

# Scalable Time-constrained Planning of Multi-robot Systems

Alexandros Nikou · Shahab Heshmati-alamdari · Dimos V. Dimarogonas

Received: date / Accepted: date

**Abstract** This paper presents a scalable procedure for time-constrained planning of a class of uncertain nonlinear multi-robot systems. In particular, we consider  $N$  robotic agents operating in a workspace which contains Regions of Interest (RoI), in which atomic propositions for each robot are assigned. The main goal is to design decentralized and robust control laws so that each robot meets an individual high-level specification given as a Metric Interval Temporal Logic (MITL), while using only local information based on a limited sensing radius. Furthermore, the robots need to fulfill certain desired transient constraints such as collision avoidance between them. The controllers, which guarantee the transition between regions, consist of two terms: a nominal control input, which is computed online and is the solution of a Decentralized Finite-Horizon Optimal Control Problem (DFHOCP); and an additive state feedback law which is computed offline and guarantees that the real trajectories of the system will belong to a hyper-tube centered along the nominal trajectory. The controllers serve as actions for the individual Weighted Transition System (WTS) of each robot, and the time duration required for the transition between regions is modeled by a weight. The DFHOCP is solved at every sampling time by each robot and then necessary information is exchanged between neighboring robots. The proposed approach is scalable since it does not require a product computation among the WTS of the robots.

---

Alexandros Nikou, Shahab Heshmati-alamdari, Dimos V. Dimarogonas  
E-mail: {anikou, shaha, dimos}@kth.se

The authors are with the Division of Decision and Control, School of Electrical Engineering and Computer Science KTH Royal Institute of Technology SE-100 44, Stockholm, Sweden.

The proposed framework is experimentally tested and the results show that the proposed framework is promising for solving real-life robotic as well as industrial applications.

**Keywords** Multi-robot systems · cooperative control · decentralized control · abstractions · Metric Interval Temporal Logic (MITL) · Nonlinear Model Predictive Control (NMPC) · robust control

## 1 Introduction

Over the last few years, the field of control of multi-robot systems under high-level specifications has been gaining significant attention [1–7]. Applications arise in the fields of autonomous driving, industrial work by autonomously operating robot systems, indoor transportation in warehouses etc [8,9]. The qualitative specification language that has primarily been used to express the high-level tasks is Linear Temporal Logic (LTL) (see, e.g., [10,11]). In practical applications, however, the desired tasks need to be accomplished within certain quantitative time bounds by the robots.

A suitable temporal logic for dealing with tasks that are required to be completed within certain time bounds is the Metric Interval Temporal Logic (MITL). MITL has been originally proposed in [12] and has been used in control synthesis frameworks in [13]. Given a robot dynamics and an MITL formula, the control design procedure is the following: first, the robot dynamics are abstracted into a Weighted Transition System (WTS), in which the time duration for navigating between states is modeled by a weight in the WTS (abstraction); second, an offline product between the WTS and an automaton that accepts the runs that satisfy the given formula is computed; and third, once an accepting run in the

product is found, it maps into a sequence of feedback control laws of the robot dynamics.

Controller synthesis for multi-robot systems under MITL specifications has been investigated in [14–17]. In our previous works [14–16], the under consideration dynamics were first order and we considered actuation over each state of each agent. Additionally, a global product WTS of the individual WTSs of each robot was computed. Furthermore, no transient constraints between the agents are taken into consideration. In particular, the work [16] can only handle multi-agent timed constrained planning in  $\mathbb{R}^2$  dimension, which is usually not the case in real-life robotic applications. Authors in [17] have addressed the Vehicle Routing Problem (VHP), which is modeled as an optimization problem, that aims at finding the optimal set of routes for a fleet of vehicles to traverse, in order to deliver the load to a given set of customers. However, the dynamics of the agents were not taken into consideration. Moreover, issues such as control input saturation and robustness against disturbances were not considered. In the same context, none of the aforementioned works deal with real-time experimental validation of the corresponding proposed frameworks.

Motivated by the aforementioned, in this work, we aim to address the latter issues. In particular, we deal with nonlinear dynamics in  $\mathbb{R}^n$  with input constraints and external uncertainties/disturbances. Then, by assigning an MITL formula to each agent, we provide decentralized feedback control laws that guarantee robust transitions between neighboring agents under transient constraints. The controllers consist of two terms: a nominal control input, which is computed online and is the solution of a Decentralized Finite-Horizon Optimal Control Problem (DFHOCP); and an additive state feedback law which is computed offline and guarantees that the real trajectories of the system will belong to a hyper-tube centered along the nominal trajectory. More specifically, the online part is responsible for minimizing the error and the control input effort in order for the robot to be navigated between RoI, while transient constraints and control input saturation are satisfied. The second part is introduced in order to guarantee that while the DFHOCP is solved for the nominal dynamics, the controller compensates for the uncertain part due to external disturbances and keeps the trajectory of the robot bounded inside a tube. Furthermore, an algorithm that computes the runs of each agent that in turn map into continuous control laws and provably satisfy the MITL formulas is provided. These control laws correspond to the transitions indicated above. The proposed scheme is experimentally validated in our lab facilities with a group of Nexus robots (see Fig. 1). More-



Fig. 1: The experimental setup demonstrating the proposed framework. Three Nexus 10011 mobile robots, in the workspace of Smart Mobility Lab (SML) [18] that contains 5 RoI.

over, the proposed approach is scalable since does not require a product computation among the WTS of the agents.

The idea of avoiding the global product between the agents lies in the fact that we address the multi-agent coupling with the low-level continuous time control design. More specifically, we exploit the inherent advantage of NMPC with reference to other control techniques: we capture the coupling constraints through the hard constraints of each agent by assuming communication capabilities between the agents. In the same context, all the algorithmic computations are performed offline and the robots are executing online only a sequence of a control actions that are the outcome of the planner. Thus, the latter leads to a framework that is scalable with the number of agents. Our contribution is thus a fully automated framework for a general class of uncertain multi-robot systems consisting of both constructing purely decentralized abstractions and conducting timed temporal logic planning with transient constraints in a scalable way.

This paper is structured as follows. In Section 2 a description of the necessary mathematical tools, the notations and the definitions are given. Section 3 provides the modeling of the proposed framework along with the formal problem statement. Section 4 discusses the technical details of the solution, while Section 5 is devoted to a real-time experiment demonstrating the proposed approach. Finally, conclusions and future research directions are discussed in Section 6.

$\mathbb{N}, \mathbb{Q}, \mathbb{R}$	Natural, rational & real numbers
$[N] := \{1, \dots, N\}$	Labeling set of robots
$[Z] := \{1, \dots, Z\}$	Labeling set of RoI
$\mathcal{R} = \bigcup_{z \in [Z]} R_z$	Union of RoI
$\mathcal{W}$	The workspace of the robots
$L$	Lipschitz constant
$\mathcal{T}_i, \mathcal{A}_i, \tilde{\mathcal{T}}_i$	WTS, TBA, product WTS of robot $i$
$(S, S^{\text{init}}, \text{Act}, \longrightarrow, t, \Gamma, \mathcal{L})$	(states, init. states, actions, transition relation, weight, atomic propos., labeling function)
$(Q, Q^{\text{init}}, \text{CL}, \text{Inv}, E, \text{FS})$	(states, init. states, clocks, invariance, accepted states)
$x_i, v_i, u_i$	position/orientation, velocity and control input of robot
$e_i = x_i - x_{i,d}$	Error signal to be minimized through DFHOCP
$\xi_i = [e_i, v_i]^\top$	stack vector
$f_i, G_i, \delta_i$	Dynamic model of robot $i$ and disturbance $\delta_i$
$\mathbf{e}_i = e_i - \bar{e}_i, \mathbf{v}_i = v_i - \bar{v}_i$	Errors between real and nominal signals (state and velocity respectively)
$\Omega_{i,1}, \Omega_{i,2}$	Invariant tubes for the errors $\mathbf{e}_i, \mathbf{v}_i$
$P_i, Q_i, R_i$	Positive definite weight matrices used in DFHOCP
$\mathcal{E}_i, \mathcal{V}_i, \mathcal{U}_i, \mathcal{F}_i$	State constraints, velocity constraints, control input constraints, terminal set of DFHOCP
$T, h$	Prediction horizon and constant sampling time period
$k_i, \rho_i$	Tube gains
$r^t, w^t$	Timed run & timed word
$\tau(l), l \geq 0$	Time stamp at position $l \geq 0$
$\mathcal{M}(x_i, r_i)$	A ball centered at $x_i$ , radius $r_i$ covering robot $i$
$d_i, \mathcal{G}_i(t)$	Sensing radius & set of neighbors of robot $i$ at time $t$
$\diamond_I, \square_I, \mathcal{U}_I$	Eventually, always and until MITL operators

Table 1: List of symbols

## 2 Notation and Preliminaries

In this section, the notation that will be used hereafter as well as the necessary mathematical background for nonlinear control systems and formal verification are provided.

### 2.1 Notation

The set of positive integers, positive rational numbers and real numbers are denoted by  $\mathbb{N}$ ,  $\mathbb{Q}_+$ , and  $\mathbb{R}$ , respectively. Given a set  $S$ , we denote by  $|S|$  its cardinality, by  $S^n = S \times \dots \times S$  its  $n$ -fold Cartesian product, and by  $2^S$  the set of all its subsets;  $\|y\|_2 := \sqrt{y^\top y}$  and  $\|y\|_M := \sqrt{y^\top M y}$ ,  $M \geq 0$  stand for the Euclidean and the weighted norm of a vector  $y \in \mathbb{R}^n$ , respectively;  $\lambda_{\min}(M)$  stands for the minimum absolute value of the real part of the eigenvalues of  $M \in \mathbb{R}^{n \times n}$ ;  $I_n \in \mathbb{R}^{n \times n}$  and  $0_{m \times n} \in \mathbb{R}^{m \times n}$  are the identity matrix and the  $m \times n$  matrix with all entries zeros, respectively. A  $n \times n$  symmetric real matrix  $M$  is said to be positive semidefinite ( $M \geq 0$ ) if  $x^\top M x \geq 0$  for every  $x \in \mathbb{R}^n$ . The notation  $\text{diag}\{M_1, \dots, M_n\}$  stands for the block diagonal matrix with the matrices  $M_1, \dots, M_n$  in the main diagonal. The set  $\mathcal{M}(\chi, \rho) = \{y \in \mathbb{R}^n : \|y - \chi\|_2 \leq \rho\}$ , represents the  $n$ -dimensional ball with center  $\chi \in \mathbb{R}^n$  and radius  $\rho \in \mathbb{R}_{>0}$ . It should be noticed that, throughout this manuscript, the nominal signals and the mag-

NMPC	Nonlinear Model Predictive Control
RoI	Regions of Interest
DFHOCP	Decentralized Finite Horizon Optimal Control Problem
TBA	Timed Büchi Automaton
WTS	Weighted Transition System
MITL	Metric Interval Temporal Logic
ISS	Input to State Stability
RCI set	Robust Control Invariant set

Table 2: List of acronyms

nitude of upper bounds of signals are denoted with  $\bar{\cdot}$ ,  $\tilde{\cdot}$ , respectively. Tables 1, 2 show a list of symbols and a list of acronyms, respectively, that are used in this manuscript.

**Definition 1** Given the sets  $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathbb{R}^n$ ,  $\mathcal{S} \subseteq \mathbb{R}^m$  and the matrix  $M \in \mathbb{R}^{n \times m}$ , the *Minkowski addition*, the *Pontryagin difference* as well as the *matrix-set multiplication* are respectively defined by:

$$\begin{aligned} \mathcal{S}_1 \oplus \mathcal{S}_2 &:= \{s_1 + s_2 : s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}, \\ \mathcal{S}_1 \ominus \mathcal{S}_2 &:= \{s_1 : s_1 + s_2 \in \mathcal{S}_1, \forall s_2 \in \mathcal{S}_2\}, \\ M \circ \mathcal{S} &:= \{m = M s, s \in \mathcal{S}\}. \end{aligned}$$

**Lemma 1** [19] For any constant  $\rho > 0$ , vectors  $z_1, z_2 \in \mathbb{R}^n$  and matrix  $P \in \mathbb{R}^{n \times n}$ ,  $P > 0$  it holds that:

$$z_1 P z_2 \leq \frac{1}{4\rho} z_1^\top P z_1 + \rho z_2^\top P z_2.$$

## 2.2 Nonlinear Control

**Definition 2** [20] A continuous function  $\alpha : [0, a) \rightarrow \mathbb{R}_{\geq 0}$  belongs to *class*  $\mathcal{K}$  if it is strictly increasing and  $\alpha(0) = 0$ . A continuous function  $\beta : [0, a) \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  belongs to *class*  $\mathcal{KL}$  if: 1) for a fixed  $s$ , the mapping  $\beta(r, s)$  belongs to *class*  $\mathcal{K}$  with respect to  $r$ ; 2) for a fixed  $r$ , the mapping  $\beta(r, s)$  is strictly decreasing with respect to  $s$ ; and it holds that:  $\lim_{s \rightarrow \infty} \beta(r, s) = 0$ .

