Distributed MPC with continuous-time STL constraint satisfaction guarantees

Maria Charitidou, *Graduate Student Member, IEEE*, Dimos V. Dimarogonas, *Fellow, IEEE*

***Abstract*— In this work a distributed model predictive control scheme (dMPC) is proposed for a multi-agent team that is subject to a set of time-constrained spatial tasks encoded in Signal Temporal Logic (STL). Here, the agents are subject to both individual and collaborative STL tasks. In order to ensure the satisfaction of the collaborative tasks while avoiding the computational burden of a centralized problem, we propose a sequential dMPC scheme and show the recursive feasibility property of the framework given appropriately designed terminal ingredients. The resulting MPC problems are solved in discrete-time yet continuous-time satisfaction of the STL tasks is ensured with appropriate tightening of the constraint sets.**

***Index Terms*— Model Predictive Control, Multi-agent systems, Signal Temporal Logic, Sampled-data control**

I. Introduction

SIGNAL Temporal Logic (STL) [1] is a formal specification language capable of expressing complex tasks that need to be performed within strict deadlines. Contrary to other logics, STL is evaluated over continuous time signals and is equipped with a metric [2] that determines how well the STL task is satisfied. Control under STL specifications for single-agent systems has been considered in [3]–[10]. In [3]–[6] the STL tasks are encoded as constraints to an integer program that aims at maximizing the space or time robustness of the formula. These approaches address the design of discrete-time plans and thus cannot ensure the satisfaction of the STL task in continuous time. Continuous time constraint satisfaction has been studied in [7] where the satisfaction of the STL constraints in continuous time is ensured by enforcing the minimum value of a control barrier function over each sampling interval to be non-negative. In [8]–[10] the satisfaction of the STL formula is ensured using control barrier function based feedback controllers for nonlinear, continuous-time, input-affine systems.

In the context of multi-agent control distributed or decentralized approaches have been studied in [11]–[14]. In [11] a hierarchical MPC scheme is proposed for a multi-agent team subject to local motion and safety STL tasks and global communication tasks expressed in spatial-temporal logic (SpaTeL). A two step approach is proposed in [12] for the satisfaction of local motion, local safety and global cooperative STL tasks, where agents re-compute their plans in a hierarchical, iterative manner using information from other agents to ensure satisfaction or minimal violation of the global tasks. In [13] a parallel distributed scheme is presented for single-integrator systems subject to reachability and safety tasks that is guaranteed to be recursively feasible with appropriately designed terminal sets. In all the aforementioned works continuous time satisfaction of the STL formulas is not guaranteed while all of them are tailored to reach-avoid problems. General STL tasks are tackled in [14], where decentralized continuous-time feedback controllers are designed using time-varying control barrier functions (CBFs). This approach is shown to be more computationally efficient alleviating integer encoding of the STL constraints. Yet designing CBF functions for systems subject to input constraints is still non-trivial. The problem of control under STL specifications with limited actuation has been considered in our previous work [15], where a continuous-time, centralized MPC scheme was proposed for linear systems subject to state, STL and input constraints. In the proposed problem the planning horizon of the problem can be chosen arbitrarily small and independent of the horizon of the STL formula given initial feasibility thanks to appropriately designed terminal sets.

Considering the favorable properties of the previous problem formulation in [15], in this work we propose a sequential distributed sampled-data MPC scheme for a multi-agent team with limited actuation capabilities that is subject to individual and cooperative STL tasks. In the proposed scheme each agent plans its own actions given information from other agents in the team in a hierarchical manner and sends its plan along with state information of higher in the hierarchy agents to its immediate neighbors. The designed plans are then executed synchronously when the last agent has planned its action. The sampled-data scheme ensures the satisfaction of the STL formula in continuous time by only evaluating a finite number of constraints thanks to an appropriate tightening of the initial constraint sets. The proposed framework is then shown to be recursively feasible thanks to appropriately designed local terminal constraints while the resulting control laws are piecewise constant and thus can be easily implemented by modern digital controllers.

II. Preliminaries and Problem Formulation

True and false are denoted by $\top, \bot$ respectively. Scalars and vectors are denoted by non-bold and bold letters respectively.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the ERC CoG LEAFHOUND and the Swedish Research Council.

Both authors are with KTH Royal Institute of Technology, Stockholm, SE 100-44 Sweden, (e-mail:{mariacha,dimos}@ kth.se).

Given a finite set $\mathcal{V} \subset \mathbb{N}$, $\prod_{k\in\mathcal{V}} \mathbb{X}_k$ denotes the Cartesian product of the sets $\mathbb{X}_k, k \in \mathcal{V}$. $\text{diag}(A_1, \ldots, A_n)$ denotes the block matrix with main diagonal elements the matrices $A_1, \ldots, A_n$. Given a convex set $\mathcal{C} \in \mathbb{R}^{m\times n}$ we define the set to set mapping as $\mathcal{C}^{-1}(Y) := \{\mathbf{z} \in \mathbb{R}^n : C\mathbf{z} \in Y, \forall C \in \mathcal{C}\}$. For a mapping $f : \mathbb{R}^m \to \mathbb{R}^n$ and set $\mathcal{C} \subseteq \mathbb{R}^m$ let $f(\mathcal{C}) := \{f(x) : x \in \mathcal{C}\}$. A *directed* graph G is defined as a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{1, \ldots, R\} \subset \mathbb{N}$ is a finite set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges of G. The set of neighbors of the r-th node, i.e., the set of nodes with incoming edges to r, is defined as $\mathcal{N}_r := \{r' \in \mathcal{V} : (r', r) \in \mathcal{E}\}$.

A. Signal Temporal Logic (STL)

Signal Temporal Logic (STL) determines whether a predicate μ is true or false. The validity of each predicate μ is evaluated based on a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ as follows: $\mu = \top$, if $h(\boldsymbol{\zeta}) \geq 0$, or $\mu = \bot$, otherwise. The basic STL formulas are given by the grammar: $\phi := \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathcal{G}_{[a,b]}\phi \mid \mathcal{F}_{[a,b]}\phi \mid \phi_1 \; \mathcal{U}_{[a,b]} \; \phi_2$ where ϕ_1, ϕ_2 are STL formulas and $\mathcal{G}_{[a,b]}$, $\mathcal{F}_{[a,b]}$, $\mathcal{U}_{[a,b]}$ is the always, eventually and until operator defined over the interval $[a, b]$ with $0 \leq a \leq b < \infty$. Let $\boldsymbol{\zeta} \models \phi$ denote the satisfaction of the formula ϕ by a signal $\boldsymbol{\zeta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. The formula ϕ is satisfiable if $\exists\, \boldsymbol{\zeta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ such that $\boldsymbol{\zeta} \models \phi$. The STL semantics and robust semantics for a signal $\boldsymbol{\zeta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ are recursively determined. The exact definitions can be found in [1], [2] but omitted here due to space limitations. Note that $\boldsymbol{\zeta} \models \phi$, if $\rho^{\phi}(\boldsymbol{\zeta}, 0) > 0$.

