work includes further computational improvement of the abstraction method and more complicated high-level tasks being imposed to the agents in order to exploit the expressiveness of MITL formulas.

## References

[1] W. Ren and R. Beard, "Consensus Seeking in Multi-Agent Systems Under Dynamically Changing Interaction Topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[2] R. Olfati-Saber and R. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[3] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[4] G. Shi and K. Johansson, "Robust Consensus for Continuous-Time Multi-Agent Dynamics," *SIAM Journal on Control and Optimization*, vol. 51, no. 5, pp. 3673–3691, 2013.

[5] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in Fixed and Switching Networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[6] M. Egerstedt and X. Hu, "Formation Constrained Multi-Agent Control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.

[7] M. Zavlanos and G. Pappas, "Distributed Connectivity Control of Mobile Networks," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416–1428, 2008.

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

[9] 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.

[10] 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.

[11] Y. Chen, Ding, X. Chu, A. Stefanescu, and C. Belta, "Formal Approach to the Deployment of Distributed Robotic Teams," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 158–171, 2012.

[12] 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 2011)*, pp. 4867–4872, 2011.

[13] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Temporal-Logic-Based Reactive Mission and Motion Planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[14] A. Ulusoy, S. Smith, X. Ding, C. Belta, and D. Rus, "Optimality and Robustness in Multi-Robot Path Planning with Temporal Logic Constraints," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 889–911, 2013.

[15] M. Quottrup, T. Bak, and R. Zamanabadi, "Multi-Robot Planning: A Timed Automata Approach," *IEEE International Conference on Robotics and Automation (ICRA 2004)*, vol. 5, pp. 4417–4422, 2004.

[16] A. Ulusoy, S. Smith, X. Ding, C. Belta, and D. Rus, "Optimal Multi - Robot Path Planning with Temporal Logic Constraints," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pp. 3087–3092, 2011.

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

[18] 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 2015)*, pp. 239–248, 2015.

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

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

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

[22] 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.

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

[24] 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.

[25] 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.

[26] E. A. Gol and C. Belta, "Time-Constrained Temporal Logic Control of Multi-Affine Systems," *Nonlinear Analysis: Hybrid Systems*, 2013.

[27] 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.

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

[29] M. Rungger and M. Zamani, "Compositional Construction of Approximate Abstractions," *18th International Conference on Hybrid Systems: Computation and Control (HSCC 2015)*, pp. 68–77, 2015.

[30] E. Dallal and P. Tabuada, "On Compositional Symbolic Controller Synthesis Inspired by Small-Gain Theorems," *54th IEEE Conference on Decision and Control (CDC)*, pp. 6133–6138, 2015.

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

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

[33] M. Mesbahi and M. Egerstedt, "Graph Theoretic Methods in Multiagent Networks," 2010.

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

[35] 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.

[36] J. Ouaknine and J. Worrell, "On the Decidability of Metric Temporal Logic," *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pp. 188–197, 2005.

[37] 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.

[38] M. Reynolds, "Metric Temporal Logics and Deterministic Timed Automata," 2010.

[39] P. Bouyer, "From Qualitative to Quantitative Analysis of Timed Systems," *Mémoire dhabilitation, Université Paris*, vol. 7, pp. 135–175, 2009.

[40] S. Tripakis, "Checking Timed Buchi Automata Emptiness on Simulation Graphs," *ACM Transactions on Computational Logic (TOCL)*, vol. 10, no. 3, 2009.

[41] 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.

[42] D. Ničković and N. Piterman, "From MTL to Deterministic Timed Automata," *Formal Modeling and Analysis of Timed Systems*, 2010.

[43] D. Boskos and D. Dimarogonas, "Robust Connectivity Analysis for Multi-Agent Systems," *54th IEEE Conference on Decision and Control (CDC 2015)*, pp. 6767–6772, 2015.

[44] M. M. Fiedler, “Algebraic Connectivity of Graphs,” *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[45] C. Baier, J. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.