**Definition 3** [21] Consider a dynamical system:

$$\dot{x} = f(x, u, \delta), \quad x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad \delta \in \mathcal{D},$$

with initial condition  $x(0) \in \mathcal{X}$ . A set  $\mathcal{X}' \subseteq \mathcal{X}$  is a *Robust Control Invariant (RCI) set* for the system, if there exists a feedback control law  $u := \kappa(x) \in \mathcal{U}$ , such that for all  $x(0) \in \mathcal{X}'$  and for all disturbances  $\delta \in \mathcal{D}$  it holds that  $x(t) \in \mathcal{X}'$  for all  $t \geq 0$ , along every solution  $x(t)$ .

**Definition 4** [20, Def. 4.7, p. 175] A nonlinear system  $\dot{x} = f(x, u, \delta)$ ,  $x \in \mathcal{X}$ ,  $u \in \mathcal{U}$ ,  $\delta \in \mathcal{D}$  with initial condition  $x(0) \in \mathcal{X}$  is said to be *Input-to-State Stable (ISS)* with respect to  $\delta \in \mathcal{D}$ , if there exist functions  $\beta \in \mathcal{KL}$ ,  $\gamma \in \mathcal{K}$  such that for any initial condition  $x(0) \in \mathcal{X}$  and for any input  $u(t) \in \mathcal{U}$ , the solution  $x(t)$  exists for all  $t \in \mathbb{R}_{\geq 0}$  and satisfies:

$$\|x(t)\|_2 \leq \beta(\|x(0)\|_2, t) + \gamma \left( \sup_{0 \leq s \leq t} \|\delta(s)\|_2 \right).$$

## 2.3 Nonlinear Model Predictive Control

NMPC [22] is formulated as solving at each sampling time step an online Finite Horizon Optimal Control Problem (FHOC) subject to nonlinear system dynamics and constraints involving states and controls. Based on measurements obtained at each sampling time step, the controller predicts the dynamic behavior of the system over a predictive horizon in the future and determines the input such that a predetermined open-loop performance objective is minimized. In order to incorporate feedback, the optimal open-loop input is implemented only until the next sampling time step. Using the new system state at the next sampling time step, the whole procedure - prediction and optimization - is repeated.

## 2.4 Formal Verification

**Definition 5** ([12]) A *time sequence*  $\tau = \tau(0)\tau(1)\dots$  is an infinite sequence of time values  $\tau(l) \in \mathbb{Q}_+$ , satisfying the following properties: 1) Monotonicity:  $\tau(l) <$

$\tau(l+1)$  for all  $l \in \mathbb{N}$ ; 2) Progress: For every  $t \in \mathbb{Q}_+$ , there exists  $l \geq 1$ , such that  $\tau(l) > t$ .

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

**Definition 6** ([12]) Let  $\Gamma$  be a finite set of atomic propositions. A *timed word*  $w$  over the set  $\Gamma$  is an infinite sequence  $w^t = (w(0), \tau(0))(w(1), \tau(1))\dots$  where  $w(0) w(1) \dots$  is an infinite word over the set  $2^\Gamma$  and  $\tau(0) \tau(1) \dots$  is a time sequence with  $\tau(l) \in \mathbb{Q}_+$ ,  $l \in \mathbb{N}$ .

**Definition 7** A *Weighted Transition System (WTS)* is a tuple  $(S, S^{\text{init}}, Act, \longrightarrow, \mathfrak{t}, \Gamma, \mathcal{L})$  where  $S$  is a finite set of states;  $S^{\text{init}} \subseteq S$  is a set of initial states;  $Act$  is a set of actions;  $\longrightarrow \subseteq S \times Act \times S$  is a transition relation;  $\mathfrak{t} : \longrightarrow \rightarrow \mathbb{Q}_+$  is a map that assigns a positive weight to each transition;  $\Gamma$  is a finite set of atomic propositions; and  $\mathcal{L} : S \rightarrow 2^\Gamma$  is a labeling function.

**Definition 8** 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  $l \geq 1$ , it holds that  $r(l) \in S$  and  $(r(l), \alpha(l), r(l+1)) \in \longrightarrow$  for a sequence of actions  $\alpha(1)\alpha(2)\dots$  with  $\alpha(l) \in Act, \forall l \geq 1$ . The *time stamps*  $\tau(l), l \geq 0$  are inductively defined as: 1)  $\tau(0) = 0$ ; 2)  $\tau(l+1) = \tau(l) + \mathfrak{t}(r(l), \alpha(l), r(l+1)), \forall l \geq 1$ . Every timed run  $r^t$  generates a *timed word*  $w^t = (w(0), \tau(0))(w(1), \tau(1))\dots$  over the set  $2^\Gamma$  where  $w(l) = L(r(l)), \forall l \in \mathbb{N}$  is the subset of atomic propositions that are true at state  $r(l)$ .

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

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \diamond_I \varphi \mid \square_I \varphi \mid \varphi_1 \mathcal{U}_I \varphi_2,$$

where  $p \in \Gamma$ , and  $\diamond, \square$  and  $\mathcal{U}$  are the eventually, always and until temporal operator, respectively;  $I = [a, b] \subseteq \mathbb{Q}_+$  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 Definition 8.

**Definition 9** ([23], [24]) Given a timed word  $w^t = (w(0), \tau(0))(w(1), \tau(1))\dots$ , an MITL formula  $\varphi$  and a position  $l$  in the timed word, the satisfaction relation  $(w^t, l) \models \varphi$ , for  $l \geq 0$  (read  $w^t$  satisfies  $\varphi$  at position  $l$ )

is inductively defined as follows:

$$\begin{aligned}
(w^t, l) &\models p \Leftrightarrow p \in w(l), \\
(w^t, l) &\models \neg\varphi \Leftrightarrow (w^t, l) \not\models \varphi, \\
(w^t, l) &\models \varphi_1 \wedge \varphi_2 \Leftrightarrow (w^t, l) \models \varphi_1 \text{ and } (w^t, l) \models \varphi_2, \\
(w^t, l) &\models \diamond_I \varphi \Leftrightarrow \exists l' \geq l, \text{ such that} \\
&\quad (w^t, l') \models \varphi, \tau(l') - \tau(l) \in I, \\
(w^t, l) &\models \square_I \varphi \Leftrightarrow \forall l' \geq l, \tau(l') - \tau(l) \in I \\
&\quad \Rightarrow (w^t, l') \models \varphi, \\
(w^t, l) &\models \varphi_1 \mathcal{U}_I \varphi_2 \Leftrightarrow \exists l' \geq l, \text{ s.t. } (w^t, l') \models \varphi_2, \\
&\quad \tau(l') - \tau(l) \in I \text{ and } (w^t, l'') \models \varphi_1, \forall l \leq l'' < l'.
\end{aligned}$$

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

*Timed Büchi Automata (TBA)* were originally introduced in [12, 25, 26]. Let  $\text{CL} = \{c_1, \dots, c_{|\text{CL}|}\}$  be a finite set of *clocks*. The set of *clock constraints*  $\Phi(\text{CL})$  is defined by the grammar:

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

where  $c \in \text{CL}$  is a clock,  $\psi \in \mathbb{Q}_+$  is a clock constant and  $\bowtie \in \{<, >, \geq, \leq, =\}$ . A clock *valuation* is a function  $\nu : \text{CL} \rightarrow \mathbb{Q}_+$  that assigns a value to each clock. A clock  $c_l$  has valuation  $\nu_l$  for  $l \in \{1, \dots, |\text{CL}|\}$ , and  $\nu = (\nu_1, \dots, \nu_{|\text{CL}|})$ . We denote by  $\nu \models \phi$  the fact that the valuation  $\nu$  satisfies the clock constraint  $\phi$ .

**Definition 10** A *Timed Büchi Automaton* is a tuple  $(Q, Q^{\text{init}}, \text{CL}, \text{Inv}, E, \text{FS}, \Gamma, \mathcal{L})$  where  $Q$  is a finite set of locations;  $Q^{\text{init}} \subseteq Q$  is the set of initial locations;  $\text{CL}$  is a finite set of clocks;  $\text{Inv} : Q \rightarrow \Phi(\text{CL})$  is the invariant;  $E \subseteq Q \times \Phi(\text{CL}) \times 2^{\text{CL}} \times Q$  gives the set of edges of the form  $e = (q, \mathbf{g}, \text{RS}, q')$ , where  $q, q'$  are the source and target states,  $\mathbf{g}$  is the guard of edge  $e$  and  $\text{RS}$  is a set of clocks to be reset upon executing the edge;  $\text{FS} \subseteq Q$  is a set of accepting locations;  $\Gamma$  is a finite set of atomic propositions; and  $\mathcal{L} : Q \rightarrow 2^\Gamma$  labels every state with a subset of atomic propositions.

Any MITL formula  $\varphi$  over  $\Gamma$  can be algorithmically translated into a TBA with the alphabet  $2^\Gamma$ , such that the language of timed words (i.e. the set of all accepted timed words) that satisfy  $\varphi$  is the language of timed words produced by the TBA ([27–30]).

**Definition 11** Given a WTS  $\mathcal{T} = (S, S^{\text{init}}, \text{Act}, \longrightarrow, \mathbf{t}, \Gamma, \mathcal{L})$ , and a TBA  $\mathcal{A} = (Q, Q^{\text{init}}, \text{CL}, \text{Inv}, E, \text{FS}, \Gamma)$  with  $\text{CL}$  clocks. Then, their *product WTS*:

$$\tilde{\mathcal{T}} = \mathcal{T} \otimes \mathcal{A} = (\tilde{Q}, \tilde{Q}^{\text{init}}, \rightsquigarrow, \tilde{\mathbf{t}}, \tilde{\text{F}}, \Gamma, \tilde{\mathcal{L}}),$$

is defined as follows:

- $\tilde{Q} = S \times Q$  is the set of states;
- $\tilde{Q}^{\text{init}} = S^{\text{init}} \times Q^{\text{init}}$  is the set of initial states;
- $\rightsquigarrow$  is the set of transitions where  $(\tilde{q}, \tilde{q}') \in \rightsquigarrow$  iff:
  - $\tilde{q} = (s, q) \in \tilde{Q}$  and  $\tilde{q}' = (s', q') \in \tilde{Q}$ ,
  - $(s, \cdot, s') \in \longrightarrow$ , and
  - there exist  $\mathbf{g}, \gamma, \text{RS}$  such that  $(q, \mathbf{g}, \text{RS}, \gamma, q') \in E$  where  $\gamma = \mathcal{L}(q')$ ;
- $\tilde{\mathbf{t}}(q, q') = \mathbf{t}(s, s')$  if  $(q, q') \in \rightsquigarrow$ , is a map that assigns a positive weight to each transition;
- $\tilde{\text{F}} = \{(s, q) \in \tilde{Q} : q \in \text{FS}\}$  is a set of accepting states; and
- $\tilde{\mathcal{L}}(s, q) = \mathcal{L}(s)$  is a labeling function.