B. Encoding STL tasks with continuous variables

For every task of the form $\varphi_i = \mathcal{T}_{[a_i,b_i]}(h_i(\boldsymbol{\zeta}(t)) \geq 0)$, where $\mathcal{T} = \{\mathcal{G}, \mathcal{F}\}$ the authors in [10] define the continuous function $\mathfrak{b}_i : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ as $\mathfrak{b}_i(\boldsymbol{\zeta}(t), t) = -\gamma_i(t) + h_i(\boldsymbol{\zeta}(t))$, where $h_i : \mathbb{R}^n \to \mathbb{R}$ is the predicate function corresponding to the task and $\gamma_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a temporal function which ensures the satisfaction of φ_i at time t_i^* with a minimum robustness $\bar{\rho}$, where $\bar{\rho} > 0$ is a designer's choice. Here, $t_i^* = a_i$ if $\mathcal{T} = \mathcal{G}$ and $t_i^* \in [a_i, b_i]$ otherwise. Here, we consider a piecewise linear function defined as: $\gamma_i(t) = \frac{\gamma_{i,\infty} - \gamma_{i,0}}{t_i^*} t + \gamma_{i,0}$, if $t < t_i^*$ or $\gamma_i(t) = \gamma_{i,\infty}$ otherwise, where $\gamma_{i,0}, \gamma_{i,\infty}$ are parameters to be tuned to ensure that $h_i(\boldsymbol{\zeta}(0)) \geq \gamma_{i,0}$ and $\gamma_{i,\infty} > \max(\bar{\rho}, \gamma_{i,0})$. For more details on the design of $\gamma_i(t)$ the reader may refer to [14]. Finally, for each STL formula φ_i we define the set Σ_i as follows: $\Sigma_i = \{a_i, b_i : a_i \neq 0\}$, if $\mathcal{T} = \mathcal{G}$ or $\Sigma_i = \{t_i^* : t_i^* \neq 0\}$, otherwise. In the latter case $t_i^* \neq 0$ ensures that only eventually tasks not trivially satisfied at $t = 0$ are considered. This set will be later used to define the time-varying constraint sets of the MPC scheme.

C. Multi-Agent Dynamics

Consider a team of R agents. Each agent $r \in \mathcal{V} := \{1, \ldots, R\}$ is modelled as a linear system whose dynamics are given as follows:

$$\dot{\mathbf{x}}_r = A_r \mathbf{x}_r + B_r \mathbf{u}_r, \tag{1}$$

where $A_r \in \mathbb{R}^{n_r \times n_r}$, $B_r \in \mathbb{R}^{n_r \times m_r}$ and $\mathbf{x}_r \in \mathbb{R}^{n_r}$, $\mathbf{u}_r \in \mathbb{R}^{m_r}$ is the state and input of agent r. Here, each agent is subject to state and input constraints of the form $\mathbf{x}_r \in \mathbb{X}_r$ and $\mathbf{u}_r \in \mathbb{U}_r$, where $\mathbb{X}_r, \mathbb{U}_r$ are compact, polyhedral sets. For simplicity we will assume them to be box sets, i.e., sets of the form $\mathbb{X}_r = \prod_{k=1}^{n_r}[x^r_{\min,k}, x^r_{\max,k}]$ and $\mathbb{U}_r = \prod_{k=1}^{m_r}[u^r_{\min,k}, u^r_{\max,k}]$ but note that the proposed approach can be extended to general compact sets of the form $\{\mathbf{y}_r : C_r^{\mathbf{y}}\mathbf{y}_r \leq c_r^{\mathbf{y}}\}$, where $\mathbf{y} \in \{\mathbf{x}, \mathbf{u}\}$.

D. Problem Formulation

In this work the agents are assumed to be subject of a global STL task ϕ described by the following fragment:

$$\psi := \top \mid \mu \mid \neg\mu \mid \psi_1 \wedge \psi_2 \tag{2a}$$
$$\varphi := \mathcal{G}_{[a,b]}\psi \mid \mathcal{F}_{[a,b]}\psi \mid \psi_1 \; \mathcal{U}_{[a,b]} \; \psi_2 \tag{2b}$$
$$\phi := \varphi_1 \wedge \ldots \wedge \varphi_{\bar{q}} \tag{2c}$$

where ψ_1, ψ_2 are STL formulas of the form (2a), $\varphi_1, \ldots, \varphi_{\bar{q}}$ are STL formulas of the form (2b) and $a \leq b < \infty$. In the following we will assume that ϕ is defined as a conjunction of always and eventually STL tasks as the satisfaction of the formulas $\mathcal{G}_{[a,t']}\psi_1 \wedge \mathcal{F}_{[t',t']}\psi_2$, $\bigwedge_l(\mathcal{F}_{[t',t']}\psi_l)$ implies the satisfaction of $\psi_1 \; \mathcal{U}_{[a,b]} \; \psi_2$, and $\mathcal{F}_{[a,b]}(\bigwedge_l \psi_l)$, respectively, when $t' \in [a, b]$. In particular we consider the following global STL task:

$$\phi = \bigwedge_{r\in\mathcal{V}} \bigwedge_{i\in\mathcal{I}_r} \bar{\varphi}_i \wedge \bigwedge_{i\in\mathcal{I}} \varphi_i, \tag{3}$$

where $\bigwedge_{i\in\mathcal{I}_r} \bar{\varphi}_i, r \in \mathcal{V}$ are individual tasks assigned to agent r and $\bigwedge_{i\in\mathcal{I}} \varphi_i$ are collaborative tasks, where $\mathcal{I}_r := \{1, \ldots, s_r\}, r \in \mathcal{V}$ and $\mathcal{I} := \{1, \ldots, s\}$. Examples of collaborative tasks are moving in a formation, connectivity maintenance or collision avoidance while examples of individual tasks are reaching a desired region or avoiding a static obstacle. For every $i \in \mathcal{I}$, let $\mathcal{V}_i \subseteq \mathcal{V}$ be the set of indices of the agents involved in the satisfaction of φ_i.

Based on the above we can define the problem considered in this work as follows:

Problem 1. *Consider a team of agents $\mathcal{V}$ whose states evolve over time according to* (1)*. The agents are subject to a global STL task ϕ defined by* (3)*. Based on the above, design the control input $\mathbf{u}_r(t) \in \mathbb{U}_r$ for each agent $r \in \mathcal{V}$ over the time interval $[0, T)$ for a known $T > 0$ using only local information such that: 1) $\rho^{\phi}(\mathbf{x}, 0) \geq \bar{\rho}$, where $\bar{\rho}$ is a design parameter and 2) $\mathbf{x}_r(t) \in \mathbb{X}_r$ for every $r \in \mathcal{V}$ and $t \in [0, T]$.*

III. Centralized MPC

We begin by introducing the centralized MPC problem for the multi-agent team under the STL constraints introduced in (3). We denote the stacked vector of the states of the agents in $\mathcal{V}_i$ and $\mathcal{V}$ by $\mathbf{x}^{\mathcal{V}_i} \in \prod_{r\in\mathcal{V}_i} \mathbb{X}_r \subset \mathbb{R}^{\overline{n}_i}$, and $\mathbf{x} \in \prod_{r\in\mathcal{V}} \mathbb{X}_r \subset \mathbb{R}^n$ respectively, where $\overline{n}_i = \sum_{r\in\mathcal{V}_i} n_r$ and $n = \sum_{r\in\mathcal{V}} n_r$.

