On the Timed Temporal Logic Planning of Coupled Multi-Agent Systems

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

KTH Center of Autonomous Systems and ACCESS Linnaeus Center, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden.

Abstract

This paper presents a fully automated procedure for controller synthesis for 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). First, a decentralized abstraction that provides a space and time discretization of the multi-agent system is designed. Second, by utilizing this abstraction and techniques from formal verification, we propose an algorithm that computes the individual runs which provably satisfy the high-level tasks. The overall approach is demonstrated in a simulation example conducted in MATLAB environment.

Key words: Multi-Agent Systems, Cooperative Control, Hybrid Systems, Formal Verification, Timed Logics, Abstractions, Discrete Event Systems.

1 Introduction

Over the last few years, the field of control of multi-agent systems under high-level 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]). There is a rich body of literature containing algorithms for verification and synthesis of multi-agent systems under high level specifications ([2–4]). Controller synthesis under timed specifications has been considered in [5–8]. In [5], the authors addressed the problem of designing high-level planners to achieve tasks for switching dynamical systems under Metric Temporal Logic (MTL) specifications and in [6], the authors utilized a counterexample-guided synthesis for cyber-physical systems subject to Signal Temporal Logic (STL) specifications. In [7], an optimal control problem for continuous-time stochastic systems subject to objectives specified in MITL was studied. In [8], the authors focused on motion planning based on the construction of an efficient timed automaton from a given MITL specification. However, all these works are restricted to single agent planning and are not extendable to multi-agent systems in a straightforward way. High-level coordination of multiple vehicles under timed specifications has been considered in [9], by solving an optimization problem over the tasks' execution time instances.

An automata-based solution for multi-agent systems was proposed in our previous work [10], 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 fulfills a global task. 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 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, we call it a free input. A decentralized abstraction procedure is provided, which leads to an in-

⋆ This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the EU H2020 Research and Innovation Programme under GA No. 731869 (Co4Robots), the SSF COIN project, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation.

Email addresses: anikou@kth.se (Alexandros Nikou), boskos@kth.se (Dimitris Boskos), tumova@kth.se (Jana Tumova), dimos@kth.se. (Dimos V. Dimarogonas).

dividual Transition System (TS) for each agent and provides a 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. Abstractions for both single and multi-agent systems can be found in [11–16]. Compositional frameworks are provided in [14] for safety specifications of discrete time systems, and [15], which is focused on feedback linearizable systems with a cascade interconnection. In addition, local invariant sets for discrete time coupled linear systems are considered in [17] and are leveraged for control synthesis. The above results are therefore not applicable to the decentralized abstraction of the multi-agent control systems we consider, which evolve in continuous time and do not require a specific network interconnection.

Motivated by our previous work [16], we start from the consensus dynamics of each agent and we construct a Weighted Transition System (WTS) for each agent in a decentralized manner. Each agent is assigned an individual task given in MITL formulas. We aim to design the free inputs so that each agent performs the desired individual task within specific time bounds. In particular, we provide an automatic controller synthesis method for coupled multi-agent systems under high-level tasks with timed constraints. A motivation for this framework comes from applications such as the deployment of aerial robotic teams. In particular, the consensus coupling allows the robots to stay sufficiently close to each other and maintain a connected network during the evolution of the system. Additionally, individual MITL formulas are leveraged to assign area monitoring tasks to each robot individually. The MITL formalism enables us to impose time constraints on the monitoring process. The interested reader is referred to [18] for an extended version of this paper that includes additional examples, detailed derivations and proofs.

2 Notation and Preliminaries

Denote by $\mathbb{R}, \mathbb{Q}_+, \mathbb{N}$ the set of real, nonnegative rational and natural numbers including 0, respectively. Given a set S, we 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. For a subset S of $\mathbb{R}^n$, denote by $\mathrm{cl}(S), \mathrm{int}(S)$ and $\partial S = \mathrm{cl}(S) \backslash \mathrm{int}(S)$ its closure, interior and boundary, respectively. 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 $A \in \mathbb{R}^{m \times n}$; an *undirected graph* $\mathcal{G}$ is a pair $(\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, \ldots, N\}$ is a finite set of nodes, representing a team of agents, and $\mathcal{E} \subseteq \{\{i, j\} : i, j \in \mathcal{I}, i \neq j\}$, is the set of edges that model the communication capability between the neighboring agents. For each agent, its neighbors' set $\mathcal{N}(i)$ is defined as $\mathcal{N}(i) = \{j_1, \ldots, j_{N_i}\} = \{j \in \mathcal{I} : \{i, j\} \in \mathcal{E}\}$ where $N_i = |\mathcal{N}(i)|$. The Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{N \times N}$ of the graph $\mathcal{G}$ is defined as $L(\mathcal{G}) = D(\mathcal{G}) D(\mathcal{G})^\top$ where $D(\mathcal{G})$ is the $N \times |\mathcal{E}|$ incidence matrix, as it is defined in [19, Chapter 2]. If we consider 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 ([19, Chapter 2]). Denote by $\widetilde{x} \in \mathbb{R}^{|\mathcal{E}|n}$ 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 $D(\mathcal{G})$.