### 3 Problem Formulation

#### 3.1 System Model

Consider a team of  $N$  robots with labels  $[N] := \{1, \dots, N\}$  operating in a bounded workspace  $\mathcal{W} \subseteq \mathbb{R}^n$ . The robots are governed by the following kinematics and dynamics model:

$$\dot{x}_i = v_i, \tag{1a}$$

$$\dot{v}_i = f_i(x_i, v_i) + G_i u_i + \delta_i, \tag{1b}$$

where  $x_i, v_i \in \mathbb{R}^n$  stands for the position/orientation and the linear/angular velocity of the robot  $i \in [N]$ , respectively;  $f_i : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a known and continuously differentiable vector fields with  $f_i(0, 0) = 0$  and  $G_i \in \mathbb{R}^{n \times n}$ ;  $u_i \in \mathbb{R}^n$  stands for the control input vector; and  $\delta_i \in \mathbb{R}^n$  models the external disturbances and uncertainties. Consider also velocity constraints, input constraints as well as bounded disturbances:

$$v_i \in \mathcal{V}_i := \{v_i \in \mathbb{R}^n : \|v_i\|_2 \leq \tilde{v}_i\},$$

$$u_i \in \mathcal{U}_i := \{u_i \in \mathbb{R}^n : \|u_i\|_2 \leq \tilde{u}_i\},$$

$$\delta_i \in \Delta_i := \{\delta_i \in \mathbb{R}^n : \|\delta_i\|_2 \leq \tilde{\delta}_i\},$$

where the constants  $\tilde{v}_i, \tilde{u}_i, \tilde{\delta}_i > 0$  are a priori given. The sets  $\mathcal{V}_i, \mathcal{U}_i$  and  $\Delta_i$  are assumed to be connected sets with the origin as an interior point. Define the corresponding *nominal kinematics/dynamics* by:

$$\dot{\bar{x}}_i = v_i, \tag{2a}$$

$$\dot{\bar{v}}_i = f_i(\bar{x}_i, \bar{v}_i) + G_i \bar{u}_i, \tag{2b}$$

which are the real kinematics/dynamics for the case of  $\delta_i = 0$ .

**Assumption 1** *The linear systems  $\dot{\eta}_i = A_i \eta_i + B_i \bar{u}_i$ , where  $\bar{\eta}_i := [\bar{x}_i^\top, \bar{v}_i^\top]^\top \in \mathbb{R}^{2n}$ , that are the outcome of the Jacobian linearization of the nominal dynamics (2a)-(2b) around the equilibrium states  $(x_i, v_i) = (0, 0)$  are stabilizable.*

**Assumption 2** *There exist strictly positive constants  $\underline{G}_i$  such that:*

$$\lambda_{\min} \left[ \frac{G_i + G_i^\top}{2} \right] \geq \underline{G}_i > 0, \quad \forall i \in \mathcal{V}. \quad (3)$$

*Remark 1* Assumption 1 is required for the NMPC nominal stability to be guaranteed [22]. Note also that in real-time robotic systems, the matrices  $G_i$  usually represents the mass matrix of the robots which are always positive-definite. Thus, Assumption 2 is satisfied.

In the given workspace, there exist  $Z \in \mathbb{N}$  disjoint Regions of Interest (RoI) labeled by  $[Z] := \{1, \dots, Z\}$ . We assume that the RoI are modeled by balls, i.e.,  $\mathcal{R}_z := \mathcal{M}(y_z, p_z)$ ,  $Z \in \mathbb{N}$ , where  $y_z$  and  $p_z > 0$  stand for the center and radius of RoI  $\mathcal{R}_z$ , respectively. Define also the union of RoI by

$$\mathcal{R} := \bigcup_{z \in [Z]} \mathcal{R}_z.$$

Due to the fact that we are interested in imposing safety constraints, at each time  $t \geq 0$ , the robot  $i$  is occupying a ball  $\mathcal{M}(x_i(t), \mathbf{r}_i)$  that covers its volume, where  $x_i(t)$  and  $\mathbf{r}_i > 0$  are its center and radius, respectively. Moreover, in order to be able to impose transient constraints among the robots, we assume that each robot  $i \in [N]$  has communication capabilities within a limited sensing range  $d_i > 0$  such that:

$$d_i > \max_{i,j \in [N], i \neq j} \{\mathbf{r}_i + \mathbf{r}_j\}, \quad (4)$$

The latter implies that each agent has sufficiently large sensing radius so as to measure the agent with the biggest volume, due to the fact that the agents' radii are not the same. Define the set of robots  $j$  that are within the sensing range of agent  $i$  at time  $t$  as:

$$\mathcal{G}_i(t) := \{j \in [N] \setminus \{i\} : \|x_i(t) - x_j(t)\|_2 < d_i\}. \quad (5)$$

### 3.2 Objectives

The goal of this paper is to design decentralized feedback control laws that steers the robots with dynamics as in (1a)-(1b) between RoI so that they obey individual high-level tasks given in MITL under transient constraints between them. Define the labeling functions:

$$\mathcal{L}_i : \mathcal{R} \rightarrow 2^{\mathcal{F}_i}, \quad (6)$$

which map each RoI with a subset of atomic propositions that hold true there. Note that some of the RoI may be assigned with labels that indicate *unsafe regions*, i.e., the robot is required to avoid visiting them (*safety specifications*).

**Definition 12** A trajectory  $x_i(t)$  of robot  $i \in [N]$  is associated with a timed run  $r_i^t = (r_i(0), \tau_i(0)) (r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \dots$ , where  $r_i(l) \in \mathcal{R}$ ,  $\forall l \in \mathbb{N}$ , is a sequence of RoI that the robot crosses, if the following hold:

1.  $\tau_i(0) = 0$ , i.e., the robot starts the motion at time  $t = 0$ ;
2.  $\mathcal{M}(x_i(\tau_i(0)), \mathbf{r}_i) \subsetneq r_i(0)$ , i.e., initially, the volume of the robot is entirely within the RoI  $r_i(0) \in \mathcal{R}$ ;
3.  $\mathcal{M}(x_i(\tau_i(l)), \mathbf{r}_i) \subsetneq r_i(l)$ ,  $\forall l \in \mathbb{N}$ , i.e., the robot changes discrete state *as soon as* its entire volume is strictly contained in the corresponding RoI;
4.  $\tau_i(l+1) := \tau_i(l) + \mathbf{t}_i(r_i(l), r_i(l+1))$ ,  $\forall l \in \mathbb{N}$ , where:

$$\mathbf{t}_i : \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{Q}_+, \quad (7)$$

are functions that model the duration that the robot needs to be driven between regions  $r_i(l)$  and  $r_i(l+1)$ .

**Definition 13** A trajectory  $x_i(t)$  satisfies an MITL formula  $\varphi_i$  over the set of atomic propositions  $\Gamma_i$ , formally written as  $x_i(t) \models \varphi_i$ ,  $\forall t \geq 0$ , if and only if there exists a timed run  $r_i^t$  to which the trajectory  $x_i(t)$  is associated, according to Definition 12, which satisfies  $\varphi_i$ .

*Remark 2* We assume that the volume of each robot is covered by a ball. We further assume that the obstacles can be modeled by RoI that are also balls. Even if the volume of an agent and/or an obstacle is not a ball, it can be over-approximated by a ball.

### 3.3 Problem Statement

The problem considered in this paper is stated as follows:

**Problem 1** Consider  $N$  robots governed by dynamics (1a)-(1b), covered by the balls  $\mathcal{M}(x_i(t), \mathbf{r}_i)$ , operating in the workspace  $\mathcal{W} \subseteq \mathbb{R}^n$  with sensing communication capabilities captured by the sets  $\mathcal{G}_i$  as defined in (5). The workspace contains the RoI  $\mathcal{R}_z$ ,  $z \in [Z]$  modeled also by balls. Given task specification formulas  $\varphi_i$  for each robot  $i \in [N]$  expressed in MITL over the set of atomic propositions  $\Gamma_i$  and labeling functions  $\mathcal{L}_i$  as in (6); then, design decentralized feedback control laws  $u_i = \kappa_i(x_i, v_i) \in \mathcal{U}_i$ , such that the robot trajectories in the workspace fulfill the MITL specifications  $\varphi_i$ , i.e.,  $x_i(t) \models \varphi_i$ ,  $\forall t \geq 0$ , according to Definition 12, while collision avoidance constraints are imposed among the robots, i.e.:

$$\|x_i - x_j\|_2 > \mathbf{r}_i + \mathbf{r}_j, \quad \forall i \in [N], j \in [N] \setminus \{i\}.$$

*Remark 3* Note that Problem 1 constitutes a general problem due to the fact that the dynamics (1a)-(1b) arise in most robotic applications and transient constraints among the robots are taken into consideration.

#### 4 Problem Solution

In this section, a systematic framework for solving Problem 1 is provided as follows:

1. In Sections 4.1-4.2, decentralized feedback control laws that guarantee the transition between RoI in the given environments are provided. The laws consist of two components: an online control law which is the outcome of a DFHOCP solved at each timed step (Section 4.1); and an offline law which guarantees that the trajectories of the real system remain in a hyper-tube (Section 4.2).
2. Then, by using the outcome of Section 4.1, we abstract the dynamics (1a)-(1b) into a WTS for each robot, exploiting the fact that the timed runs in the WTS project onto associated trajectories according to Definition 12. (Section 4.3)
3. By invoking ideas from our previous work [13], a controller synthesis procedure that provides a sequence of control laws that serve as solution to Problem 1 is consulted. (Section 4.4)
4. Lastly, the computational complexity of the proposed framework is discussed in Section 4.5.

##### 4.1 Decentralized Feedback Control Design - Part I

Consider the robot  $i$  with dynamics (1a)-(1b) occupying a RoI  $\mathcal{R}_{i,s} \in \mathcal{R}$  at time  $t_{i,s} \geq 0$ . The decentralized feedback control should guarantee that the robot is navigated towards a desired RoI  $\mathcal{R}_{i,d} \in \mathcal{R}$ ,  $\mathcal{R}_{i,d} \neq \mathcal{R}_{i,s}$  without intersection with any other RoI or other agents  $j \in [N]$ ,  $j \neq i$ . Denote by  $x_{i,d} \in \mathcal{R}_{i,d}$  the center of the RoI  $\mathcal{R}_{i,s}$ . Define the error vector:

$$e_i := x_i - x_{i,d} \in \mathbb{R}^n, \quad i \in [N],$$

as well as the *uncertain error kinematics/dynamics* by:

$$\dot{e}_i = v_i, \quad (8a)$$

$$\dot{v}_i = f_i(e_i + x_{i,d}, v_i) + G_i u_i + \delta_i, \quad (8b)$$

The corresponding *uncertain nominal error kinematics/dynamics* are given by:

$$\dot{\bar{e}}_i = \bar{v}_i, \quad (9a)$$

$$\dot{\bar{v}}_i = f_i(\bar{e}_i + x_{i,d}, \bar{v}_i) + G_i \bar{u}_i. \quad (9b)$$

Consider the feedback control law:

$$u_i := \bar{u}_i(\bar{e}_i, \bar{v}_i) + \kappa_i(e_i, v_i, \bar{e}_i, \bar{v}_i), \quad (10)$$

which consists of a nominal control action  $\bar{u}_i(\bar{e}_i, \bar{v}_i) \in \mathcal{U}_i$  and a state feedback laws  $\kappa_i(e_i, v_i, \bar{e}_i, \bar{v}_i)$ . The control action  $\bar{u}_i(\bar{e}_i, \bar{v}_i)$  is the outcome of a DFHOCP solved on-line at each sampling time step; the state-feedback law will be tuned off-line according to a procedure that will be presented thereafter.

Define the sets that capture the state constraints of each robot as:

$$\begin{aligned} \mathcal{E}_i := & \left\{ e_i(t) \in \mathbb{R}^n : \|e_i(t) + x_{i,d} - e_j(t) - x_{j,d}\|_2 > \right. \\ & \left. \mathfrak{r}_i + \mathfrak{r}_j + \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad \forall j \in \mathcal{G}_i(t), \right. \\ & \left. \mathcal{M}(e_i(t) + x_{i,d}, \mathfrak{r}_i) \cap \{\mathcal{R} \setminus \{\mathcal{R}_{i,s}, \mathcal{R}_{i,d}\}\} = \emptyset \right\}. \end{aligned}$$

The first constraint captures the fact that the robots should not collide with each other; the latter one, captures the fact that each robot needs to be navigated from RoI  $\mathcal{R}_{i,s}$  to RoI  $\mathcal{R}_{i,d}$  without intersecting with any other RoI of the workspace due to the fact that we are interested in imposing safety specifications.

**Assumption 3** *It is assumed that:*

$$\mathfrak{r}_i + \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}} < p_z, \quad \forall z \in [Z], \quad i \in [N]. \quad (11)$$

More specifically, (11) states that the radius of the ball that covers every robot plus the radius of the disturbance tube is smaller than the radius of any of the RoI of the workspace. As it will be shown later, this assumption is required in order to compute the time that a robot needs to be navigated between the RoI in the workspace.