For every individual STL task $\bar{\varphi}_i, i \in \mathcal{I}_r, r \in \mathcal{V}$ and every collaborative task $\varphi_i, i \in \mathcal{I}$ define the functions $\bar{\mathfrak{b}}_i : \mathbb{R}^{n_r} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $\mathfrak{b}_i : \mathbb{R}^{\overline{n}_i} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ as in Section II-B as follows:

$$\bar{\mathfrak{b}}_i(\mathbf{x}_r(t), t) = -\bar{\gamma}_i(t) + \bar{h}_i(\mathbf{x}_r(t)), \quad i \in \mathcal{I}_r \tag{4a}$$

$$\mathfrak{b}_i(\mathbf{x}^{\mathcal{V}_i}(t),t) = -\gamma_i(t) + h_i(\mathbf{x}^{\mathcal{V}_i}(t)), \quad i \in \mathcal{I}, \tag{4b}$$

where $\overline{\gamma}_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$, $\gamma_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are appropriately designed temporal functions ensuring the satisfaction of the corresponding task with robustness $\bar{\rho} > 0$ and $\overline{h}_i : \mathbb{R}^{n_r} \to \mathbb{R}$, $h_i : \mathbb{R}^{\overline{n}_i} \to \mathbb{R}$, is the predicate function corresponding to $\bar{\varphi}_i$ and φ_i, respectively. Here, we make the following assumption:

Assumption 1. *The predicate functions $\overline{h}_i : \mathbb{R}^{n_r} \to \mathbb{R}, i \in \mathcal{I}_r, r \in \mathcal{V}$ and $h_i : \mathbb{R}^{\overline{n}_i} \to \mathbb{R}, i \in \mathcal{I}$ are linear.*

Assumption 1 is introduced to ensure that the constraint set of the proposed MPC is polyhedral. Similar assumptions have been made for planning under STL specifications e.g., in [3], [4]. Such functions can be used to innerapproximate the superlevel set of more complex, nonlinear predicate functions offering computational benefits in the implementation of the MPC in real time at the cost of a certain conservatism in expressing the spatial tasks.

Definition 1. *A task $\varphi_i = \mathcal{T}_{[a_i,b_i]}(h_i(\mathbf{x}^{\mathcal{V}_i}) \geq 0), i \in \mathcal{I}$ (or $\bar{\varphi}_i = \mathcal{T}_{[a_i,b_i]}(\overline{h}_i(\mathbf{x}_r) \geq 0), i \in \mathcal{I}_r, r \in \mathcal{V}$) is called active at time t iff $t \in [0,b_i]\backslash\Sigma_i$, when $\mathcal{T} = \mathcal{G}$ or $t \in [0,t_i^*]\backslash\Sigma_i$, when $\mathcal{T} = \mathcal{F}$.*

Roughly speaking, Definition 1 determines the tasks that have not been satisfied yet at t. We denote the set of active cooperative tasks at time t by $\mathcal{A}(t)$ and the active individual tasks of agent r at t by $\mathcal{A}_r(t)$. Let $N > 0$ denote the optimization horizon of the problem and assume that the agents' states are available at time instants $\tau_j, j \in \mathcal{J} := \{0, 1, \dots, M\}$ where $\tau_0 = 0, M \leq +\infty$. Here, we pose the following assumption on τ_j:

Assumption 2. *Consider the monotonically increasing sequence $\{\tau_j\}_{j=0}^M$ with $\tau_0 = 0$, $\tau_{M_1+1} = \max(\Sigma\backslash\Sigma^{\mathcal{G}})$ and $M > M_1 + 1$, where $\Sigma^{\mathcal{G}} := \bigcup_{i\in\overline{\mathcal{I}}}\{a_i : \varphi_i' = \mathcal{G}_{[a_i,b_i]}\mu_i'\}$, $\overline{\mathcal{I}} = \mathcal{I} \cup \bigcup_{r\in\mathcal{V}} \mathcal{I}_r$. Then, for every $\sigma \in \Sigma$, where $\Sigma := \bigcup_{i\in\overline{\mathcal{I}}} \Sigma_i$, there exists $j \in \mathcal{J} = \{1, \dots, M\}$ such that $\sigma = \tau_j$. In addition, for every $j \in \mathcal{J}_1 := \{0, \dots, M_1\} \subset \mathcal{J}$ there exists $j' \in \mathcal{J}$ such that $\tau_{j'} = \tau_j + N$.*

Assumption 2 ensures that it is always possible to obtain the state of the agents at the time instants at which a constraint expressing the satisfaction of a given STL task is deactivated. In addition, since the sampling instants are not necessarily periodic it ensures that $\tau_j + N$ is a sampling instant as well. Let $\mathfrak{p} : \mathcal{J}_1 \to \mathcal{J}$ be a function that assigns each $j \in \mathcal{J}_1$ to some $j' \in \mathcal{J}$ such that $\tau_{\mathfrak{p}(j)} = \tau_j + N$, where $j' = \mathfrak{p}(j)$. For example, if periodic sampling is considered and $N = k\delta$, where $k \in \mathbb{N}$ and $\delta > 0$ is the sampling period, then $\mathfrak{p}(j) = j + k$. At each time instant $\tau_j, j \in \mathcal{J}_1$ we consider the following continuous-time optimization problem:

$$\inf_{\mathbf{u}_r, r\in\mathcal{V}} \sum_{r\in\mathcal{V}} \int_{\tau_j}^{\tau_{\mathfrak{p}(j)}} L_r(\mathbf{u}_r(t), \mathbf{x}_r(t))dt \tag{5}$$

$$\dot{\mathbf{x}}_r(t) = A_r\mathbf{x}_r(t) + B_r\mathbf{u}_r(t),\ \mathbf{x}_r(\tau_j) = \overline{\mathbf{x}}_r^j \tag{5a}$$

$$\overline{\mathfrak{b}}_i(\mathbf{x}_r(t),t) \geq 0, \quad i \in \mathcal{A}_r(t) \tag{5b}$$

$$\mathfrak{b}_i(\mathbf{x}^{\mathcal{V}_i}(t),t) \geq 0, \quad i \in \mathcal{A}(t) \tag{5c}$$

$$\mathbf{x}_r(\tau_{\mathfrak{p}(j)}) \in \mathbb{X}_r^f(\tau_{\mathfrak{p}(j)}), \tag{5d}$$

$$\mathbf{x}_r(t) \in \mathbb{X}_r, \qquad \mathbf{u}_r(t) \in \mathbb{U}_r, \tag{5e}$$

for every $r \in \mathcal{V}$ and $t \in T_j := [\tau_j, \tau_j + N)$, where the admissible inputs $\mathbf{u}_r(t), r \in \mathcal{V}$ are assumed to be piecewise continuous, $L_r : \mathbb{R}^{n_r} \times \mathbb{R}^{m_r} \to \mathbb{R}$ is a continuously differentiable function with respect to the state and input of agent r, $\overline{\mathbf{x}}_r^j \in \mathbb{R}^{n_r}$ is the measured state of agent r at τ_j, and $\mathbb{X}_r^f(t) \subseteq \mathbb{X}_r, r \in \mathcal{V}$ is a polyhedral terminal set defined with respect to the states of the r-th agent only satisfying certain properties discussed in detail in Section IV. Let $\mathcal{X}_r(t) = \{\mathbf{x}_r \in \mathbb{X}_r : \text{(5b) is satisfied at } t\}, r \in \mathcal{V}$ and $\mathcal{X}(t) = \{\mathbf{x} \in \prod_{r\in\mathcal{V}}\mathbb{X}_r : \text{(5c) is satisfied at } t\}$. If $\mathcal{A}_r(t) = \emptyset$ or $\mathcal{A}(t) = \emptyset$ at some t, then we write $\mathcal{X}_r(t) = \mathbb{X}_r$ and $\mathcal{X}(t) = \prod_{r\in\mathcal{V}}\mathbb{X}_r$, respectively. Observe that due to Assumption 1 the sets $\mathcal{X}(t)$ and $\mathcal{X}_r(t), r \in \mathcal{V}$, are polyhedral. Nevertheless, despite the linear nature of the system dynamics and constraints the continuous-time MPC problem (5) is often hard to be solved in practice as it requires the satisfaction of an infinite number of constraints in every time interval T_j.