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 int$(S_\ell) \cap$ int$(S_{\hat{\ell}}) = \emptyset$ for all $\ell, \hat{\ell} \in \mathbb{I}$ with $\ell \neq \hat{\ell}$ and $\cup_{\ell \in \mathbb{I}} S_\ell = \mathcal{D}$. The interiors of the cells are non-empty.*

Definition 2 *([20]) A time sequence $\tau = \tau(0)\tau(1)\ldots$ is an infinite sequence of time values $\tau(j) \in \mathbb{T}$, with $\mathbb{T} = \mathbb{Q}_+$, 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 $\tau(j) > t$.*

Definition 3 *([20]) An atomic proposition p is a statement that is either True ($\top$) or False ($\perp$). 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 to each transition; AP is a finite set of atomic propositions; and $L : S \to 2^{AP}$ is a labeling function. For every $s \in S$ and $\alpha \in Act$ define $Post(s, \alpha) = \{s' \in S : (s, \alpha, s') \in \longrightarrow\}$.*

Definition 5 *A timed run of a WTS is an infinite sequence $r^t = (r(0), \tau(0))(r(1), \tau(1))\ldots$, 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 $\alpha(1)\alpha(2)\ldots$ 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) $\tau(j+1) = \tau(j) + d(r(j), \alpha(j), r(j+1))$, $\forall\, j \geq 1$. Every timed run r^t generates a timed word $w(r^t) = (w(0), \tau(0))\ (w(1), \tau(1))\ldots$ over the set $2^{AP} \times \mathbb{T}$ where $w(j) = L(r(j)), \forall\, j \geq 0$ is the subset of atomic propositions that are true at state $r(j)$.*

The syntax of *Metric Interval Temporal Logic (MITL)* over a set of atomic propositions AP is defined by the grammar: $\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc_I \varphi \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1\, \mathcal{U}_I\, \varphi_2$, where $p \in AP$, and $\bigcirc$, $\Diamond$, $\Box$, and $\mathcal{U}$, are the next, eventually, always, and until, temporal operator, respectively; $I = [a, b] \subseteq \mathbb{T}$ where $a, b \in [0, \infty]$ with $a < b$ is a non-empty timed interval. The MITL formulas are interpreted over timed words like the ones produced by a WTS which is given in Def. 5. The semantics of MITL can be found in [18, Section 2]. It has been proved that MITL is decidable in infinite words and point-wise semantics, which is the case considered here (see [21] for details).

Let $C = \{c_1, \ldots, c_{|C|}\}$ 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.

Definition 6 *([20, 22, 23]) A Timed Büchi Automaton is a tuple $\mathcal{A} = (Q, Q^{init}, C, Inv, E, F, AP, \mathcal{L})$ where Q is a finite set of locations; $Q^{init} \subseteq Q$ 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 of the form $e = (q, \gamma, R, q')$, where q, q' are the source and target states, γ is the guard of edge e and R is a set of clocks to be reset upon executing the edge; $F \subseteq Q$ 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.*

Any MITL formula φ over AP can be algorithmically translated into a TBA with the alphabet 2^{AP}, such that the language of timed words that satisfy φ is the language of timed words produced by the TBA [21,24,25].

3 Problem Formulation

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

$$\dot{x}_i = -\sum_{j \in \mathcal{N}(i)} (x_i - x_j) + v_i, x_i \in \mathbb{R}^n, i \in \mathcal{I}. \quad (1)$$

The dynamics (1) 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 $v_{\max}$, i.e., $\|v_i(t)\| \leq v_{\max},\ \forall\ i \in \mathcal{I}, t \geq 0$. The topology of the multi-agent network is modeled through an undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, where $\mathcal{I} = \{1, \ldots, N\}$, and we assume that $\mathcal{G}$ is undirected, connected and static i.e., every agent preserves the same neighbors for all times.

Our goal is to control the multi-agent system (1) so that each agent's behavior obeys a desired individual specification φ_i given in MITL. In particular, it is required to drive each agent to a desired subset 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* $\Sigma_i, i \in \mathcal{I}$. The position x_i of each agent $i \in \mathcal{I}$ is labeled with services that are offered there. Thus, a *service labeling function* $\Lambda_i : \mathbb{R}^n \to 2^{\Sigma_i}$, 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 the agent i *can* provide in position x_i. It is noted that although the term service labeling function it is used, these functions are not necessarily related to the labeling functions of a WTS as in Definition 4. Define also by $\Lambda(x) = \bigcup_{i \in \mathcal{I}} \Lambda_i(x)$ the union of all the service labeling functions. We also 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 necessary for formally defining the problem.