Consider a sequence of sampling times  $\{t_k\}$ ,  $k \in \mathbb{N}$ , with a constant sampling period  $0 < h < T$ , where  $T$  stands for the finite prediction horizon. It holds that  $t_{k+1} = t_k + h$ ,  $\forall k \in \mathbb{N}$ . It should be noted that both  $t_k$  and  $T$  are multiples of  $h$ . At every discrete sampling time  $t_k$  a DFHOCP is solved by each robot  $i \in [N]$  as follows:

$$\min_{\bar{u}_i(\cdot)} \left\{ \|\bar{\xi}_i(t_k + T)\|_{P_i}^2 + \int_{t_k}^{t_k+T} \left[ \|\bar{\xi}_i(\mathfrak{s})\|_{Q_i}^2 + \|\bar{u}_i(\mathfrak{s})\|_{R_i}^2 \right] d\mathfrak{s} \right\} \quad (12a)$$

subject to:

$$\dot{\bar{\xi}}_i(\mathfrak{s}) = \mathfrak{f}_i(\bar{e}_i(\mathfrak{s}), \bar{v}_i(\mathfrak{s}), \bar{u}_i(\mathfrak{s})), \quad (12b)$$

$$\bar{\xi}_i(\mathfrak{s}) \in \bar{\mathcal{E}}_i \times \bar{\mathcal{V}}_i, \quad \bar{u}_i(\mathfrak{s}) \in \bar{\mathcal{U}}_i, \quad \forall \mathfrak{s} \in [t_k, t_k + T], \quad (12c)$$

$$\bar{\xi}_i(t_k + T) \in \mathcal{F}_i. \quad (12d)$$

In the aforementioned optimal control problem we defined:

$$\begin{aligned}\bar{\xi}_i &:= [\bar{e}_i, \bar{v}_i]^\top \in \mathbb{R}^{2n}, \\ \mathbf{f}_i(\bar{\xi}_i, \bar{u}_i) &:= \begin{bmatrix} \bar{v}_i \\ f_i(\bar{e}_i + x_{i,d}, \bar{v}_i) + \bar{u}_i \end{bmatrix}.\end{aligned}$$

The matrices  $Q_i, P_i \in \mathbb{R}^{2n}$  and  $R_i \in \mathbb{R}^n$  are positive definite weighting matrices. The sets  $\mathcal{F}_i$  stand for the terminal sets that are used to enforce the stability of the nominal system (see [22] for more details).

Hereafter, the sets  $\bar{\mathcal{E}}_i, \bar{\mathcal{V}}_i$  and  $\bar{\mathcal{U}}_i$  are explained. In order to guarantee that while the DFHOCP (12a)-(12d) is solved for the nominal dynamics (9a)-(9b), the real states  $e_i, v_i$  and control inputs  $u_i$  satisfy the corresponding state and input constraints  $\mathcal{E}_i, \mathcal{V}_i$  and  $\mathcal{U}_i$ , respectively, the latter sets are appropriately modified as:

$$\begin{aligned}\bar{\mathcal{E}}_i &:= \mathcal{E}_i \ominus \Omega_{i,1}, \\ \bar{\mathcal{V}}_i &:= \mathcal{V}_i \ominus \Omega_{i,2}, \\ \bar{\mathcal{U}}_i &:= \mathcal{U}_i \ominus [-k_i \circ \bar{\Omega}_i],\end{aligned}$$

with  $\bar{\Omega}_i := \Omega_{i,1} \oplus \Omega_{i,2}$ ,  $\Omega_{i,1}, \Omega_{i,2}$  as given in (16a), (16b), respectively, and  $k_i$  to be defined later. This constitutes a standard constraints set modification technique adopted in tube-based NMPC frameworks (for more details see [21]). The advantage of the tube-based framework compared to other robust NMPC approaches is that the constraint tightening is performed offline and it does not depend on the length of the horizon. Algorithm 1 depicts the procedure of how the control law is calculated and applied to a real robot. This is a procedure of implementing the continuous-time tube-based NMPC in a real-time system that has been introduced in [21].

## 4.2 Decentralized Feedback Control Design - Part II

For each agent  $i \in [N]$  define by:

$$\begin{aligned}\mathbf{e}_i &:= e_i - \bar{e}_i, \\ \mathbf{v}_i &:= v_i - \bar{v}_i,\end{aligned}$$

the deviation between the real states  $e_i, v_i$  of the uncertain system (8a)-(8b) and the states of the nominal system (9a)-(9b) with  $\mathbf{e}_i(0) = \mathbf{v}_i(0) = 0$ . It will be proven that the states  $\mathbf{e}_i, \mathbf{v}_i$  remain invariant in certain compact sets. The dynamics of the states  $\mathbf{e}_i$  and  $\mathbf{v}_i$  are

---

### Algorithm 1 Implementation of feedback control laws $u_i(t), i \in [N]$

---

**Step 0 :** At time  $t_0 := 0$ , set  $e_i(0) = \bar{e}_i(0), v_i(0) = \bar{v}_i(0)$ .  
**Step 1 :** At time  $t_k$  and current state  $(e_i(t_k), v_i(t_k), \bar{e}_i(t_k), \bar{v}_i(t_k))$ , solve the DFHOCP (12a)-(12d) to obtain the nominal control action  $\bar{u}_i(t_k)$  and the actual control action

$$u_i(t_k) = \bar{u}_i(t_k) + \kappa_i(e_i(t_k), \bar{e}_i(t_k), v_i(t_k), \bar{v}_i(t_k)).$$

**Step 2 :** Apply the control  $u_i(t_k)$  to the system (8a)-(8b), during sampling interval  $[t_k, t_{k+1})$ , where  $t_{k+1} = t_k + h$ .

**Step 3 :** Measure the state  $(e_i(t_{k+1}), v_i(t_{k+1}))$  at the next time instant  $t_{k+1}$  of the system (8a)-(8b) and compute the successor state  $(\bar{e}_i(t_{k+1}), \bar{v}_i(t_{k+1}))$  of the nominal system (9a)-(9b) under the nominal control action  $\bar{u}_i(t_k)$ .

**Step 4 :** Set

$$\begin{aligned}(e_i(t_k), \bar{e}_i(t_k), v_i(t_k), \bar{v}_i(t_k)) &\leftarrow (e_i(t_{k+1}), \\ &\bar{e}_i(t_{k+1}), v_i(t_{k+1}), \bar{v}_i(t_{k+1})), \\ t_k &\leftarrow t_{k+1};\end{aligned}$$

**Go to Step 1.**

---

written as:

$$\begin{aligned}\dot{\mathbf{e}}_i &= \dot{e}_i - \dot{\bar{e}}_i \\ &= v_i - \bar{v}_i \\ &= \mathbf{v}_i,\end{aligned}\tag{13a}$$

$$\begin{aligned}\dot{\mathbf{v}}_i &= \dot{v}_i - \dot{\bar{v}}_i \\ &= f_i(e_i + x_{i,d}, v_i) - f_i(\bar{e}_i + x_{i,d}, \bar{v}_i) \\ &\quad + G_i u_i - G_i \bar{u}_i + \delta_i \\ &= g(e_i, \bar{e}_i, v_i, \bar{v}_i) + G_i (u_i - \bar{u}_i) + \delta_i,\end{aligned}\tag{13b}$$

where the functions  $g_i$  are defined by:

$$g_i(e_i, \bar{e}_i, v_i, \bar{v}_i) := f_i(e_i + x_{i,d}, v_i) - f_i(\bar{e}_i + x_{i,d}, \bar{v}_i),$$

and they are upper bounded by:

$$\begin{aligned}\|g_i(e_i, \bar{e}_i, v_i, \bar{v}_i)\|_2 &\leq \|f_i(e_i + x_{i,d}, v_i) - f_i(\bar{e}_i + x_{i,d}, \bar{v}_i)\|_2 \\ &= \|f_i(e_i + x_{i,d}, v_i) - f_i(e_i + x_{i,d}, \bar{v}_i) \\ &\quad + f_i(e_i + x_{i,d}, \bar{v}_i) - f_i(\bar{e}_i + x_{i,d}, \bar{v}_i)\|_2 \\ &\leq \|f_i(e_i + x_{i,d}, v_i) - f_i(e_i + x_{i,d}, \bar{v}_i)\|_2 \\ &\quad + \|f_i(e_i + x_{i,d}, \bar{v}_i) - f_i(\bar{e}_i + x_{i,d}, \bar{v}_i)\|_2 \\ &\leq L_{v,i} \|v_i - \bar{v}_i\|_2 + L_{e,i} \|e_i + x_{i,d} - \bar{e}_i - x_{i,d}\|_2 \\ &= L_{v,i} \|v_i - \bar{v}_i\|_2 + L_{e,i} \|e_i - \bar{e}_i\|_2 \\ &= L_{v,i} \|\mathbf{v}_i\|_2 + L_{e,i} \|\mathbf{e}_i\|_2 \\ &\leq L_i (\|\mathbf{e}_i\|_2 + \|\mathbf{v}_i\|_2).\end{aligned}$$

The constants  $L_{e,i}, L_{v,i}$  stand for the Lipschitz constants of functions  $f_i$  with respect to the variable  $e_i$  and  $v_i$ , respectively, and

$$L_i := \max\{L_{e,i}, L_{v,i}\}, \quad i \in [N].$$



**Lemma 2** *The state feedback laws designed by:*

$$\kappa_i(e_i, \bar{e}_i, v_i, \bar{v}_i) := -k_i(e_i - \bar{e}_i) - k_i(v_i - \bar{v}_i), \quad i \in [N], \quad (14)$$

where  $k_i, \rho_i > 0$  are chosen such that the following hold:

$$k_i > \frac{1}{\underline{G}_i} [1 + (1 + 2\rho_i)L_i], \quad \rho_i > \frac{L_i}{2}, \quad (15)$$

renders the sets:

$$\Omega_{i,1} := \left\{ \mathbf{e}_i \in \mathbb{R}^n : \|\mathbf{e}_i\|_2 \leq \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}} \right\}, \quad (16a)$$

$$\Omega_{i,2} := \left\{ \mathbf{v}_i \in \mathbb{R}^n : \|\mathbf{v}_i\|_2 \leq \frac{2\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}} \right\}, \quad (16b)$$

RCI sets for the error dynamics (13a), (13b), according to Definition 3, where the constants  $\alpha_{i,1}, \alpha_{i,2} > 0$  are defined by:

$$\alpha_{i,1} := 1 - \frac{L_i}{2\rho_i}, \quad \alpha_{i,2} := k_i \underline{G}_i - 1 - (1 + 2\rho_i)L_i. \quad (17)$$

*Proof* A backstepping control methodology will be used [31]. The state  $\mathbf{v}_i$  in (13b) can be seen as virtual input to be designed such that the candidate Lyapunov function:

$$\mathfrak{L}_1(\mathbf{e}_i) := \frac{1}{2} \|\mathbf{e}_i\|_2^2,$$

for the dynamical system (13a) is always decreasing. The time derivative of  $\mathfrak{L}_1$  along the trajectories of system (13a) is given by:

$$\dot{\mathfrak{L}}_1(\mathbf{e}_i) = \mathbf{e}_i^\top \dot{\mathbf{e}}_i = \mathbf{e}_i^\top \mathbf{v}_i.$$

Thus, by designing  $\mathbf{v}_i \equiv -\mathbf{e}_i$ , it yields that

$$\dot{\mathfrak{L}}_1(\mathbf{e}_i) = -\|\mathbf{e}_i\|_2^2.$$

Define the backstepping auxiliary errors  $\zeta_{i,1}, \zeta_{i,2} \in \mathbb{R}^n$  by:

$$\begin{aligned} \zeta_{i,1} &:= \mathbf{e}_i, \\ \zeta_{i,2} &:= \mathbf{v}_i + \mathbf{e}_i. \end{aligned}$$