Next, we will design a discrete-time MPC problem that requires the evaluation of the state constraints over a finite number of points while continuous-time constraint satisfaction will be ensured by appropriate tightening of the constraints.

Let $\mathbf{u}_r(t) := \sum_{j\in\mathcal{J}} o_j(t)\mathbf{u}_r^j$, be a piecewise constant control input, where $o_j(t) = 1$, if $t \in \overline{T}_j$, or $o_j(t) = 0$, otherwise, $\overline{T}_j := [\tau_j, \tau_{j+1})$ and $\mathbf{u}_r^j \in \mathbb{R}^{m_r}$ is the constant control input within $\overline{T}_j$. Let further $\mathbf{x}_r(t; \tau_0, \mathbf{x}_r^0, \mathbf{u}_r)$, $\mathbf{x}_r^j := \mathbf{x}_r(\tau_j; \tau_0, \mathbf{x}_r^0, \mathbf{u}_r)$ denote the continuous-time solution of (1) when $\mathbf{u}_r(t)$ is applied to the system at time t and $\tau_j, j \in \mathcal{J}$, respectively. The stacked vector of the solutions of all agents at time t and τ_j is denoted by $\mathbf{x}(t; \tau_0, \mathbf{x}^0, \mathbf{u})$ and $\mathbf{x}^j := \mathbf{x}(\tau_j; \tau_0, \mathbf{x}^0, \mathbf{u})$, respectively. For each $r \in \mathcal{V}$ the solution of (1) at time $t \in \overline{T}_j$ is given by:

$$\mathbf{x}_r(t; \tau_j, \mathbf{x}_r^j, \mathbf{u}_r) = \Delta_r(t - \tau_j)\mathbf{x}_r^j + \Gamma_r(t - \tau_j)\mathbf{u}_r^j, \tag{6}$$

where $\Delta_r(t) := \exp(A_r t)$ and $\Gamma_r(t) := \int_0^t \exp(A_r s)B_r ds$. Then, the local constraint $\mathbf{x}_r(t) \in \mathcal{X}_r(t)$ and the coupled constraints $\mathbf{x}(t) \in \mathcal{X}(t)$ can be expressed for $t \in \overline{T}_j$ as:

$$\Delta_r(t - \tau_j)\mathbf{x}_r^j + \Gamma_r(t - \tau_j)\mathbf{u}_r^j \in \mathcal{X}_r(t), \tag{7a}$$

$$\Delta(t - \tau_j)\mathbf{x}^j + \Gamma(t - \tau_j)\mathbf{u}^j \in \mathcal{X}(t), \tag{7b}$$

where $\Delta(t) := \mathrm{diag}(\Delta_1(t), \dots, \Delta_R(t))$ and $\Gamma(t) := \mathrm{diag}(\Gamma_1(t), \dots, \Gamma_R(t))$. We start with our first observation depicted in the following lemma:

Lemma 1. *Define $\Phi_r(t) = \begin{bmatrix}\Delta_r(t) & \Gamma_r(t)\end{bmatrix}, r \in \mathcal{V}$, $\Phi(t) = \begin{bmatrix}\Delta(t) & \Gamma(t)\end{bmatrix}$ and $d_j = \tau_{j+1} - \tau_j$ for $j \in \mathcal{J}$ and let Assumptions 1-2 hold. Then, the following are true:*

$$(\mathbf{x}_r^j, \mathbf{u}_r^j) \in \overline{\mathcal{Z}}_r^j \Rightarrow \mathbf{x}_r(t; \tau_j, \mathbf{x}_r^j, \mathbf{u}_r) \in \mathcal{X}_r(t),\ \forall t \in \overline{T}_j \tag{8a}$$

$$(\mathbf{x}^j, \mathbf{u}^j) \in \overline{\mathcal{Z}}^j \Rightarrow \mathbf{x}(t; \tau_j, \mathbf{x}^j, \mathbf{u}) \in \mathcal{X}(t),\ \forall t \in \overline{T}_j \tag{8b}$$

where $\overline{\mathcal{Z}}_r^j := \bigcap_{w\in[0,d_j)}(\Phi_r(w))^{-1}(\bar{\mathcal{X}}_r(\tau_{j+1})), r \in \mathcal{V}$, $\overline{\mathcal{Z}}^j := \bigcap_{w\in[0,d_j)}(\Phi(w))^{-1}(\bar{\mathcal{X}}(\tau_{j+1}))$, $\bar{\mathcal{X}}_r(\tau_{j+1}) := \lim_{t\to\tau_{j+1}^-}\mathcal{X}_r(t)$ and $\bar{\mathcal{X}}(\tau_{j+1}) := \lim_{t\to\tau_{j+1}^-}\mathcal{X}(t)$.

Proof. Observe that constraints (5b)-(5c) and (5d) potentially become only a subset of the initial constraint set corresponding to the STL formulas defined in (3). Those constraints are gradually deactivated according to Definition 1 at time instants $\sigma \in \Sigma = \{\sigma_1 \leq \ldots \leq \sigma_p \leq \sigma_{p+1} \leq \ldots\}$. By design of (4), and due to the monotonicity of $\overline{\gamma}_i(t), i \in \mathcal{I}_r, r \in \mathcal{V}$ and $\gamma_i(t), i \in \mathcal{I}$ (discussed in Section II-B), the constraint sets have the following property: $\mathcal{X}_r(t_2) \subseteq \mathcal{X}_r(t_1)$ for every $r \in \mathcal{V}$ and $\mathcal{X}(t_2) \subseteq \mathcal{X}(t_1)$, where $\sigma_p \leq t_1 \leq t_2 < \sigma_{p+1}$. Observe that due to Assumption 2, the deactivation times $\sigma \in \Sigma$ are also sampling instants. We distinguish among two cases: i) $\tau_{j+1} \notin \bigcup_{i \in \mathcal{I}_r} \Sigma_i$ and ii) $\tau_{j+1} \in \bigcup_{i \in \mathcal{I}_r} \Sigma_i$. For case (i) note that $\bar{\mathcal{X}}_r(\tau_{j+1}) = \mathcal{X}_r(\tau_{j+1})$ due to the continuity of $\overline{\gamma}_i(t)$ and since the constraints determining $\mathcal{X}_r(\tau_{j+1} - \epsilon), \mathcal{X}_r(\tau_{j+1})$ are the same for a sufficiently small $\epsilon > 0$. In addition, for case (ii) it holds that $\bar{\mathcal{X}}_r(\tau_{j+1}) \subseteq \mathcal{X}_r(\tau_{j+1})$ as there exists at least one constraint that is removed from $\mathcal{X}_r(\tau_{j+1})$ at τ_{j+1} while for the rest it holds that $\lim_{t \to \tau_{j+1}^-} \overline{\gamma}_i(t) = \overline{\gamma}_i(\tau_{j+1})$ due to the continuity of $\overline{\gamma}_i(t)$. Nevertheless, in both cases, due to the property of the constraint sets mentioned above it follows that $\bar{\mathcal{X}}_r(\tau_{j+1}) \subseteq \mathcal{X}_t(t)$, for every $t \in \overline{T}_j$. Therefore, imposing the stricter condition $\mathbf{x}_r(t; \tau_j, \mathbf{x}_r^j, \mathbf{u}_r) \in \bar{\mathcal{X}}_r(\tau_{j+1})$, for every $t \in \overline{T}_j$ we can conclude that $\mathbf{x}_r(t; \tau_j, \mathbf{x}_r^j, \mathbf{u}_r) \in \mathcal{X}_r(t), t \in \overline{T}_j$. By (6) the constraint $\mathbf{x}_r(t; \tau_j, \mathbf{x}_r^j, \mathbf{u}_r) \in \bar{\mathcal{X}}_r(\tau_{j+1}), \forall t \in \overline{T}_j$ is equivalent to $\Delta_r(t - \tau_j)\mathbf{x}_r^j + \Gamma_r(t - \tau_j)\mathbf{u}_r^j \in \bar{\mathcal{X}}_r(\tau_{j+1}), \forall t \in \overline{T}_j$ which by definition of the pre-image and the fact that $\Phi_r(t) = \begin{bmatrix}\Delta_r(t) & \Gamma_r(t)\end{bmatrix}$ implies that $(\mathbf{x}_r^j, \mathbf{u}_r^j) \in (\Phi_r(t - \tau_j))^{-1}(\bar{\mathcal{X}}_r(\tau_{j+1})), \forall t \in \overline{T}_j$. The latter set of constraints is equivalent to $(\mathbf{x}_r^j, \mathbf{u}_r^j) \in \bigcap_{t \in \overline{T}_j} (\Phi_r(t - \tau_j))^{-1}(\bar{\mathcal{X}}_r(\tau_{j+1}))$ which results in (8a) after setting $w = t - \tau_j$. Similar arguments can be made for (8b). ∎