Assumption 1 *There exists a decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ of the workspace $\mathbb{R}^n$ which forms a cell decomposition according to Def. 1 and respects the labeling function Λ i.e., for all $S_\ell \in S$ it holds that $\Lambda(x) = \Lambda(x'), \forall\, x, x' \in S_\ell$. This assumption implies that the same services hold at all the points that belong to the same cell of the decomposition.*

Define for each agent $i \in \mathcal{I}$ the labeling function $\mathcal{L}_i : S \to 2^{\Sigma_i}$, which denotes the fact that when agent i visits a region $S_\ell \in S$ it can choose to *provide* a subset of the 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 $x_i(t)$ is associated with a unique sequence $r^t_{x_i} = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2))\ldots$, of regions that the agent i crosses, where for all $j \geq 0$ it holds that: $x_i(\tau_i(j)) \in r_i(j)$ and $\Lambda_i(x_i(t)) = \mathcal{L}_i(r_i(j)), \forall\ t \in [\tau_i(j), \tau_i(j+1))$ for some $r_i(j) \in S$ and $r_i(j) \neq r_i(j+1)$. The timed word $w^t_{x_i} = (\mathcal{L}_i(r_i(0)), \tau_i(0))(\mathcal{L}_i(r_i(1)), \tau_i(1))(\mathcal{L}_i(r_i(2)), \tau_i(2))\ldots$, 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 $x_i(t), t \geq 0$.

Define a *timed service word* as $\widetilde{w}^t_{x_i} = (\beta_i(z_0), \widetilde{\tau}_i(z_0))(\beta_i(z_1), \widetilde{\tau}_i(z_1))(\beta_i(z_2), \widetilde{\tau}_i(z_2))\ldots$, 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 $\widetilde{\tau}_i(z_j) \in [\tau_i(z_j), \tau_i(z_j + 1))$. The timed service word is a sequence of services that are actually provided by agent i and is compliant with the trajectory $x_i(t), t \geq 0$ by construction.

The specification task φ_i given as an MITL formula over the set of services Σ_i, 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 formula φ_i given in MITL over the set Σ_i, and formally write $x_i(t) \models \varphi_i, \forall t \geq 0$, if and only if there exists a *timed service word* $\widetilde{w}^t_{x_i}$ that complies with $x_i(t)$ and satisfies φ_i.

Problem 1 *Given N agents that are governed by dynamics as in* (1)*, modeled by the undirected communication graph $\mathcal{G}$, N task specification formulas $\varphi_1, \ldots, \varphi_N$ expressed in MITL over the mutually disjoint sets of services $\Sigma_1, \ldots, \Sigma_N$, respectively, service labeling functions $\Lambda_1, \ldots, \Lambda_N$, a cell decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ as* *in Assumption 1 and the labeling functions $\mathcal{L}_1, \dots, \mathcal{L}_N$, assign control laws to the free inputs $v_1, \dots, v_N$ such that each agent fulfills its individual specification i.e., $x_i(t) \models \varphi_i, \forall i \in \mathcal{I}, t \geq 0$, given the upper bound v_{max}.*

4 Proposed Solution

In this section, a systematic solution to Problem 1 is introduced. Our overall approach builds on abstracting system (1) 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 $\varphi_i, i \in \mathcal{I}$. The following analysis is performed: (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. (Section 4.1); (2) We utilize decentralized abstraction techniques for the multi-agent system, i.e., a discretization of both the workspace and time in order to model the motion capabilities of each agent by a WTS $\mathcal{T}_i$, $i \in \mathcal{I}$ (Section 4.2); (3) Given the WTSs, consistent runs are defined in order to take into consideration the coupling constraints among the agents. The computation of the product of the individual WTSs is also required (Section 4.3); (4) A five-step automated procedure for controller synthesis which serves as a solution to Problem 1 is provided in Section 4.4; (5) Finally, the computational complexity of the proposed approach is discussed in Section 4.5.

4.1 Boundedness Analysis

Theorem 1 *Consider the multi-agent system* (1) *modeled by the undirected communication graph $\mathcal{G}$. Assume that $\mathcal{G}$ 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 $\bar{R} > K_2 v_{max}$ be a positive constant, where $K_2 = \frac{2\sqrt{N}(N-1)\|D(\mathcal{G})^\top\|}{\lambda_2^2(\mathcal{G})} > 0$ 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 $\widetilde{x}(t) \in \mathcal{X}$, $\forall t \geq T$, where $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\widetilde{x}\| \leq \bar{R}\}$.*

PROOF. (Sketch) The proof is based on assuming the Lyapunov function $V(x) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j \in \mathcal{N}(i)} \|x_i - x_j\|^2$ and showing that $\dot{V} \leq -\frac{\lambda_2^2(\mathcal{G})}{2(N-1)}\|\widetilde{x}\|\left(\|\widetilde{x}\| - K_2 v_{\max}\right) < 0$ when $\|\widetilde{x}\| > K_2 v_{\max}$. Thus, there exists a finite time $T > 0$ such that the trajectory will enter the compact set $\mathcal{X}$ and remain there for all future times. □

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 (1). This bound will be later exploited in order to capture the behavior of the system in $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\widetilde{x}\| \leq \bar{R}\}$, by a discrete state WTS.

4.2 Abstraction

In this section we provide the abstraction technique that is adopted from our previous work [16] in order to capture the dynamics of each agent into Transition Systems. Firstly, some additional notation is introduced. Given an index set $\mathbb{I}$ and an agent $i \in \mathcal{I}$ with neighbors $j_1, \dots, j_{N_i} \in \mathcal{N}(i)$, define the mapping $\mathrm{pr}_i : \mathbb{I}^N \to \mathbb{I}^{N_i+1}$ which assigns to each N-tuple $\mathbf{l} = (l_1, \dots, l_N) \in \mathbb{I}^N$ the $N_i + 1$ tuple $\mathbf{l}_i = (l_i, l_{j_1}, \dots, l_{j_{N_i}}) \in \mathbb{I}^{N_i+1}$ which denotes the indices of the cells where the agent i and its neighbors belong.