Then, the auxiliary error dynamics are written as:

$$\dot{\zeta}_{i,1} = \dot{\mathbf{e}}_i = \mathbf{v}_i = \zeta_{i,2} - \mathbf{e}_i = -\zeta_{i,1} + \zeta_{i,2} \quad (18a)$$

$$\dot{\zeta}_{i,2} = -\zeta_{i,1} + \zeta_{i,2} + g_i(\cdot) + G_i(u_i - \bar{u}_i) + \delta_i, \quad (18b)$$

with:

$$\begin{aligned} \|g_i(\cdot)\|_2 &\leq L_i (\|\mathbf{e}_i\|_2 + \|\mathbf{v}_i\|_2) \\ &= L_i (\|\zeta_{i,1}\|_2 + \|\zeta_{i,1} - \zeta_{i,2}\|_2) \\ &\leq 2L_i \|\zeta_{i,1}\|_2 + L_i \|\zeta_{i,2}\|_2, \end{aligned} \quad (18c)$$

and  $\zeta_{i,1}(0) = \zeta_{i,2}(0) = 0$ . Define the stack vector  $\zeta_i := [\zeta_{i,1}^\top, \zeta_{i,2}^\top]^\top \in \mathbb{R}^{2n}$  and consider the candidate Lyapunov

function  $\mathfrak{L}(\zeta_i) = \frac{1}{2} \|\zeta_i\|_2^2$  with  $\mathfrak{L}(0) = 0$ . The time derivative of  $\mathfrak{L}$  along the trajectories of system (13a)-(13b) is given by:

$$\begin{aligned} \dot{\mathfrak{L}}(\zeta_i) &= \zeta_i^\top \dot{\zeta}_i \\ &= \zeta_{i,1}^\top \dot{\zeta}_{i,1} + \zeta_{i,2}^\top \dot{\zeta}_{i,2} \\ &= -\|\zeta_{i,1}\|_2^2 + \|\zeta_{i,2}\|_2^2 + \zeta_{i,2}^\top g_i(\cdot) \\ &\quad + \zeta_{i,2}^\top G_i(u_i - \bar{u}_i) + \zeta_{i,2}^\top \delta_i \\ &\leq -\|\zeta_{i,1}\|_2^2 + \|\zeta_{i,2}\|_2^2 + \|\zeta_{i,2}\|_2 \|g_i(\cdot)\|_2 \\ &\quad + \zeta_{i,2}^\top G_i(u_i - \bar{u}_i) + \zeta_{i,2}^\top \delta_i \end{aligned}$$

By using (18c), the latter becomes:

$$\begin{aligned} \dot{\mathfrak{L}}(\zeta_i) &= -\|\zeta_{i,1}\|_2^2 + (L_i + 1)\|\zeta_{i,2}\|_2^2 + 2L_i \|\zeta_{i,1}\|_2 \|\zeta_{i,2}\|_2 \\ &\quad + \zeta_{i,2}^\top G_i(u_i - \bar{u}_i) + \zeta_{i,2}^\top \delta_i. \end{aligned}$$

By using Lemma 1 we have:

$$\begin{aligned} \|\zeta_{i,1}\|_2 \|\zeta_{i,2}\|_2 &\leq \frac{\|\zeta_{i,1}\|_2^2}{4\rho_i} + \rho_i \|\zeta_{i,2}\|_2^2 \\ \Rightarrow 2L_i \|\zeta_{i,1}\|_2 \|\zeta_{i,2}\|_2 &\leq \frac{L_i \|\zeta_{i,1}\|_2^2}{2\rho_i} + 2\rho_i L_i \|\zeta_{i,2}\|_2^2. \end{aligned}$$

with  $\rho_i$  satisfying (15). Then, it holds that:

$$\begin{aligned} \dot{\mathfrak{L}}(\zeta_i) &\leq -\left(1 - \frac{L_i}{2\rho_i}\right) \|\zeta_{i,1}\|_2^2 + [1 + (1 + 2\rho_i)L_i] \|\zeta_{i,2}\|_2^2 \\ &\quad + \zeta_{i,2}^\top G_i(u_i - \bar{u}_i) + \|\zeta_{i,2}\|_2 \tilde{\delta}_i. \end{aligned}$$

By designing  $u_i - \bar{u}_i = -k_i \zeta_{i,2} = -k_i \mathbf{e}_i - k_i \mathbf{v}_i = -k_i(e_i - \bar{e}_i) - k_i(v_i - \bar{v}_i)$  which is the same with (14) and compatible with (10) we get:

$$\begin{aligned} \dot{\mathfrak{L}}(\zeta_i) &\leq -\left(1 - \frac{L_i}{2\rho_i}\right) \|\zeta_{i,1}\|_2^2 + [1 + (1 + 2\rho_i)L_i] \|\zeta_{i,2}\|_2^2 \\ &\quad - k_i \zeta_{i,2}^\top G_i \zeta_{i,2} + \|\zeta_{i,2}\|_2 \tilde{\delta}_i. \end{aligned}$$

Writing the matrices  $G_i$  as  $G_i = \frac{G_i + G_i^\top}{2} + \frac{G_i - G_i^\top}{2}$  and taking into account that:

$$y^\top \left( \frac{G_i - G_i^\top}{2} \right) y = 0, \quad \forall y \in \mathbb{R}^n,$$

$$y^\top P y \geq \lambda_{\min}(P) \|y\|_2^2, \quad \forall y \in \mathbb{R}^n, \quad P \in \mathbb{R}^{n \times n}, \quad P > 0,$$

we obtain:

$$\begin{aligned} \dot{\mathfrak{L}}(\zeta_i) &\leq \\ &\quad -\left(1 - \frac{L_i}{2\rho_i}\right) \|\zeta_{i,1}\|_2^2 + [1 + (1 + 2\rho_i)L_i] \|\zeta_{i,2}\|_2^2 \\ &\quad - k_i \lambda_{\min} \left( \frac{G_i + G_i^\top}{2} \right) \|\zeta_{i,2}\|_2^2 + \|\zeta_{i,2}\|_2 \tilde{\delta}_i. \end{aligned}$$

By using Assumption 2 and (17), the latter becomes:

$$\begin{aligned}\dot{\mathcal{L}}(\zeta_i) &\leq -\alpha_{i,1}\|\zeta_{i,1}\|_2^2 - \alpha_{i,2}\|\zeta_{i,2}\|_2^2 + \|\zeta_{i,2}\|\tilde{\delta}_i \\ &\leq -\min\{\alpha_{i,1}, \alpha_{i,2}\} (\|\zeta_{i,1}\|_2^2 + \|\zeta_{i,2}\|_2^2) + \|\zeta_{i,2}\|\tilde{\delta}_i \\ &= -\min\{\alpha_{i,1}, \alpha_{i,2}\}\|\zeta_i\|_2^2 + \|\zeta_{i,2}\|\tilde{\delta}_i \\ &= \|\zeta_i\|_2 \left[ -\min\{\alpha_{i,1}, \alpha_{i,2}\}\|\zeta_i\|_2 + \tilde{\delta}_i \right]\end{aligned}$$

Thus,  $\dot{\mathcal{L}}(\zeta_i) < 0$  when  $\|\zeta_i\|_2 > \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}$ . Taking the latter into account and the fact that  $\zeta_i(0) = 0$  we have that  $\|\zeta_i(t)\|_2 \leq \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}$ ,  $\forall t \geq 0$ . Moreover, the following inequalities hold:

$$\begin{aligned}\|\mathbf{e}_i\|_2 &= \|\zeta_{i,1}\|_2 \leq \|\zeta_i\|_2 \\ \Rightarrow \|\mathbf{e}_i\|_2 &\leq \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad \forall t \geq 0,\end{aligned}$$

and:

$$\begin{aligned}|\|\mathbf{e}_i\|_2 - \|\mathbf{v}_i\|_2| &\leq \|\mathbf{e}_i + \mathbf{v}_i\|_2 \leq \|\zeta_{i,2}\|_2 = \|\zeta_i\|_2 \\ \Rightarrow \|\mathbf{v}_i\|_2 &\leq \frac{2\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad \forall t \geq 0,\end{aligned}$$

which leads to the conclusion of the proof.

The aforementioned result states that real trajectories  $e_i(t)$ ,  $v_i(t)$  will belong to a hyper-tubed which is centered along the nominal trajectories  $\bar{e}_i(t)$ ,  $\bar{v}_i(t)$ . The tubes' radii are  $\frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}$  and  $\frac{2\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}$ , respectively, as it is depicted in Fig. 2.

*Remark 4* It should be noted that the volume of the hyper-tubes depends on the upper bound of the disturbances  $\delta_i$ , the Lipschitz constants  $L_i$  and the constants  $\underline{G}_i$ . By tuning the gains  $k_i$  and  $\rho_i$  as in (15), (17) appropriately, the volume of the tubes can be adjusted. However, these gains cannot be set arbitrarily high due to the fact that the robots have limited actuation resources which are captured by the upper bound of the control input. The higher the upper bound of the control input is, the smaller the volume of the tube can be set.

By using (10), the closed-loop system is written as:

$$\dot{e}_i = v_i, \quad (19a)$$

$$\begin{aligned}\dot{v}_i &= f_i(e_i + x_{i,d}, v_i) + \bar{u}_i(\bar{e}_i, \bar{v}_i) - k_i(e_i - \bar{e}_i) \\ &\quad - k_i(v_i - \bar{v}_i) + \delta_i.\end{aligned} \quad (19b)$$

Due to the fact that Problem 1 imposes transient constraints between the agents (collision avoidance) and the agents have communication capabilities within the sensing range  $d_i$  as given in (4)-(5), we adopt here the decentralized procedure depicted in Algorithm 2 and explained hereafter. Assume that each agent knows its

labeling number in the set  $[N]$ . After each sampling time  $t_k$ ,  $\forall k \geq 0$  that agent  $i$  solves its own DFHOCP and obtains the estimated open-loop trajectory  $\bar{\xi}_i(\mathbf{s})$ ,  $\mathbf{s} \in [t_k, t_k + T]$ , it transmits it to all agents  $j \in \mathcal{G}_i(t_k)$ ,  $j \neq i$ , i.e., to agents that are within its sensing radius at time  $t_k$ . Then, agents'  $j \in \mathcal{G}_i(t_k)$ ,  $j \neq i$  hard constraints  $\bar{\mathcal{E}}_j$  are updated by incorporating the predicted trajectory of agent  $i$ , i.e.,  $\bar{\xi}_i(\mathbf{s})$ ,  $\mathbf{s} \in [t_k, t_k + T]$ . Among all agents  $j \in \mathcal{G}_i(t_k)$ , the one with higher priority, i.e., smaller labeling number in the set  $[N]$ , solves its own DFHOCP (for example, agent 2 has higher priority than agents 3, 4, ...). This *sequential procedure* is continued until all agents  $i \in [N]$  solve their own DFHOCP, and then the sampling time is updated.

In other words, each time an agent solves its own individual optimization problem, it knows the (open-loop) state predictions that have been generated by the solution of the optimization problem of all agents within its sensing range at that time, for the next  $T$  time units. These pieces of information are required, as each agent's trajectory is constrained not by constant values, but by the trajectories of its associated agents through time: at each solution time  $t_k$  and within the next  $T$  time units, an agent's predicted configuration at time  $\mathbf{s} \in [t_k, t_k + T]$  needs to be constrained by the predicted configuration of its neighboring and perceivable agents (agents within its sensing range) at the same time instant  $\mathbf{s}$ , so that collisions are avoided. We assume that the above pieces of information are *always available*, *accurate* and can be exchanged without delay. We will show thereafter that by adopting the aforementioned sequential communication procedure, and given that at  $t = 0$  the DFHOCP (12a) - (12d) of all agents are feasible, the agents are navigated to the desired RoI, while all distance and input constraints imposed by Problem 1 are satisfied.

*Remark 5* It should be noted that the constraint sets  $\bar{\mathcal{E}}_i$ ,  $i \in [N]$  in (12c) depend on the estimated open-loop trajectories  $\bar{e}_i(\mathbf{s})$  and  $\bar{v}_j(\mathbf{s})$  for all  $i \in [N]$ ,  $j \in \mathcal{G}(t_k)$ , with  $\mathbf{s} \in [t_k, t_k + T]$ . Moreover, they are updated when each robot has received the transmitted trajectories by its neighbors.

*Remark 6* By considering a real-time scenario where the state vector  $\bar{\xi}$  is comprised of 12 real numbers encoded by 4 bytes the overall downstream bandwidth required by each robot is:

$$BW_d = 12 \times 32 \text{ [bits]} \times |\mathcal{G}_i(t_k)| \times \frac{T}{h} \times f \text{ [sec}^{-1}\text{]}.$$

Given a conservative sampling time  $f = 100$  Hz and a horizon of  $\frac{T}{h} = 100$  time steps, the wireless protocol

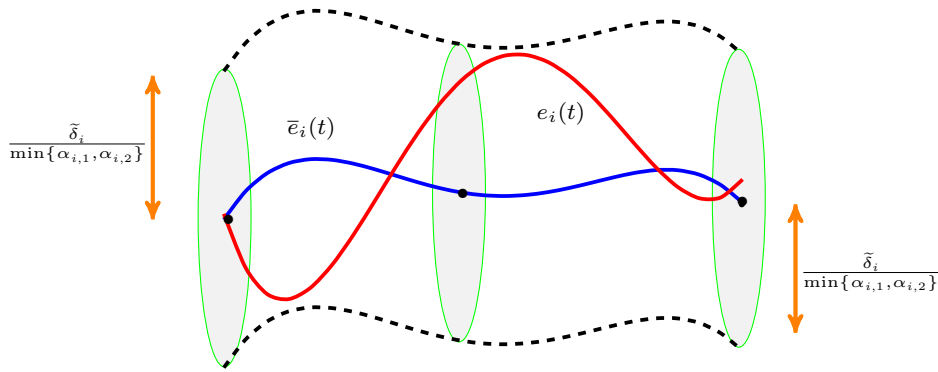


Fig. 2: The hyper-tube centered along the trajectory  $\bar{e}_i(t)$  (depicted by blue line) with radius  $\frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}$ . Under the proposed control law, the real trajectory  $e_i(t)$  (depicted with red line) lies inside the hyper-tube for all times, i.e.,  $\|e_i(t)\| \leq \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \forall t \in \mathbb{R}_{\geq 0}$ .