To further simplify the constraints we will consider polytopic overapproximations of $\Phi_r(w)$ over the intervals $[0, d_j)$.

Definition 2 ([16]). *Let $f : \mathbb{R} \to \mathbb{R}^n$. A polytope C is called a polytopic overapproximation of f over the interval $[0, d)$ if $f([0, d)) \subseteq C$.*

Lemma 2. *For every $j \in \mathcal{J}$, let $\mathcal{S}_r^j, r \in \mathcal{V}$ and $\mathcal{S}^j$ be any polytopic overapproximation of Φ_r and Φ over $[0, d_j)$ respectively. Define the sets $\mathcal{Z}_r^j = (\mathcal{S}_r^j)^{-1}(\bar{\mathcal{X}}_r(\tau_{j+1})), r \in \mathcal{V}$ and $\mathcal{Z}^j = (\mathcal{S}^j)^{-1}(\bar{\mathcal{X}}(\tau_{j+1}))$. Then, the following hold for all $w \in [0, d_j)$:*

$$(\mathbf{x}_r^j, \mathbf{u}_r^j) \in \mathcal{Z}_r^j \Rightarrow \Delta_r(w)\mathbf{x}_r^j + \Gamma_r(w)\mathbf{u}_r^j \in \bar{\mathcal{X}}_r(\tau_{j+1}),\ r \in \mathcal{V}$$
$$(\mathbf{x}^j, \mathbf{u}^j) \in \mathcal{Z}^j \Rightarrow \Delta(w)\mathbf{x}^j + \Gamma(w)\mathbf{u}^j \in \bar{\mathcal{X}}(\tau_{j+1}).$$

Proof. The proof follows similar arguments to [16, Lem. 1]. ∎

Polytopic overapproximations for functions of the form $\Delta_r(t), \Gamma_r(t)$ (or $\Delta(t), \Gamma(t)$) have been studied extensively e.g., in [17]. The interested reader may refer to [16], [17] and references therein for more details.

Remark 1. *Here, we consider aperiodic sampling in order to account for cases when the deactivation points in Σ do not have a common divisor. If a common divisor exists and periodic sampling is considered, then the the number of polytopic overapproximations per agent is reduced to one.*

Based on the above we may define the sampled-data MPC for the centralized system as follows:

$$\inf_{\mathbf{u}_r^k, r \in \mathcal{V}, k \in T_j'} \sum_{r \in \mathcal{V}} \sum_{k \in T_j'} L_r(\mathbf{u}_r^k, \mathbf{x}_r^k) \tag{9}$$

$$\mathbf{x}_r^{k+1} = A_r^k \mathbf{x}_r^k + B_r^k \mathbf{u}_r^k,\ \mathbf{x}_r^j = \overline{\mathbf{x}}_r^j,\ r \in \mathcal{V} \tag{9a}$$
$$(\mathbf{x}_r^k, \mathbf{u}_r^k) \in \mathcal{Z}_r^k,\ r \in \mathcal{V} \tag{9b}$$
$$(\mathbf{x}^k, \mathbf{u}^k) \in \mathcal{Z}^k, \tag{9c}$$
$$\mathbf{x}_r^{\mathfrak{p}(j)} \in \mathbb{X}_r^f(\tau_{\mathfrak{p}(j)}), r \in \mathcal{V} \tag{9d}$$
$$\mathbf{u}_r^k \in \mathbb{U}_r, \quad r \in \mathcal{V} \tag{9e}$$

for every $k \in T_j' := \{j, \ldots, \mathfrak{p}(j) - 1\}$, where $\mathcal{Z}_r^k, r \in \mathcal{V}, \mathcal{Z}^k$ are the sets defined in Lemma 2, $A_r^k := \Delta_r(d_k)$ and $B_r^k := \Gamma_r(d_k)$, for every $k \in T_j'$.

Proposition 1. *Let Assumptions 1-2 hold. If for a fixed $j \in \mathcal{J}_1$ and a given initial condition $\mathbf{x}^j = \overline{\mathbf{x}}^j$ the feasible set $\mathcal{U}_N(\overline{\mathbf{x}}^j) := \{\mathbf{u}^k, k \in T_j' :$ (9a)-(9e) are satisfied$\}$ is non-empty, then for a feasible control sequence $(\mathbf{u}^k, k \in T_j') \in \mathcal{U}_N(\overline{\mathbf{x}}^j)$, where $\mathbf{u}^k = \begin{bmatrix}\mathbf{u}_1^{kT} & \ldots & \mathbf{u}_R^{kT}\end{bmatrix}^T$, the following hold: 1) $\mathbf{x}_r(t; \tau_j, \overline{\mathbf{x}}_r^j, \overline{\mathbf{u}}_r(t)) \in \mathcal{X}_r(t), r \in \mathcal{V}$ and 2) $\mathbf{x}(t; \tau_j, \overline{\mathbf{x}}^j, \overline{\mathbf{u}}(t)) \in \mathcal{X}(t)$ for every $t \in T_j$, where $\overline{\mathbf{u}}_r(t) := \mathbf{u}_r^k$, for every $t \in \overline{T}_k, k \in T_j'$, and $\overline{\mathbf{x}}^j, \overline{\mathbf{u}}(t)$ are the stacked vectors of $\overline{\mathbf{x}}_r^j, \overline{\mathbf{u}}_r(t), r \in \mathcal{V}$ respectively.*

Proof. The proof follows from Lemmas 1-2 after noting that $\mathcal{Z}_r^k \subseteq \overline{\mathcal{Z}}_r^k, r \in \mathcal{V}$, and $\mathcal{Z}^k \subseteq \overline{\mathcal{Z}}^k$, for every $k \in T_j'$. ∎