4.2.1 Well-Posed Abstractions

Loosely speaking, an abstraction is characterized by a discretization of the workspace into cells, which we denote by $\bar{S} = \{\bar{S}_l\}_{l \in \mathbb{I}}$, a time step δt and selection of feedback laws in place of the free inputs $v_i, \forall i \in \mathcal{I}$. The time step δt models the time that an agent needs to transit from one cell to another, and v_i is the controller that guarantees such a transition. Note that the time step δt is the same for all the agents. Let us denote by $(\bar{S}, \delta t)$ the aforementioned *space-time discretization*.

Before defining formally the concept of well-posed abstractions, an intuitive graphical representation is provided. Consider a cell decomposition $\bar{S} = \{\bar{S}_l\}_{l \in \mathbb{I} = \{1, \dots, 12\}}$ as depicted in Fig. 1 and a time step δt. The tails and the tips of the arrows in the figure depict the initial state and the endpoints of agent's i trajectories at time δt respectively. In both cases in the figure we focus on agent i and consider the same cell configuration for i and its neighbors. By configuration we mean the cell that the agent i and its neighbors belong at a current time. 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 δt. 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 δt, 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).

More specifically, consider a $(\bar{S}, \delta t)$-space-time discretization which is the outcome of the abstraction technique that is designed for the problem solution and will be presented in Section 4.2.3. Let

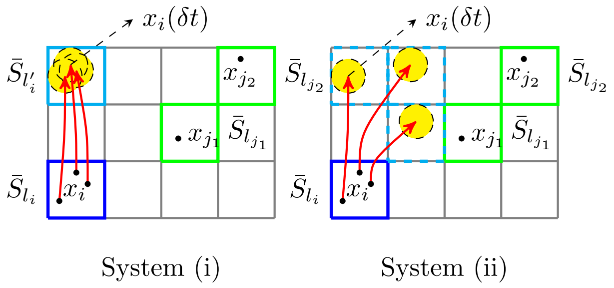

Fig. 1. Illustration of a space-time discretization which is well posed for system (i) but non-well posed for system (ii).

$\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{I}}}$ be a cell decomposition in which the agent i occupies the cell $\bar{S}_{l_i}$, δt be a time step and $\bar{d}_{\max} = \sup\{\|x-y\| : x, y \in \bar{S}_l, l \in \mathbb{I}\}$ be the diameter of the cell decomposition $\bar{S}$. It should be noted that this decomposition is not necessarily the same cell decomposition S from Assumption 1 and Problem 1. Through the aforementioned space and time discretization $(\bar{S}, \delta t)$ we aim to capture the reachability properties of the continuous system (1), by creating an individual WTS for each agent. 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 at time δt, 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.

4.2.2 Sufficient Conditions

We present at this point the sufficient conditions that relate the dynamics of the multi-agent system (1), the time step δt and the diameter $\bar{d}_{\max}$, and guarantee the existence of the aforementioned well-posed transitions for each cell. Based on our previous work [16], in order to derive well-posed abstractions, a nonlinear system of the form:

$$\dot{x}_i = f_i(x_i, \mathbf{x}_j) + v_i, i \in \mathcal{I}, \tag{2}$$

where $\mathbf{x}_j = (x_{j_1}, \dots, x_{j_{N_i}}) \in \mathbb{R}^{N_i n}$, should fulfill the following sufficient conditions: **(C1)** There exists $M > v_{\max} > 0$ such that $\|f_i(x_i, \mathbf{x}_j)\| \le M$, $\forall i \in \mathcal{I}, \forall\, x \in \mathbb{R}^{Nn}$: $\text{pr}_i(x) = (x_i, \mathbf{x}_j)$ and $\widetilde{x} \in \mathcal{X}$, by applying the projection operator pr_i for $\mathbb{I} = \mathbb{R}^n$; **(C2)** There exist Lipschitz constants L_1, $L_2 > 0$ such that: $\|f_i(x_i, \mathbf{x}_j) - f_i(x_i, \mathbf{y}_j)\| \le L_1 \|(x_i, \mathbf{x}_j) - (x_i, \mathbf{y}_j)\|$ and $\|f_i(x_i, \mathbf{x}_j) - f_i(y_i, \mathbf{x}_j)\| \le L_2 \|(x_i, \mathbf{x}_j) - (y_i, \mathbf{x}_j)\|$, $\forall\, i \in \mathcal{I}, \forall\, i \in \mathcal{I}$, x_i, $y_i \in \mathbb{R}^n$, $\mathbf{x}_j$, $\mathbf{y}_j \in \mathbb{R}^{N_i n}$.