IEEE 802.11n-2009 (a standard for present-day devices) can accommodate up to

$$|\mathcal{G}_i(t_k)| = \frac{600 \text{ [Mbit} \cdot \text{sec}^{-1}]}{12 \times 32 \text{ [bit]} \times 10^4 \text{ [sec}^{-1}]} \approx 16 \cdot 10^2 \text{ robots,}$$

within the range of one robot. We deem this number to be large enough for practical applications of the proposed approach.

The following theorem guarantees the navigation of the agents between RoI and thereafter we will propose algorithms computing the corresponding transition times.

**Theorem 1** *Suppose that Assumptions 1-3 hold. Suppose that the robots start at time  $t_{i,s} \geq 0$  from the RoI  $\mathcal{R}_{i,s}$  and they need to be navigated to RoI  $\mathcal{R}_{i,d}$  for every  $i \in [N]$ . Suppose also that at time  $t_{i,s}$  the DFHOCP (12a)-(12d) sequentially solved by all the robots  $i \in [N]$ , is feasible. Then, the proposed decentralized feedback control law (10), (14), renders the closed-loop system (19a)-(19b) of each robot  $i \in [N]$  Input to State Stable with respect to  $\delta_i(t) \in \Delta_i$ .*

*Proof* The proof of the theorem consists of two parts:

**Feasibility Analysis:** It can be shown that recursive feasibility is established and it implies subsequent feasibility. The proof of this part is similar to the feasibility proof of [32, Theorem 2, Sec. 4, p. 12].

**Convergence Analysis:** Recall that:

$$e_i = x_i - x_{i,d}, \quad \mathbf{e}_i = e_i - \bar{e}_i, \quad \mathbf{v}_i = v_i - \bar{v}_i.$$

Then, we get:

$$\begin{aligned} \|x_i(t) - x_{i,d}\|_2 &\leq \|\bar{e}_i(t)\|_2 + \|\mathbf{e}_i(t)\|_2, \\ \|v_i(t)\|_2 &\leq \|\bar{v}_i(t)\|_2 + \|\mathbf{v}_i(t)\|_2, \end{aligned}$$

which, by using the fact that:

$$\|\bar{e}_i\|_2 \leq \|\bar{\xi}_i\|_2, \quad \|\bar{v}_i\|_2 \leq \|\bar{\xi}_i\|_2,$$

as well as the bounds from (16a), (16b), become:

$$\|x_i(t) - x_{i,d}\|_2 \leq \|\bar{\xi}_i(t)\|_2 + \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad (20a)$$

$$\|v_i(t)\|_2 \leq \|\bar{\xi}_i(t)\|_2 + \frac{2\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad \forall t \geq 0. \quad (20b)$$

Since only the nominal system dynamics (9a)-(9b) are used for the online computation of the control action  $\bar{u}_i(\mathbf{s}) \in \bar{\mathcal{U}}_i$ ,  $\mathbf{s} \in [t_k, t_k + T]$  through the DFHOCP (12a)-(12d), by invoking nominal NMPC stability results found on [22], it can be shown that there exist class  $\mathcal{KL}$  functions  $\beta_i$ , such that:

$$\|\bar{\xi}_i(t)\|_2 \leq \beta_i(\|\bar{\xi}_i(t_{i,s})\|_2, t), \quad \forall t \in \mathbb{R}_{\geq 0}. \quad (21)$$

By combining (20a)-(20b) with (21) we get:

$$\|x_i(t) - x_{i,d}\|_2 \leq \beta_i(\|\bar{\xi}_i(t_{i,s})\|_2, t) + \frac{\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad (22a)$$

$$\|v_i(t)\|_2 \leq \beta_i(\|\bar{\xi}_i(t_{i,s})\|_2, t) + \frac{2\tilde{\delta}_i}{\min\{\alpha_{i,1}, \alpha_{i,2}\}}, \quad (22b)$$

for every  $t \in \mathbb{R}_{\geq 0}$ . The latter inequalities leads to the conclusion of the proof.

### 4.3 Discrete System Abstraction

Theorem 1 implies that for each robot  $i \in [N]$  with kinematics/dynamics as in (1a), (1b), starting from the RoI  $\mathcal{R}_{i,s}$  at time  $t_{i,s}$ , is driven by the controller (10) towards a desired RoI  $\mathcal{R}_{i,d}$ , while all state, input and transient constraints are satisfied. Hereafter, we provide an algorithm for constructing the WTS of each agent. By observing (22a) and taking into account Assumption 3, it holds that there exists a time instant  $t_{i,d}$  such that the volume of robot  $i$  will be included strictly within

**Algorithm 2** Computation of  $t_{i,d} := t_i(\mathcal{R}_{i,s}, \mathcal{R}_{i,d})$ 


---

```

1: Input:  $t_{i,s}, \bar{x}_i(t_k), k \in \mathbb{N}$ ;
2: Output:  $t_{i,d}$ ;
3:  $t_k \leftarrow t_{i,s}$ ;
4:  $\text{flag} \leftarrow 1$ ;
5: while  $\text{flag} = 1$ ; do
6:   solve DFHOCP (12a)-(12d) with  $\tilde{\mathcal{E}}_i$  as in (23);
7:   measure  $\bar{x}_i(t_k)$ ;
8:   if  $\left\| \left( \|\bar{x}_i(t_k)\| + \tau_i + \frac{\tilde{\delta}_i}{\min\{\alpha_1, \alpha_2\}} \right) - x_{i,d} \right\|_2 < p_d$ 
     then
9:      $\text{flag} \leftarrow 0$ ;           {robot  $i$  is within RoI  $\mathcal{R}_{i,d}$ }
10:    break;
11:    Go to "line 15"
12:  end if
13:   $t_k \leftarrow t_k + h$ ;
14: end while
15:  $t_{i,d} \leftarrow t_k$ ;

```

---

the RoI  $\mathcal{R}_{i,d}$ . Furthermore, due to the fact that we have knowledge of the nominal dynamics and the MITL tasks  $\varphi_i$  are independent for each robot, for the computation of the time  $t_{i,d}$  an offline computer simulation of the DFHOCP (12a)-(12d) with state constraints as:

$$\tilde{\mathcal{E}}_i := \left\{ e_i(t) \in \mathbb{R}^n : \mathcal{M}(e_i(t) + x_{i,d}, \tau_i) \cap \{\mathcal{R} \setminus \{\mathcal{R}_{i,s}, \mathcal{R}_{i,d}\}\} \right\} \ominus \Omega_{i,1}, \quad (23)$$

is conducted. In particular, (23) captures constraints regarding the navigation of robot  $i$  from RoI  $\mathcal{R}_{i,s}$  to RoI  $\mathcal{R}_{i,d}$  without intersecting with any other RoI of the workspace. It should be noted that if any collision is about to occur in real-time when the robots are executing the on-line control actions, the transition time between the RoI will be different. In order to overcome the aforementioned issue, we will provide thereafter an algorithm that monitors the collision offline and updates the transition times appropriately. Then, the process of computing  $t_{i,d}$  is described in Algorithm 3. The abstraction that captures the dynamics of each robot into a WTS is given through the following definition.

**Definition 14** The motion of robot  $i$  in the workspace  $\mathcal{W}$  is modeled by the WTS

$$\mathcal{T}_i = (S_i, S_i^{\text{init}}, \text{Act}_i, \longrightarrow, t_i, \Gamma_i, L_i),$$

where:

- $S_i = \mathcal{R} = \bigcup_{z \in [Z]} \mathcal{R}_z$  is the set of states of the robot that contains all the RoI of the workspace  $\mathcal{W}$ ;
- $S_i^{\text{init}} \subseteq S_i$  is a set of initial states defined by the robot's initial position  $x_i(0)$  in the workspace;
- $\text{Act}_i$  is the set of actions containing the union of all feedback controllers (10) which can navigate the robot  $i$  between RoI;

- $\longrightarrow_i \subseteq S_i \times \text{Act}_i \times S_i$  is the transition relation. We say that  $(\mathcal{R}_{i,s}, u_i, \mathcal{R}_{i,d}) \in \longrightarrow_i$ , with  $\mathcal{R}_{i,s}, \mathcal{R}_{i,d} \in \mathcal{R}$  with  $\mathcal{R}_{i,s} \neq \mathcal{R}_{i,d}$  if there exist feedback control law  $u_i \in \text{Act}_i$  as in (10) which can drive the robot from the region  $\mathcal{R}_{i,s}$  to the region  $\mathcal{R}_{i,d}$  without intersecting with any other RoI of the workspace;
- $t_i$  is the time weight as given in (7) and it is computed by Algorithm 2;
- $\mathcal{L}_i$  is the labeling function as given in (6);
- and  $\Gamma_i$  is the set of atomic propositions imposed by Problem 1.

The aforementioned WTS of each robot allows us to work directly at the discrete level and design a sequence of feedback controllers as in (10) that solve Problem 1. By construction, each timed run produced by the WTS  $\mathcal{T}_i$ , where the notion of timed run is given in Definition 8, is associated with the trajectory  $x_i(t)$  of the system (1a)-(1b), as given in Definition 12. Hence, if a timed run of  $\mathcal{T}_i$  of each robot  $i \in [N]$  satisfying the given MITL formula  $\varphi_i$  is found, a desired timed word of the original system, and hence a trajectory  $x_i(t)$  that is a solution to Problem 1 is found.

#### 4.4 Control Synthesis

Fig. 3 depicts a framework under which a sequence of feedback control laws  $u_i(x_i, v_i)$  of each robot that guarantees the satisfaction of the MITL formula  $\varphi_i$  can be computed. First, a TBA  $\mathcal{A}_i$  that accepts all the timed runs satisfying the specification formula  $\varphi_i$  is constructed. Second, a product between the WTS  $\mathcal{T}_i$  given in Definition 14 and the TBA  $\mathcal{A}_i$  is computed which gives the product WTS  $\tilde{\mathcal{T}}_i$ . By performing graph search to the product WTS  $\tilde{\mathcal{T}}_i$ , a timed run that satisfies the MITL formula  $\varphi$  can be found. For more details regarding the control synthesis procedure we refer to our previous work [13, 15].

In view of Algorithm 3, (23) and the offline plan computation, it is possible that while each agent is executing online its individual actions and transits between RoI, there might be a cluster of agents that avoid collision between each other. In such a scenario, the online feedback control law avoids the possible collisions, but the navigation time between the RoI will have been different than the one computed by Algorithm 3. In order to resolve this, we propose an offline collision detection algorithm (see Algorithm 4) which detects the cluster of agent that will avoid potential collision when the plan of each agent is executed and updates the transition times between RoI of each agent appropriately.