IV. Distributed sampled-data MPC

In previous sections we focused on a centralized MPC problem formulation which is known for not scaling well with the number of the agents R. To that end, in this section we will design a sequential dMPC framework in which each agent is responsible for designing its own plan using state information from the other agents in the team. To facilitate the analysis later we pose the following assumption:

Assumption 3. *There exist control laws $\boldsymbol{\kappa}_r : \mathbb{R}^{n_r} \times \mathbb{R}_{\geq 0} \to \mathbb{U}_r, r \in \mathcal{V}$ such that the following hold for every $k \in \{0, \ldots, \mathfrak{p}(M_1)\}$:*

$$\boldsymbol{\kappa}_r(\mathbf{x}_r, t) = \overline{\mathbf{u}}_r,\ t \in \overline{T}_k, \mathbf{x}_r \in \mathbb{X}_r^f(\tau_k), r \in \mathcal{V}$$
$$(\mathbf{x}_r, \boldsymbol{\kappa}_r(\mathbf{x}_r, \tau_k)) \in \mathcal{Z}_r^k, \quad \mathbf{x}_r \in \mathbb{X}_r^f(\tau_k), r \in \mathcal{V}$$
$$(\mathbf{x}, \boldsymbol{\kappa}(\mathbf{x}, \tau_k)) \in \mathcal{Z}^k, \quad \mathbf{x} \in \mathbb{X}^f(\tau_k),$$
$$A_r^k \mathbf{x}_r + B_r^k \overline{\mathbf{u}}_r \in \mathbb{X}_r^f(\tau_{k+1}), \quad r \in \mathcal{V}$$

where $\mathbb{X}^f(\tau_k) := \prod_{r \in \mathcal{V}} \mathbb{X}_r^f(\tau_k)$, $\boldsymbol{\kappa}(\mathbf{x}, t)$ is the stacked vector of $\boldsymbol{\kappa}_r(\mathbf{x}_r, t), r \in \mathcal{V}$ and $\overline{\mathbf{u}}_r \in \mathbb{U}_r$ is a constant, known vector.

In the proposed dMPC framework the MPC problem corresponding to the r-th agent at time $\tau_j, j \in \mathcal{J}_1 \backslash \{0\}$ is defined as follows:

$$\inf_{\mathbf{u}_r^k, k \in T_j'} \sum_{k \in T_j'} L_r(\mathbf{u}_r^k, \mathbf{x}_r^k) \tag{10}$$

$$\mathbf{x}_r^{k+1} = A_r^k \mathbf{x}_r^k + B_r^k \mathbf{u}_r^k, \quad \mathbf{x}_r^j = \overline{\mathbf{x}}_r^j, \tag{10a}$$

$$(\mathbf{x}_r^k, \mathbf{u}_r^k) \in \mathcal{Z}_r^k, \tag{10b}$$
$$(\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k) \in \mathcal{Z}^k, \tag{10c}$$
$$\mathbf{x}_r^{\mathfrak{p}(j)} \in \mathbb{X}_r^f(\tau_{\mathfrak{p}(j)}), \tag{10d}$$
$$\mathbf{u}_r^k \in \mathbb{U}_r, \tag{10e}$$

where the constraints are evaluated for every $k \in T_j'$. Here, $\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k, k \in T_j'$ are defined for each agent r according to its order in the hierarchy as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}^k &= \begin{bmatrix}\mathbf{x}_1^{k,*T} & \ldots & \mathbf{x}_{r-1}^{k,*T} & \mathbf{x}_r^{kT} & \mathbf{x}_{r+1}^{k,-T} & \ldots & \mathbf{x}_R^{k,-T}\end{bmatrix}^T, \\
\tilde{\mathbf{u}}^k &= \begin{bmatrix}\mathbf{u}_1^{k,*T} & \ldots & \mathbf{u}_{r-1}^{k,*T} & \mathbf{u}_r^{kT} & \mathbf{u}_{r+1}^{k,-T} & \ldots & \mathbf{u}_R^{k,-T}\end{bmatrix}^T,
\end{aligned} \tag{11}$$

where $\mathbf{x}_r^{k,*}, \mathbf{u}_r^{k,*}$ denote the optimal state and control input of (10) at time step k computed at τ_j and $\mathbf{x}_r^{k,-}, \mathbf{u}_r^{k,-}$ the k-th optimal state and control input computed at $\tau_{j-1}, j \geq 1$. Note that for every $j \in \mathcal{J}_1 \backslash \{0\}$ and every $k \in \mathcal{J} \cap [\mathfrak{p}(j-1), \mathfrak{p}(j)-1]$, i.e., for all sampling instants $\tau_k, k \in \mathcal{J}$ between $\tau_{j-1}+N$ and $\tau_j + N$, the control input $\mathbf{u}_r^{k,-}$ is chosen as $\mathbf{u}_r^{k,-} = \boldsymbol{\kappa}_r(\mathbf{x}_r^{k,-}, \tau_k)$, where $\boldsymbol{\kappa}_r(\mathbf{x}_r, t), r \in \mathcal{V}$ is the terminal control input of Assumption 3. If $\mathfrak{p}(j) = \mathfrak{p}(j-1)+1$, i.e., when $\tau_{j-1}+N$ and τ_j+N are consecutive time instants within $\{\tau_k : k \in \mathcal{J}\}$, then $\mathbf{x}_r^{\mathfrak{p}(j),-}$ is chosen as the solution of the system defined in (10a) under the terminal control $\boldsymbol{\kappa}_r(\mathbf{x}_r^{\mathfrak{p}(j-1),-}, \tau_{\mathfrak{p}(j-1)})$. On the other hand, if due to the aperiodic sampling there exist τ_k such that $\tau_{j-1}+N < \tau_k < \tau_j + N$, then for every $k \in \mathcal{J} \cap [\mathfrak{p}(j-1)+1, \mathfrak{p}(j)-1]$, $\mathbf{x}_r^k$ is chosen as the solution of (10a) under the terminal control $\boldsymbol{\kappa}_r(\mathbf{x}_r^{k-1,-}, \tau_{k-1})$.

Here, we assume that agents are capable of communicating with a small subset of their peers. Let $G = (\mathcal{V}, \mathcal{E})$ be a static, directed communication graph with $(r', r) \in \mathcal{E}$ iff r' is capable of sending information to agent r. We make the following assumption on G :

Assumption 4. *The graph $G = (\mathcal{V}, \mathcal{E})$ is a static, cyclic graph with $\mathcal{E} := \{(r, r+1) : r \in \mathcal{V}\backslash\{R\}\} \cup \{(R, 1)\}$.*

Assumption 4 restricts the choice of the communication graph to one with a small number edges ensuring that agents receive the necessary information to solve (10). Other choices of graphs can also be made as long as each agent r receives state information for all other agents involved in the same STL tasks.