Using the result of Theorem 1, it can be shown that system (1) with $f_i(x_i, \mathbf{x}_j) = -\sum_{j \in \mathcal{N}(i)} (x_i - x_j)$ satisfies the the conditions **(C1)-(C2)**. The proof can be found in [18, Appendix C]. Based on the sufficient conditions for well posed abstractions in [16], the diameter $\bar{d}_{\max}$ and time step δt of the discretization $(\bar{S}, \delta t)$ can be selected as: $\bar{d}_{\max} \in \left(0, \frac{(1-\lambda)^2 v_{\max}^2}{4ML}\right]$,

$$\delta t \in \Big[\tfrac{(1-\lambda)v_{\max} - \sqrt{(1-\lambda)^2 v_{\max}^2 - 4ML\bar{d}_{\max}}}{2ML}, \tfrac{(1-\lambda)v_{\max} + \sqrt{(1-\lambda)^2 v_{\max}^2 - 4ML\bar{d}_{\max}}}{2ML}\Big],$$

where $L = 3L_2 + 4L_1 \max\left\{\sqrt{N_i}, i \in \mathcal{I}\right\}$ and with the dynamics bound M and the Lipschitz constants L_1, L_2 as previously defined. Furthermore, $\lambda \in (0, 1)$ is a design parameter which quantifies the part of the free input that is additionally exploited for reachability purposes. In particular, given an agent's initial cell configuration, the agent can reach any point inside an appropriate ball at δt through a parameterized feedback law in place of the free input v_i. The center of this ball is selected as the endpoint of a reference trajectory for agent i, which is obtained by considering its neighbors fixed at certain reference points in their cells during the transition, providing thus an estimate of the agent's reachable states at δt. The radius of this ball increases proportionally to λ, and thus, also to the number of the agent's successor cells, which are the ones intersecting the ball. It is noted that an increasing choice of λ results in finer discretizations, therefore providing a quantifiable trade-off between the discrete model's accuracy and complexity. Furthermore, it follows from the acceptable values of $\bar{d}_{\max}$ that the cells can be selected coarser, when (i) the available control $v_{\max}$ is larger, and, (ii) the coupling term bound M together with the dynamics' variation, which is captured through the parameter L, are smaller. Analogous restrictions need to hold for the time step δt. In particular, the time step cannot be selected very large, because of the feedback which is used to modify the agent's couplings in accordance to the dynamics of its reference trajectory. More specifically, the corresponding control effort increases with time due to the evolution of the agent's neighbors away from their reference points during the transition interval. Finally, the time step cannot be selected very small compared to the diameter of the cells, because controlling the agent to the same point from each initial condition in its cell, will require a large control effort over a very short transition interval.

Having shown that the dynamics of system (1) satisfy the sufficient conditions **(C1)-(C2)**, a well-posed space-time discretization $(\bar{S}, \delta t)$ has been obtained. Recall now Assumption 1. 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: $\bar{S}_l \cap S_\ell \in S \cup \{\emptyset\}, \forall\, \bar{S}_l \in \bar{S}, S_\ell \in S, l \in \bar{\mathbb{I}}, \ell \in \mathbb{I}$. In order to address this problem, define: $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l} \in \hat{\mathbb{I}}} = \{\bar{S}_l \cap S_\ell : l \in \bar{\mathbb{I}}, \ell \in \mathbb{I}\} \backslash \{\emptyset\}$, which forms a cell decomposition and is compliant with the cell decomposition S from Problem 1 with diameter $\hat{d}_{\max} = \sup\{\|x-y\| : x, y \in \hat{S}_{\hat{l}}, \hat{l} \in \hat{\mathbb{I}}\} \le \bar{d}_{\max}$ and serves as the abstraction solution of this problem.

4.2.3 Discrete System Abstraction

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

Definition 7 *The motion of each agent* $i \in \mathcal{I}$ *in the workspace is modeled by the WTS* $\mathcal{T}_i = (S_i, S_i^{init}, Act_i, \longrightarrow_i, d_i, AP_i, \hat{L}_i)$ *where:* $S_i = \hat{\mathbb{I}}$ *is the set of states of each agent which is the set of indices of the cell decomposition;* $S_i^{init} \subseteq S_i$ *is a set of initial states defined by the agents' initial positions in the workspace;* $Act_i = \hat{\mathbb{I}}^{N_i+1}$*, 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* $\mathbf{l}_i = (l_i, l_{j_1}, \ldots, l_{j_{N_i}}) \in Act_i$*;* $d_i : \longrightarrow_i \rightarrow \mathbb{T}$*, 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;* $L_i : S_i \rightarrow 2^{AP_i}$*, is the labeling function that maps every state* $s \in S_i$ *into the services that can be provided in this state.*

Every WTS $\mathcal{T}_i$, $i \in \mathcal{I}$ generates timed runs and timed words of the form $r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \ldots$, $w_i^t = (L_i(r_i(0)), \tau_i(0))(L_i(r_i(1)), \tau_i(1))(L_i(r_i(2)), \tau_i(2)) \ldots$ respectively, over the set 2^{AP_i} according to Def. 5 with $\tau_i(j) = j\delta t, \forall\, j \geq 0$.

4.3 Consistency of Runs

Due to the coupled dynamics between the agents, it is required that each individual agent's run is compliant with the corresponding discrete trajectories of its neighbors, which determine the actions in the agent's run. Therefore, even though we have the individual WTS of each agent, the runs that the latter generates may not be performed by the agent due to the constrained motion that is imposed by the coupling terms. Hence, we need to synchronize the agents at each time step δt and determine which of the generated runs of the individual WTS can be performed by the agent. Hereafter, they will be called *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. 9) can later be projected onto consistent individual runs. The following two definitions deal with the product WTS and consistent runs respectively.