More specifically, the input to Algorithm 4 is the transition times of each agent and the output is the

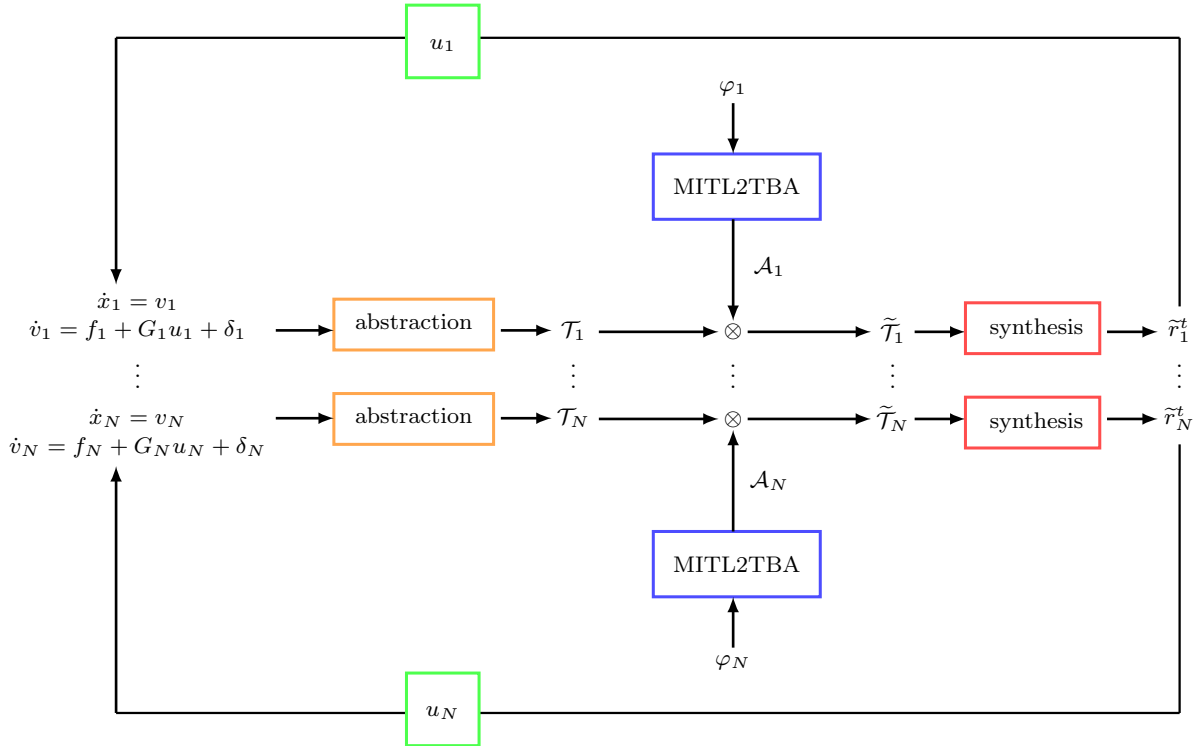


Fig. 3: A graphic illustration of the proposed framework.

updated realistic transition times denoted by  $t_{i,d}^{\text{real}}$  as well as the formula bounds relaxation. The function  $\text{computePlanAgent}(i)$  computes the sequence of RoI that agent  $i$  needs to follow in order to satisfy the formula. The function  $\text{executePlanAgent}(i)$  executes a simulated plan for each agent. Then, by using a monitoring function

$\text{collisionClusterMonitoring}$ ,

the cluster of the agents that are colliding can be detected. Then, we need to update the transition times of each of the colliding agents by a term which models the time duration of the maneuvering that the corresponding agent is performing in order to avoid the collision. This time is denoted in Algorithm 4 by  $T_{\text{maneuver}}^i$ . By finding the maximum of the aforementioned times, the time bounds of the MITL formula of each agent are relaxed. The function  $\text{relaxBounds}(\varphi_i, \max_i)$  updates each formula time interval of the form  $[a, b]$ ,  $a > b \geq 0$ , to  $[a, b + \max_i]$ .

**Proposition 1** *The solution that it is obtained from the controller synthesis procedure provides a sequence of feedback control laws  $u_i(x_i, v_i)$  as in (10) that guarantees the satisfaction of the formula  $\varphi$  of the robot governed by dynamics as in (1a)-(1b), thus, providing a solution to Problem 1.*

---

**Algorithm 3** Offline collision detection and formula bounds relaxation

---

```

1: Input:  $t_{i,s}, \varphi_i, i \in [N]$ ;
2: Output:  $t_{i,d}^{\text{real}}, \bar{\varphi}_i, i \in [N]$ ;
3:  $\max_i \leftarrow 0$ ;
4: for  $i \in [N]$  do
5:    $\text{computePlanAgent}(i)$ ; {Execute the simulated plan of each agent}
6:    $\text{executePlanAgent}(i)$ ;
7: end for
8: for  $i \in \text{collisionClusterMonitoring}$  do
9:    $T_{\text{maneuver}}^i \leftarrow \text{computeManeuverTime}$ ;
10:  if  $T_{\text{maneuver}}^i > \max_i$  then
11:     $\max_i \leftarrow T_{\text{maneuver}}^i$ 
12:  end if
13:   $t_{i,d}^{\text{real}} \leftarrow t_{i,s} + T_{\text{maneuver}}^i$ 
14: end for
15: for  $i \in [N]$  do
16:   $\bar{\varphi}_i \leftarrow \text{relaxBounds}(\varphi_i, \max_i)$ ;
17: end for
    
```

---

#### 4.5 Complexity Analysis

The proposed framework consists of the computational complexity of the following steps:

- **C1:** the computational complexity of the offline construction of WTS  $\tilde{\mathcal{T}}_i$  and graph search. In particular, the graph search is performed over the product WTS  $\tilde{\mathcal{T}}_i$  which has  $|S_i| \cdot |Q_i|$  number of states, i.e., the multiplication between the states of the WTS

(number of RoI of the workspace) and the number of states of the TBA. The complexity of the Dijkstra algorithm that is used for the graph search is:

$\mathcal{O}\left(|S_i| \cdot |Q_i| + |\text{edges}| \log(|S_i| \cdot |Q_i|)\right)$ , where  $|\text{edges}|$

is the number of edges of the product WTS  $\tilde{T}_i$ .

- **C2**: Algorithm 2 is an offline computer simulation and the computational complexity is the same with the complexity of a nominal NMPC algorithm;
- **C3**: Algorithm 3 is an offline computer simulation of collision detection which scales with the number of agents;
- **C4**: the DFHOCP (12a)-(12d) is the only online commutation of the proposed framework and has the same complexity with the nominal NMPC algorithm (quadratic programming optimization technique).

By taking into account that **C1** is standard in timed verification, and the fact that **C2**, **C4** have the same complexity with nominal NMPC, and **C3** is a computer simulation that scales with the number of agents, the proposed approach is scalable with the number of agents.

## 5 Experimental Setup and Results

In this section the efficacy of the proposed framework via a real-time experiment employing  $N = 3$  Nexus 10011 mobile robots is validated. The experiment was conducted at Smart Mobility Lab (SML) (see Fig. 1 and [18]). By controlling the speed of each wheel, the Nexus Robot 10011 is able to move forward, backward, left, and right. The robot can also rotate clockwise and counterclockwise. In other words, it has three degrees of freedoms, i.e. moving forward/backward, moving left/right and rotation. By combining the three degrees of freedom, the Nexus Robot is able to move towards any direction. SML provides a motion capture system (Mo-Cap) with 12 cameras spread across the lab. The Mo-Cap provides the robot state vector, including pose, orientation as well as linear and angular velocities at frequency of 100Hz. The software implementation of the proposed control strategy was conducted in C++ under Robot Operating System (ROS) [33]. Moreover, the optimization algorithms described in this chapter are implemented by employing the NLOPT Optimization library found in [34].

The state of each robot is  $x_i = [x_{i,1}, x_{i,2}, x_{i,3}]^\top$  where  $x_{i,1}$ ,  $x_{i,2}$  indicate the position of the robot and  $x_{i,3}$  its orientation. The workspace that the robots can operate in as well as a panoramic view of it is depicted in Fig. 1 and Fig. 4, respectively. The workspace is cap-

tured by the set:

$$\mathcal{W} := \{w \in \mathbb{R}^2 : |w_k| \leq 2.5, \quad k \in \{1, 2\}\},$$

and it contains 5 RoI which are divided as follows:

- the RoI  $\mathcal{R}_z$ ,  $z \in \{1, 2, 3, 4\}$  depicted with blue color in Fig. 4 which stand for the RoI that the robots are required to visit. The RoI  $\mathcal{R}_z$ ,  $z \in \{1, 2, 3, 4\}$  map into the atomic propositions that model missions for each robot;
- the RoI  $\mathcal{R}_5$  depicted with red color in Fig. 4 stands for an unsafe region that the robots should avoid collision with. It holds that  $\mathcal{L}_i(\mathcal{R}_5) = \{\text{obs}\}$  for every  $i \in [N]$ .

The control input constraints of each robot are set to:

$$\mathcal{U}_i = \{u_i \in \mathbb{R}^3 : |u_{i,k}| \leq 0.15, \quad k \in \{1, 2, 3\}\}, \quad i \in [N],$$

where  $u_{i,1}$ ,  $u_{i,2}$  stand for the linear velocities and  $u_{i,3}$  stands for the angular velocity. The ball that covers the volume of each robot has radius  $r_i = 0.4\text{m}$  for every  $i \in [N]$ . The sensing radius of each robot is  $d_i = 2\text{m}$ . The robots 1, 2 and 3 are initially placed in the ROI  $\mathcal{R}_1$ ,  $\mathcal{R}_2$  and  $\mathcal{R}_3$ , respectively. The set of atomic propositions of each robot is given by:

$$\Pi_1 = \{\text{obs}, \text{mission}_{11}, \text{mission}_{13}\},$$

$$\Pi_2 = \{\text{obs}, \text{mission}_{22}, \text{mission}_{24}\},$$

$$\Pi_3 = \{\text{obs}, \text{mission}_{33}, \text{mission}_{32}\},$$

with the corresponding labeling functions:

$$\mathcal{L}_1(\mathcal{R}_1) = \{\text{mission}_{11}\}, \quad \mathcal{L}_1(\mathcal{R}_2) = \emptyset,$$

$$\mathcal{L}_1(\mathcal{R}_3) = \{\text{mission}_{13}\}, \quad \mathcal{L}_1(\mathcal{R}_4) = \emptyset,$$

$$\mathcal{L}_2(\mathcal{R}_1) = \emptyset, \quad \mathcal{L}_2(\mathcal{R}_2) = \{\text{mission}_{22}\},$$

$$\mathcal{L}_2(\mathcal{R}_3) = \emptyset, \quad \mathcal{L}_2(\mathcal{R}_4) = \{\text{mission}_{24}\},$$

$$\mathcal{L}_3(\mathcal{R}_1) = \emptyset, \quad \mathcal{L}_3(\mathcal{R}_2) = \{\text{mission}_{32}\},$$

$$\mathcal{L}_3(\mathcal{R}_3) = \{\text{mission}_{33}\}, \quad \mathcal{L}_3(\mathcal{R}_4) = \emptyset.$$

The desired MITL tasks are set to:

$$\varphi_1 = \square_{[0,120]} \{\neg \text{obs}\} \wedge \diamond_{[10,25]} \{\text{mission}_{13}\} \\ \wedge \diamond_{[30,45]} \{\text{mission}_{11}\},$$

$$\varphi_2 = \square_{[0,120]} \{\neg \text{obs}\} \wedge \diamond_{[25,45]} \{\text{mission}_{22}\} \\ \wedge \diamond_{[50,80]} \{\text{mission}_{24}\},$$

$$\varphi_3 = \square_{[0,120]} \{\neg \text{obs}\} \wedge \diamond_{[30,45]} \{\text{mission}_{33}\} \\ \wedge \diamond_{[60,75]} \{\text{mission}_{32}\},$$

respectively. The prediction horizon is chosen  $T = 2.0$  sec. The tube of each robot is given by the set:

$$\Omega_i = \left\{ \mathbf{e}_i : \|\mathbf{e}_i\| \leq \frac{\tilde{\delta}_i}{k_i} \right\}.$$

The NMPC gains are set to:

$$Q_i = P_i = R_i = 0.5I_3, \quad i \in [N].$$

By using Algorithm 3 and Algorithm 4, the total transition times of the navigation of the robots between the RoI of the workspace are computed as follows:

$$\begin{aligned} t_1(\mathcal{R}_1, \mathcal{R}_3) &= t_1(\mathcal{R}_3, \mathcal{R}_1) = 18, \\ t_2(\mathcal{R}_2, \mathcal{R}_4) &= t_2(\mathcal{R}_4, \mathcal{R}_2) = 20, \\ t_3(\mathcal{R}_2, \mathcal{R}_3) &= t_3(\mathcal{R}_3, \mathcal{R}_2) = 16. \end{aligned}$$

By using the proposed framework, we find a sequence of runs of each agent that fulfills the given MITL task. The sequence of runs maps into a sequence of feedback control laws that the robot execute online and fulfill the given tasks. By online executing the proposed plan, the trajectories of the robots in the workspace are depicted in Fig. 5- Fig. 7.