Motivated by [18], we propose solving the dMPC problems according to Algorithm 1. For $r \in \mathcal{V}$, let $\mathcal{V}_r := \{r' \in \mathcal{V} : r' > r\}$ and $\mathcal{V}^r := \mathcal{V} \backslash (\mathcal{V}_r \cup \{r\})$ be the downstream and upstream agents of the r-th agent, respectively. Note that for $r = 1$, $\mathcal{V}^1 = \emptyset$ holds. At $\tau_0 = 0$ we propose solving the centralized MPC problem (9). At future sampling instants $\tau_j, j \in \mathcal{J}_1 \backslash \{0\}$ the r-th agent receives from agent $r' \in \mathcal{N}_r$ the necessary information to create $(\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k), k \in T_j'$ and solves (10). Here, due to Assumption 4 the set $\mathcal{N}_r, r \in \mathcal{V}$ is defined as $\mathcal{N}_r := \{r-1\}$, if $r \in \mathcal{V}\backslash\{1\}$ or $\mathcal{N}_r := \{R\}$, otherwise. When everyone computes its optimal solution, all agents apply synchronously their first control input over $\overline{T}_j$ until the next state measurements become available and the procedure is repeated.

Algorithm 1: dMPC scheme

Input: Initial states $\overline{\mathbf{x}}_r^0, r \in \mathcal{V}$
for $j \in \mathcal{J}_1$ **do**
  if $j = 0$ **then**
    Solve the centralized MPC problem (9);
  else
    $r = 1$;
    while $r < R + 1$ **do**
      For every $k \in T_j'$, receive
        $(\mathbf{x}_r^{k,-}, \mathbf{u}_r^{k,-}), r \in \mathcal{V}_r$ and
        $(\mathbf{x}_r^{k,*}, \mathbf{u}_r^{k,*}), r \in \mathcal{V}^r$ from agent $r' \in \mathcal{N}_r$;
      Create $\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k, k \in T_j'$ based on (11);
      Solve the local MPC problem (10);
      $r \leftarrow r + 1$;
    end
  end
  Each $r \in \mathcal{V}$ applies $\mathbf{u}_r(t) = \mathbf{u}_r^{j*}$, for every $t \in \overline{T}_j$;
end

Theorem 1. *Let Assumptions 1-4 hold. If* (9) *is feasible at $\tau_0 = 0$, then the dMPC scheme of Algorithm 1 is recursively feasible. In addition, $\rho^\phi(\mathbf{x}, 0) \geq \bar{\rho}$.*

Proof. $\rho^\phi(\mathbf{x}, 0) \geq \bar{\rho}$ is ensured by design of $\mathcal{Z}_r^k, r \in \mathcal{V}$ and $\mathcal{Z}^k, k \in \{0, \ldots, \mathfrak{p}(M_1)\}$ (Lemma 2) provided that the proposed dMPC scheme is recursively feasible. The recursive feasibility property of the dMPC scheme will be shown by induction. Consider the following candidate control sequence for agent r: $\overline{\overline{\mathbf{u}}}_r = \begin{bmatrix}\mathbf{u}_r^{1,-T} & \ldots & \mathbf{u}_r^{\mathfrak{p}(0)-1,-T} & \hat{\mathbf{u}}_r^{\mathfrak{p}(0)} & \ldots & \hat{\mathbf{u}}_r^{\mathfrak{p}(1)-1}\end{bmatrix}^T$, where $\hat{\mathbf{u}}_r^k = \boldsymbol{\kappa}_r^T(\mathbf{x}_r^{k,-}, \tau_k), k \in \mathcal{J} \cap [\mathfrak{p}(0), \mathfrak{p}(1)-1]$. At τ_1 agent 1 receives the state and control information of the downstream agents. After setting $(\mathbf{x}_1^k, \mathbf{u}_1^k) = (\mathbf{x}_1^{k,-}, \mathbf{u}_1^{k,-})$ it defines $\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k$ for every $k \in \{1, \ldots, \mathfrak{p}(1)-1\}$. For $k \in \{1, \ldots, \mathfrak{p}(0)-1\}$ note that (10a)-(10c) and (10e) are satisfied with the proposed control sequence $\overline{\overline{\mathbf{u}}}_1$ by feasibility of (9) at τ_0. By Assumption 3, (10a)-(10c) and (10e) are satisfied for $k \in \mathcal{J} \cap [\mathfrak{p}(0), \mathfrak{p}(1)-1]$ and $\mathbf{x}_1^{\mathfrak{p}(1)} \in \mathbb{X}_1^f(\tau_{\mathfrak{p}(1)})$. Hence, $\overline{\overline{\mathbf{u}}}_1$ ensures feasibility of (10) when $r = 1$. Following similar arguments for $r > 1$ define $\tilde{\mathbf{x}}^k, \tilde{\mathbf{u}}^k$ for every $k \in \{1, \ldots, \mathfrak{p}(1)-1\}$ using the information sent by agent $r-1$ and after setting $(\mathbf{x}_r^k, \mathbf{u}_r^k) = (\mathbf{x}_r^{k,-}, \mathbf{u}_r^{k,-})$. Then, note that (10a)-(10c) and (10e) are satisfied for $k \in \{1, \ldots, \mathfrak{p}(0)-1\}$ due to feasibility of (9) and (10) for $r-1$. In addition, (10a)-(10c) and (10e) are satisfied for $k \in \mathcal{J} \cap [\mathfrak{p}(0), \mathfrak{p}(1)-1]$ and $\mathbf{x}_r^{\mathfrak{p}(1)} \in \mathbb{X}_r^f(\tau_{\mathfrak{p}(1)})$ by Assumption 3. Hence, $\overline{\overline{\mathbf{u}}}_r$ is feasible. Finally, note that feasibility at $\tau_{j+1}, j \in \mathcal{J}_1\backslash\{0\}$ given feasibility at τ_j can be shown similarly since the information agent 1 receives from the R-th agent at τ_{j+1} includes the optimal solutions of all agents at τ_j, i.e., it is analogous to the information $r = 1$ received at τ_0. ∎

V. Simulation Example

The proposed dMPC scheme is applied to a vehicle coordination scenario considering $R = 3$ vehicles modelled as double integrators, i.e., $\ddot{\mathbf{q}}_r = \mathbf{u}_r$, where $\mathbf{q}_r, \dot{\mathbf{q}}_r, \mathbf{u}_r \in \mathbb{R}^2$

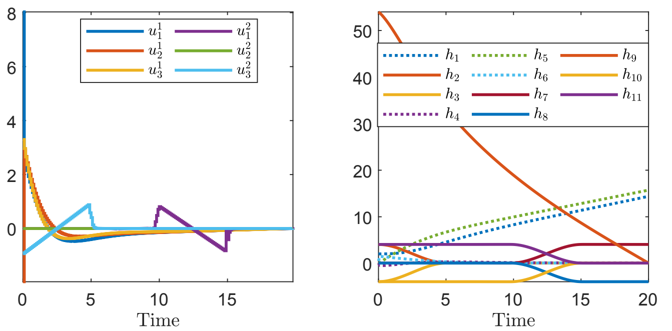

Fig. 1: Control inputs and evolution of the predicate functions with the proposed dMPC scheme.