Definition 8 *Given the individual WTSs* $\mathcal{T}_i, i \in \mathcal{I}$ *from Def. 7, the product WTS* $\mathcal{T}_p = (S_p, S_p^{init}, \longrightarrow_p, \cup_{i=1}^N \Sigma_i, L_p)$ *is defined as follows:* $S_p = \hat{\mathbb{I}}^N$*;* $(s_1, \ldots, s_N) \in S^{init}$ *if* $s_i \in S_i^{init}, \forall i \in \mathcal{I}$*;* $(\mathbf{l}, \mathbf{l}') \in \longrightarrow_p$ *iff* $l_i' \in Post_i(l_i, pr_i(\mathbf{l})), \forall i \in \mathcal{I}, \forall \mathbf{l} = (l_1, \ldots, l_N), \mathbf{l}' = (l_1', \ldots, l_N')$*;* $L_p : \hat{\mathbb{I}}^N \rightarrow 2^{\cup_{i=1}^N \Sigma_i}$ *defined as* $L_p(\mathbf{l}) = \cup_{i=1}^N L_i(l_i)$*;* $d_p : \longrightarrow_p \rightarrow \mathbb{T}$ *as in the individual WTS's case, with transition weight* $d_p(\cdot) = \delta t$*.*

Definition 9 *Given a timed run* $r_p^t = ((r_p^1(0), \ldots, r_p^N(0)), \tau_p(0))((r_p^1(1), \ldots, r_p^N(1)), \tau_p(1)) \ldots$*, that is generated by the product 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* δt *it holds that* $\tau_p(j) = j\delta t, j \geq 0$*.*

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 time serviced words $\widetilde{w}_1^t, \ldots, \widetilde{w}_N^t$. Note that we use the tilde accent to denote timed runs and words that correspond to the problem solution. The corresponding compliant trajectories $x_1(t), \ldots, x_N(t)$ of the timed words $\widetilde{w}_1^t, \ldots, \widetilde{w}_N^t$ satisfy the corresponding MITL formulas $\varphi_1, \ldots, \varphi_N$, and they are a solution to Problem 1. This follows from the fact that the product transition system is simulated by the δt-sampled version of the continuous system (see [26] for the definition of a simulation relation). In particular, let $\mathcal{T}_{\delta t}$ be the δt-sampled WTS of system (1), as defined in [26, Def. 11.4], with labeling function $L_{\delta t} : \mathbb{R}^{Nn} \rightarrow 2^{\cup_{i=1}^N \Sigma_i}$ given as $L_{\delta t}(x_1, \ldots, x_N) = \cup_{i=1}^N \Lambda_i(x_i)$ and Λ_i as defined in Section 2. Consider also the WTS $\mathcal{T}_p$ and the relation $\mathcal{R} \subseteq S_p \times \mathcal{X}^N$ given as $(\mathbf{l}, (x_1, \ldots, x_N)) \in \mathcal{R}$, iff $(x_1, \ldots, x_N) \in S_{l_1} \times \cdots \times S_{l_N}$, where $\mathbf{l} = (l_1, \ldots, l_N)$. Then, from the definition of the agent's individual transitions in each WTS $\mathcal{T}_i$ and the fact that for all points in a cell the same atomic propositions hold true, it can be deduced that $\mathcal{R}$ is a simulation relation from $\mathcal{T}_p$ to the δt-sampled WTS $\mathcal{T}_{\delta t}$.

4.4 Controller Synthesis

The proposed controller synthesis procedure is described with the following steps; **Step 1**: N TBAs $\mathcal{A}_i$, $i \in \mathcal{I}$ that accept all the timed runs satisfying the corresponding specification formulas φ_i are constructed; **Step 2**: A Büchi WTS $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i$ 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 φ_i; **Step 3**: We pick a set of accepting runs $\{\widetilde{r}_1^t, \ldots, \widetilde{r}_N^t\}$ from Step 2. We check if they are consistent according to Def. 9. 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, we perform a finite predefined number of selections R_{selec}; if a consistent set of accepting runs is not found, we proceed with the less efficient centralized procedure in Step 4, which however searches through all sets of all possible accepting runs; **Step 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 $\{\widetilde{r}_1, \ldots, \widetilde{r}_N\}$; **Step 5**: The abstraction procedure allows to find an explicit feedback law for each transition in $\mathcal{T}_i$. Therefore, an ac-