**Video:** A video demonstrating the experiment of this section can be found in the following link:

<https://www.youtube.com/watch?v=9ZNV1EjKZ9g>

## 6 Conclusions and Future Work

In this paper, a scalable framework for time-constrained planning of multi-robot systems has been proposed. Considering  $N$  robots operating in a bounded workspace which contains RoI, assigned with tasks given in MITL, a framework for efficiently designing decentralized feedback control laws that guarantee the satisfaction of the corresponding tasks has been provided. The controllers are the outcome of DFHOCP solved by each robot at each sampling time and form the actions of the WTS. By proposing high-level controller synthesis algorithms, a sequence of feedback laws for each robot can be designed. The approach is scalable since the local products are computed offline and only the DFHOCP of each robot is computed online which has complexity similar with the nominal NMPC framework. Future research directions will be devoted towards incorporating event-triggered strategies between the robots in order to save valuable actuation and sensing resources.

**Acknowledgements** This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the EU H2020 Co4Robots project, the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

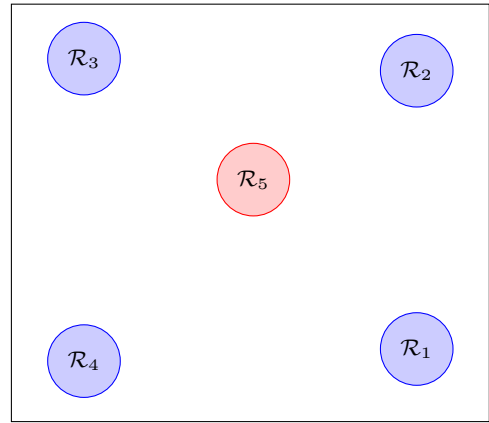


Fig. 4: A panoramic view of the workspace with the 5 RoI.

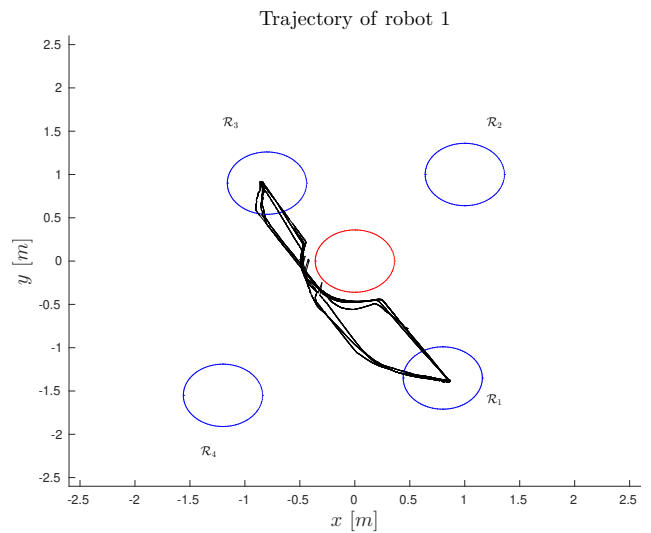


Fig. 5: The trajectory of robot 1 in the workspace.

## References

1. T. Wongpiromsarn, U. Topcu, and R. Murray, "Receding Horizon Temporal Logic Planning for Dynamical Systems," pp. 5997–6004, December 2009.
2. A. Nikou, "Robust and Decentralized Control of Multi-agent Systems under High-level Tasks," *Doctoral Thesis, KTH Royal Institute of Technology*, 2019.
3. Y. Kantaros and M. Zavlanos, "A Distributed LTL-Based Approach for Intermittent Communication in Mobile Robot Networks," *American Control Conference (ACC)*, pp. 5557–5562, Boston, USA, 2016.
4. M. Hasanbeig, Y. Kantaros, A. Abate, D. Kroening, G. J. Pappas, and I. Lee, "Reinforcement Learning for Temporal Logic Control Synthesis with Probabilistic Satisfaction Guarantees," *arXiv*, 2019.
5. Y. Pant, R. Quaye, H. Abbas, A. Varre, and R. Mangharam, "Fly-by-logic: A tool for unmanned aircraft system fleet planning using temporal logic," *NASA Formal Methods Symposium*, 2019.
6. Y. Pant, H. Abbas, R. Quaye, and R. Mangharam, "Fly-by-Logic: Control of Multi-drone Fleets with Temporal

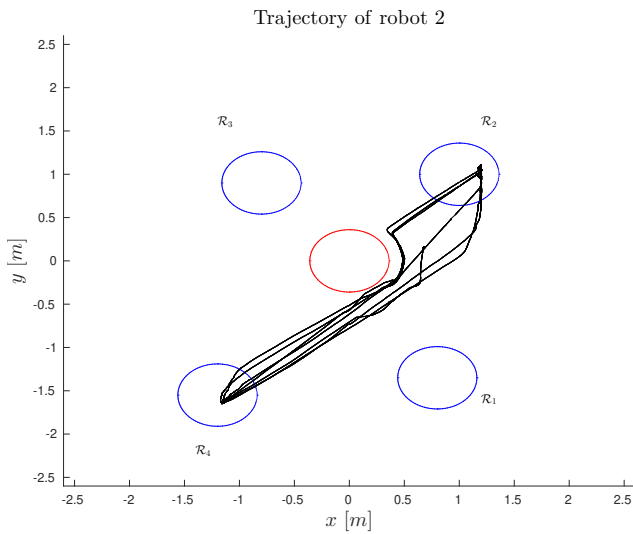


Fig. 6: The trajectory of robot 2 in the workspace.

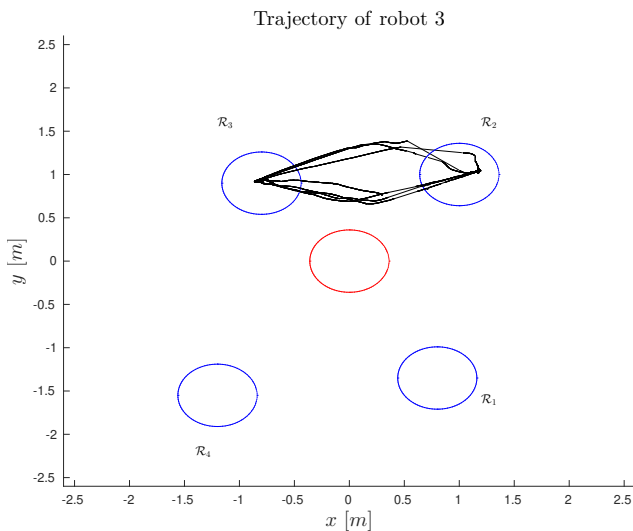


Fig. 7: The trajectory of robot 3 in the workspace.

- Logic Objectives,” *9th ACM/IEEE International Conference on Cyber-physical Systems*, 2018.
7. V. Raman, A. Donze, M. Maasoumy, R. Murray, A. Vincentelli, and S. Seshia, “Model Predictive Control for Signal Temporal Logic Specifications,” *IEEE Conference on Decision and Control (CDC)*, 2014.
  8. 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 (IJRR)*, vol. 32, no. 8, pp. 889–911, July 2013.
  9. Z. Xu, M. Ornik, A. Julius, and U. Topcu, “Information-guided Temporal Logic Inference with Prior Knowledge,” *2019 American Control Conference (ACC)*, pp. 1891–1897, July 2019.
  10. G. Fainekos, A. Girard, H. Kress-Gazit, and G. Pappas, “Temporal Logic Motion Planning for Dynamic Robots,” *Automatica*, vol. 45, no. 2, pp. 343–352, February 2009.
  11. J. Fang, Z. Zhang, and R. Cowlagi, “Decentralized Route-planning to Satisfy Global Linear Temporal Logic Spec-

- ifications on Multiple Aircraft,” *2018 AIAA Guidance, Navigation, and Control Conference*, p. 1862, 2018.
12. R. Alur and D. Dill, “A Theory of Timed Automata,” *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.
  13. A. Nikou, J. Tumova, and D. V. Dimarogonas, “Cooperative Task Planning of Multi-Agent Systems Under Timed Temporal Specifications,” *American Control Conference (ACC)*, pp. 7104–7109, Boston, USA, 2016.
  14. A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, “Cooperative Planning Synthesis for Coupled Multi-Agent Systems Under Timed Temporal Specifications,” *American Control Conference (ACC)*, pp. 1847–1852, Seattle, USA, 2017.
  15. A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, “On the Timed Temporal Logic Planning of Coupled Multi-Agent Systems,” *Automatica*, vol. 97, pp. 339–345, 2018.
  16. A. Nikou, S. Heshmati-alamdari, C. Verginis, and D. V. Dimarogonas, “Decentralized Abstractions and Timed Constrained Planning of a General Class of Coupled Multi-Agent Systems,” *56th IEEE Conference on Decision and Control (CDC)*, pp. 990–995, Melbourne, Australia, 2017.
  17. S. Karaman and E. Frazzoli, “Vehicle Routing Problem with Metric Temporal Logic Specifications,” *2008 47th IEEE Conference on Decision and Control*, pp. 3953–3958, 2008.
  18. “Smart Mobility Lab (SML),” <https://www.kth.se/dcs/research/control-of-transport/smart-mobility-lab/smart-mobility-lab-1.441539>.
  19. A. Nikou and D. V. Dimarogonas, “Decentralized Tube-based Model Predictive Control of Uncertain Nonlinear Multi-Agent Systems,” *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 29, pp. 2799–2818, June 2019.
  20. H. Khalil, *Nonlinear Systems*. Prentice-Hall, New Jersey, 1996.
  21. S. Yu, C. Maier, H. Chen, and F. Allgöwer, “Tube MPC Scheme Based on Robust Control Invariant Set with Application to Lipschitz Nonlinear Systems,” *Systems and Control Letters*, vol. 62, no. 2, pp. 194–200, 2013.
  22. H. Chen and F. Allgöwer, “A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability,” *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
  23. 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.
  24. J. Ouaknine and J. Worrell, “On the Decidability of Metric Temporal Logic,” *20th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pp. 188–197, 2005.
  25. P. Bouyer, “From Qualitative to Quantitative Analysis of Timed Systems,” *Mémoire d’habilitation, Université Paris*, vol. 7, pp. 135–175, 2009.
  26. S. Tripakis, “Checking Timed Buchi Automata Emptiness on Simulation Graphs,” *ACM Transactions on Computational Logic (TOCL)*, vol. 10, no. 3, 2009.
  27. 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.
  28. 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.



29. D. Ničković and N. Piterman, “From MTL to Deterministic Timed Automata,” *Formal Modeling and Analysis of Timed Systems*, 2010.
30. 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.
31. M. Krstic, I. Kanellakopoulos, and P. Kokotovic, “Nonlinear and Adaptive Control Design,” *Publisher: Wiley New York*, 1995.
32. A. Filotheou, A. Nikou, and D. V. Dimarogonas, “Robust Decentralized Navigation of Multi-Agent Systems with Collision Avoidance and Connectivity Maintenance Using Model Predictive Controllers,” *International Journal of Control (IJC)*, 2018.
33. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: An Open-Source Robot Operating System,” *International Conference on Robotics and Automation (ICRA) Workshop*, 2009.
34. S. G. Johnson, “The NLOpt Nonlinear-Optimization Package,” <http://ab-initio.mit.edu/nlopt>, 2009.

**Alexandros Nikou** received the Diploma in Electrical and Computer Engineering in 2012 and the M.Sc. in Automatic Control in 2014, both from National Technical University of Athens (NTUA), Greece. He received his PhD in Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden in 2019. He is currently senior researcher at Ericsson Research AI,



Stockholm, Sweden. His current research interests include robust and decentralized control, formal methods and reinforcement learning.

**Shahab Heshmati-alamdari** received the Diploma degree in mechanical engineering, the M.S. degree in robotics and automatic control systems, and the Ph.D. degree in mechanical engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2009, 2012, and 2018, respectively. He is currently a Post-Doctoral Researcher with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. His research



interests include nonlinear control systems applications in sensor-based motion planning, marine robotics, control of multiple robotic systems, such as free flying manipulators (e.g., Underwater Vehicle Manipulator Systems) and vehicles (mobile, underwater, and aerial).



**Dimos V. Dimarogonas** was born in Athens, Greece, in 1978. He received the Diploma in Electrical and Computer Engineering in 2001 and the PhD in Mechanical Engineering in 2007, both from NTUA, Athens, Greece. Between May 2007 and February 2009, he held post doctoral positions at KTH, Stockholm, Sweden, and MIT, Boston, USA. He is currently Professor at the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. His current research interests include Multi-Agent Systems, Hybrid Systems and Control, Robot Navigation and Networked Control. He serves in the Editorial Board of *Automatica*, the *IEEE Transactions on Control of Network Systems* and the *Unmanned Systems* and is a Senior member of IEEE and the Technical Chamber of Greece.