is the position, velocity and acceleration of the r-th agent respectively and thus $\mathbf{x}_r = \begin{bmatrix} \mathbf{q}_r^T & \dot{\mathbf{q}}_r^T \end{bmatrix}^T$. The team is subject to the task $\phi = \bigwedge_{i=1}^{5} \varphi_i$, where $\varphi_1 = \mathcal{G}_{[5,20]} \bigwedge_{l=1}^{4} \mu_l$, $\varphi_2 = \mathcal{G}_{[0,20]}(\mu_5 \wedge \mu_6)$, $\varphi_3 = \mathcal{G}_{[0,10]}(\mu_7 \wedge \mu_8)$, $\varphi_4 = \mathcal{F}_{[18,20]}\mu_9$, $\varphi_5 = \mathcal{G}_{[15,20]}(\mu_{10} \wedge \mu_{11})$, where each predicate $\mu_l, l \in \{1, \dots, 11\}$ is evaluated by the corresponding predicate function $h_l(\mathbf{x})$ given by: $h_1(\mathbf{x}_2, \mathbf{x}_3) = q_2^1 - q_3^1 - d_s$, $h_2(\mathbf{x}_3) = q_3^2 - y_1 + \epsilon$, $h_3(\mathbf{x}_3) = -q_3^2 + y_1 + \epsilon$, $h_4(\mathbf{x}_2, \mathbf{x}_3) = \dot{q}_2^1 - \dot{q}_3^1 - d_v$, $h_5(\mathbf{x}_1, \mathbf{x}_2) = q_1^1 - q_2^1 - d_s$, $h_6(\mathbf{x}_1, \mathbf{x}_2) = \dot{q}_1^1 - \dot{q}_2^1 - d_v$, $h_7(\mathbf{x}_1) = q_1^2 - y_1 + \epsilon$, $h_8(\mathbf{x}_1) = -q_1^2 + y_1 + \epsilon$, $h_9(\mathbf{x}_1) = -q_1^1 + x_{\text{goal}} + d_s'$, $h_{10}(\mathbf{x}_1) = q_1^2 - y_2 + \epsilon$, $h_{11}(\mathbf{x}_1) = -q_1^2 + y_2 + \epsilon$, where $\mathbf{q}_r = \begin{bmatrix} q_r^1 & q_r^2 \end{bmatrix}^T$ and $\dot{\mathbf{q}}_r = \begin{bmatrix} \dot{q}_r^1 & \dot{q}_r^2 \end{bmatrix}^T$. The sampling period is chosen to be 0.1 time steps, $N = 2.9$ and $L_r(\mathbf{x}_r^k, \mathbf{u}_r^k) = \|\mathbf{u}_r^k\|_2^2$. Here, $\epsilon = 0.06$, $y_1 = 2, y_2 = 6$, $d_s = 3, d_v = 0.5$, $d_s' = 2$ and $x_{\text{goal}} = 60$. We set $\mathbb{X}_r = [0, 100] \times [0, 8] \times [0, 10] \times [-10, 10]$, for every $r \in \mathcal{V} \backslash \{2\}$, $\mathbb{X}_2 = [0, 100] \times [0, 8] \times [0, 10] \times \{0\}$, and $\mathbb{U}_r = [-10, 10]^2, r \in \mathcal{V}$ and enforce the satisfaction of ϕ with a minimum robustness $\bar{\rho} = 0.05$ choosing $t_i^* = b_i$ for every eventually formula $\varphi_i, i \in \overline{\mathcal{I}}$. The polytopic overapproximations over $[0, 0.1)$ are computed using the real Jordan form of A_r as in [17]. The initial states of the agents are given as follows: $\mathbf{x}_1(0) = \begin{bmatrix} 8 & 2 & 2 & 0 \end{bmatrix}^T$, $\mathbf{x}_2(0) = \begin{bmatrix} 5 & 2 & 1 & 0 \end{bmatrix}^T$ and $\mathbf{x}_3(0) = \begin{bmatrix} 0 & 6 & 0.45 & 0 \end{bmatrix}^T$. In Figure 1 the control inputs and evolution of the predicate functions over the horizon of the formula are given with $h_l(\mathbf{x}) \geq 0.05, l \in \{1, \dots, 11\}$ over the desired time intervals. Despite the small number of agents a significant decrease on the computational time as well as similar performance to to the centralized MPC problem (cMPC) has been observed. In particular the average computational time of the proposed dMPC for agents $1, 2, 3$ is 0.12, 0.05 and 0.09 s, respectively, as opposed to 0.72s of the cMPC while the maximum absolute difference of the predicate function values over $i \in \overline{\mathcal{I}}$ and $t \in [0, 20]$ when evaluated using the solutions of the dMPC and cMPC is $1.6 \cdot 10^{-3}$. All computations were performed on an Intel Core i7-8665U with 16GB RAM using `quadprog` in MATLAB.

VI. Conclusions

In this work a sampled-data distributed MPC problem is presented for linear systems subject to coupled STL tasks. The local MPC problems are solved sequentially and the recursive feasibility property of the framework is guaranteed. The resulting controllers are piecewise constant and the closed loop trajectories ensure the satisfaction of the STL task in continuous time. Future work will focus on the design of a parallel MPC scheme for more complex nonlinear systems.

References

[1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. FTRTFT 2004, FORMATS 2004*, Y. Lakhnech and S. Yovine, Eds., vol. 3253, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2004, pp. 152–166.

[2] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Formal Modeling and Analysis of Timed Systems. FORMATS 2010*, K. Chatterjee and T. Henzinger, Eds., vol. 6246, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, pp. 92–106.

[3] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *IEEE Conference on Decision and Control*, 2014, pp. 81–87.

[4] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, 2015, pp. 772–779.

[5] Y. E. Sahin, R. Quirynen and S. D. Cairano, "Autonomous vehicle decision-making and monitoring based on signal temporal logic and mixed-integer programming," in *American Control Conference*, 2020, pp. 454–459.

[6] A. Rodionova, L. Lindemann, M. Morari and G. J. Pappas, "Time-robust control for stl specifications," in *60th IEEE Conference on Decision and Control*, 2021, pp. 572–579.

[7] G. Yang, C. Belta, and R. Tron, "Continuous-time signal temporal logic planning with control barrier functions," in *American Control Conference*, 2020, pp. 4612–4618.

[8] K. Garg and D. Panagou, "Control-lyapunov and control-barrier functions based quadratic program for spatio-temporal specifications," in *IEEE Conference on Decision and Control*, 2019, pp. 1422–1429.

[9] W. Xiao, C. A. Belta, and C. G. Cassandras, "High order control lyapunov-barrier functions for temporal logic specifications," in *American Control Conference*, 2021, pp. 4886–4891.

[10] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2018.

[11] Z. Liu, B. Wu, J. Dai, and H. Lin, "Distributed communication-aware motion planning for multi-agent systems from stl and spatel specifications," in *IEEE Conference on Decision and Control*, 2017, pp. 4452–4457.

[12] A. T. Büyükkoçak, D. Aksaray, and Y. Yazicioglu, "Distributed planning of multi-agent systems with coupled temporal logic specifications," in *AIAA Scitech 2021 Forum*, 2021.

[13] X. Zhou and Y. Zou and S. Li and X. Li and H. Fang, "Distributed model predictive control for multi-robot systems with conflicting signal temporal logic tasks," *IET Control Theory & Applications*, vol. 16, no. 5, pp. 554–572, 2022.

[14] L. Lindemann and D. V. Dimarogonas, "Barrier function-based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[15] M. Charitidou, D. V. Dimarogonas, "Receding horizon control with online barrier function design under signal temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3545–3556, 2023.

[16] P. Sopasakis, P. Patrinos, and H. Sarimveis, "Mpc for sampled-data linear systems: Guaranteeing constraint satisfaction in continuous-time," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 1088–1093, 2014.

[17] W. P. M. H. Heemels and N. van de Wouw and R. H. Gielen and M. Donkers, L. Hetel and S. Olaru and M. Lazar and J. Daafouz, and S. Niculescu, "Comparison of overapproximation methods for stability analysis of networked control systems," in *ACM International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 181–190.

[18] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, 2007.