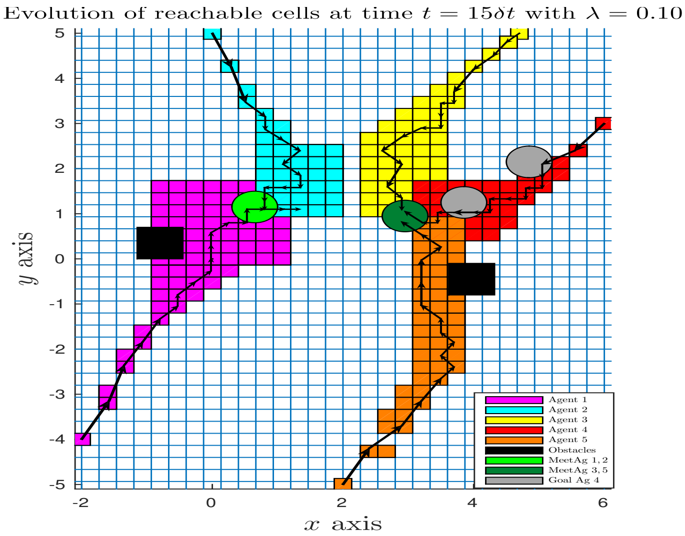

Fig. 2. A simulation scenario with $N = 5$ agents and $\lambda = 0.10$, $\bar{d}_{\max} = 0.25$. The figure shows the evolution of the agents' reachable cells up to time $t = 15\delta t$

.

cepting run $\widetilde{r}_i^t$ in $\mathcal{T}_i$ that takes the form of a sequence of transitions is realized in the system in (1) via the corresponding sequence of feedback laws.

For the constructions of the Büchi WTSs $\widetilde{\mathcal{T}}_p$ and $\widetilde{\mathcal{T}}_i, i \in \mathcal{I}$ defined in Steps 2 and 4, we refer the reader to [18, Section 4, Def. 10]. 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 $\widetilde{r}_i^t = (q_i(0), \tau_i(0))(q_i(1), \tau_i(1)) \ldots$ using the terminology of Def. 5. It is *accepting* if $q_i(j) \in \widetilde{F}_i$ 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.

Proposition 1 *A solution obtained from Steps 1-5, gives a sequence of controllers* $v_1, \ldots, v_N$ *that guarantees the satisfaction of the formulas* $\varphi_1, \ldots, \varphi_N$ *of the agents* $1, \ldots, N$ *respectively, governed by the dynamics as in* (1)*, thus, they are a solution to Problem 1.*

4.5 Complexity

Denote by $|\varphi|$ the length of an MITL formula φ. A TBA $\mathcal{A}_i, i \in \mathcal{I}$ can be constructed in space and time $2^{\mathcal{O}(|\varphi_i|)}, i \in \mathcal{I}$. Let $\varphi_{\max} = \max_{i \in \mathcal{I}}\{|\varphi_i|\}$ be the MITL formula with the longest length. Then, the complexity of Step 1 is $N2^{\mathcal{O}(|\varphi_{\max}|)}$. Step 2 costs $\mathcal{O}(N2^{|\varphi_i|}|\mathcal{S}_i|)$, where $|\mathcal{S}_i| = |\hat{\mathbb{I}}|$ is the number of states of the WTS $\mathcal{T}_i$. We have the best case complexity as $\mathcal{O}(NR_{\text{selec}}2^{|\varphi_{\max}|}|\hat{\mathbb{I}}|)$, 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 $\mathcal{O}(2^{|\varphi_{\max}|}|\hat{\mathbb{I}}|^N)$.

Step	Reach. States	Time	Step	Reach. States	Time
δt	4	0.03 sec	$9\delta t$	4960	27.01 sec
$2\delta t$	16	0.05 sec	$10\delta t$	10822	68.79 sec
$3\delta t$	24	0.05 sec	$11\delta t$	20706	102.46 sec
$4\delta t$	40	0.06 sec	$12\delta t$	34856	153.95 sec
$5\delta t$	126	0.09 sec	$13\delta t$	59082	194.68 sec
$6\delta t$	378	1.59 sec	$14\delta t$	69060	220.12 sec
$7\delta t$	1074	3.51 sec	$15\delta t$	74546	322.61 sec
$8\delta t$	2908	10.06 sec		Total Time: 1105 sec	

Table 1
This table shows the simulation statistics. Columns 1, 4 show the considered time steps. Columns 2, 5 show the corresponding number of states reachable in the WTS $\mathcal{T}_p$. Columns 3, 6 indicate the time required for their computation.

5 Simulation Results

For the simulation example, a multi-agent system with $x_i \in \mathbb{R}^2$, $i \in \mathcal{I} = \{1, 2, 3, 4, 5\}$, $\mathcal{N}(1) = \{2\}, \mathcal{N}(2) = \{1, 3\}$, $\mathcal{N}(3) = \{2, 4\}$, $\mathcal{N}(4) = \{3, 5\}$ and $\mathcal{N}(5) = \{4\}$ is considered. According to (1), the dynamics are given as: $\dot{x}_1 = -(x_1 - x_2) + v_1, \dot{x}_2 = -(x_2 - x_1) - (x_2 - x_3) + v_2$, $\dot{x}_3 = -(x_3 - x_2) - (x_3 - x_4) + v_3$, $\dot{x}_4 = -(x_4 - x_3) - (x_4 - x_5) + v_4$ and $\dot{x}_5 = -(x_5 - x_4) + v_5$. The simulation parameters are set to $\delta t = 0.1$, $\lambda = 0.10$, and $\bar{d}_{\max} = 0.25$, with $L_1 = \sqrt{2}$ and $L_2 = 2$ obtained from the agents' dynamics. The workspace is decomposed into square cells, which are depicted with blue color in Fig. 2. The initial agents' positions are set to $(-2, 4)$, $(0, 5)$, $(4.7, 5)$, $(6, 3)$ and $(4, -5)$. The specification formulas are set to $\varphi_1 = \Diamond_{[1.2, 1.7]} \{\text{meet}_{12}\} \wedge \Box_{[0,2]} \{\neg\text{obs}_1\}$, $\varphi_2 = \Diamond_{[1.2, 1.7]} \{\text{meet}_{21}\}$, $\varphi_3 = \Diamond_{[1.5, 1.8]} \{\text{meet}_{35}\}$, $\varphi_4 = \Diamond_{[0.1, 0.5]} \{\text{grey}_1\} \wedge \Diamond_{[0.7, 1.2]} \{\text{grey}_2\}$, and $\varphi_5 = \Diamond_{[1.2, 1.8]} \{\text{meet}_{53}\} \wedge \Box_{[0,2]} \{\neg\text{obs}_2\}$, respectively. Descriptively, agents 1, 2 as well as agents 3, 5 need to meet in the light and dark green region, respectively, within certain time bounds; agents 1 and 5 have additional safety specifications; and agent 4 has two reachability goals within certain time bounds (deadlines). The cell decomposition presented in this paper, the reachable cells of each agent up to time $t = 15\delta t$ and the goal regions are depicted in Fig. 2. The reachable cells of each agent are depicted with purple, cyan, yellow, red and orange, respectively. The individual consistent runs $\widetilde{r}_1^t$, $\widetilde{r}_2^t$, $\widetilde{r}_3^t$, $\widetilde{r}_4^t$ and $\widetilde{r}_5^t$ of agents 1, 2, 3, 4 and 5, respectively, that satisfy the formulas φ_1, φ_2, φ_3, φ_4 and φ_5, respectively are depicted in Fig. 2 with black arrows. Each arrow represents a transition from a state to another according to Def. 7. Table 1 shows the simulation statistics of the simulation scenario. For each time step δt, $2\delta t$, $\ldots$, $15\delta t$ the number of reachable states of the product WTS $\mathcal{T}_p$ is mentioned, along with the necessary computation time. The simulation takes 1182 sec (1105 sec for the abstraction and 77 sec for the graph search) on a desktop with 8 cores, 3.60GHz CPU and 16GB of RAM.

6 Conclusions and Future Work

A systematic method for 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 a boundedness analysis, the abstraction of each agent's motion into WTSs, TBAs as well as Büchi WTSs 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] 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)*, pages 101–110, 2010.

[2] M. Guo and D. V. Dimarogonas. Multi-Agent Plan Reconfiguration Under Local LTL Specifications. *The International Journal of Robotics Research*, 34(2):218–235, 2015.

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

[4] I. Saha, R. Ramaithitima, V. Kumar, G. Pappas, and S. Seshia. Implan: Scalable Incremental Motion Planning for Multi-Robot Systems. *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCP)*, 2016.

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

[6] 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)*, pages 239–248, 2015.

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

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

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

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

[11] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete Abstractions of Hybrid Systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[12] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic Models for Nonlinear Control Systems without Stability Assumptions. *IEEE Transactions on Automatic Control (TAC)*, 57(7), 2012.

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

[14] P. Meyer, A. Girard, and E. Witrant. Compositional Abstraction and Safety Synthesis Using Overlapping Symbolic Models. *IEEE Transaction on Automatic Control (TAC)*, 2017.

[15] O. Hussein, A. Ames, and P. Tabuada. Abstracting Partially Feedback Linearizable Systems Compositionally. *IEEE Control Systems Letters, vol. 1, no. 2, pp. 227-232*, Oct. 2017.

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

[17] P. Nilsson and N. Ozay. Synthesis of Separable Controlled Invariant Sets for Modular Local Control Design. *American Control Conference (ACC)*, July 6-8, 2016.

[18] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas. On the Timed Temporal Logic Planning of Couple Multi-Agent Systems. *2017, https://arxiv.org/abs/1709.06888.*

[19] M. Mesbahi and M. Egerstedt. Graph Theoretic Methods in Multiagent Networks. 2010.

[20] R. Alur and D.L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[21] R. Alur, T. Feder, and T. A. Henzinger. The Benefits of Relaxing Punctuality. *Journal of the ACM (JACM)*, 43(1):116–146, 1996.

[22] P. Bouyer. From Qualitative to Quantitative Analysis of Timed Systems. *Mémoire d'habilitation, Université Paris*, 7:135–175, 2009.

[23] S. Tripakis. Checking Timed Buchi Automata Emptiness on Simulation Graphs. *ACM Transactions on Computational Logic (TOCL)*, 10(3), 2009.

[24] O. Maler, D. Nickovic, and A. Pnueli. From MITL to Timed Automata. *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 274–289, 2006.

[25] T. Brihaye, G. Geeraerts, H. Ho, and B. Monmege. MightyL: A Compositional Translation from MITL to Timed Automata. *29th International Conference on Computer Aided Verification (CAV)*, 2017.

[26] P. Tabuada. *Verification and Control of Hybrid Systems: a Symbolic Approach.* Springer Science and Business Media, 2009.