Cooperative Object Manipulation under Signal Temporal Logic Tasks and Uncertain Dynamics*

Mayank Sewlia[1], Christos K. Verginis[2], and Dimos V. Dimarogonas[1]

Abstract— **We address the problem of cooperative manipulation of an object whose tasks are specified by a Signal Temporal Logic (STL) formula. We employ the Prescribed Performance Control (PPC) methodology to guarantee predefined transient and steady-state performance on the object trajectory in order to satisfy the STL formula. More specifically, we first provide a way that translates the problem of satisfaction of an STL task to the problem of state evolution within a user-defined time-varying funnel. We then design a control strategy for the robotic agents that guarantees compliance with this funnel. The control strategy is decentralized, in the sense that each agent calculates its own control signal, and does not use any information on the agents' and object's dynamic terms, which are assumed to be unknown. We experimentally verify the results on two manipulator arms, cooperatively working to manipulate an object based on a STL formula.**

I. Introduction

The cooperative-manipulation problem considers the control of multiple robots towards the manipulation of a grasped object. Robots working in such cooperation have the advantage of sharing the load, and are hence able to manipulate heavier objects than in single-robot setups [1]. Cooperative manipulation systems find applications in human-robot interaction, aerial lifting of objects, transportation in cluttered environments, and industrial automation works.

Unlike the related literature, which mainly considers the tracking/regulation of the manipulated object, we are here interested in more complex tasks over time, such as "never take the object to dangerous regions" or "keep moving the object from region A to B within a predefined time interval", which must be executed via the control actions of the agents. Such complex tasks can be expressed by temporal logics. Examples include Linear Temporal Logic (LTL) [2], Metric-interval Temporal Logic (MITL) [3] and Signal Temporal Logic (STL); LTL and MITL use automata representations of the task and require discrete abstractions of the underlying continuous systems; STL, on the other hand, introduced in [4], is a formalism used to specify properties of dense-time, real-valued signals. It allows for inclusion of time and state bounds of a continuous-time system and hence a useful formalism to model complex tasks in robotics [5]. Moreover, STL offers a degree of space robustness, allowing us to measure the extent of satisfaction of a signal against a given formula [6].

This paper addresses the cooperative manipulation problem under STL tasks for systems with uncertain dynamics. Our main contribution is the design of a control mechanism that guarantees that satisfaction of an STL-encoded task for a cooperatively manipulated object. Firstly, we translate the problem of STL satisfaction to the problem of state evolution within a time-varying funnel. Next, we employ the Prescribed Performance Control (PPC) methodology [7] to guarantee compliance with this funnel and, consequently, satisfaction of the task. The control mechanism exhibits the following attributes. Firstly, it is decentralized in the sense that each agent calculates its own control signal, without needing to communicate with the rest of the team. Secondly, it does not use any information on the dynamics of the robotic agents and the object or any related estimation/approximation models. Thirdly, it does not require the use of force/torque sensors at the grasping points among the agents and the object. Finally, we avoid the use of approximations in robustness metrics of STL and consider a richer STL fragment compared to previous works. To the best of our knowledge, there have been no other works addressing the cooperative manipulation problem with STL constraints and uncertain dynamics.

Many works consider STL tasks for multi-agent systems [8], but without addressing the special case of cooperative manipulation schemes. The latter exhibit unique challenges due to their complex dynamics and the interaction of the robotic agents through their grasping points with the object. On the other hand, decentralized control of cooperative manipulators is a widely-studied problem in the literature [1], [9]–[13]. However, most related works study the problem of trajectory tracking for the object, possibly with simultaneous force regulation and state constraints, such as collision avoidance. On the contrary, we focus on the cooperative manipulation of an object such that it satisfies a more general task given by an STL formula. Works such as [14] and [2] use LTL to achieve multi-robot planning. However there are certain inherent differences between Linear Temporal Logic(LTL) and Signal Temporal Logic(STL). STL is defined over continuous space and time, and is accompanied by a robustness metric signifying the degree of satisfaction of a formula. In contrast, LTL is a proposition based logic evaluated over infinite sequence of states without time constraints. Temporal logics were also used in [3] for cooperative manipulation schemes in the form of MITL constraints, resorting however to workspace discretization and abstraction techniques, which we avoid in this work. MITL

*This work was supported by the ERC CoG LEAFHOUND, the Swedish Research Council (VR), the Knut och Alice Wallenberg Foundation (KAW) and the H2020 European Project CANOPIES.

M. Sewlia and D. V. Dimarogonas are with Division of Decision and Control, School of EECS, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden. {sewlia, dimos}@kth.se

C. K. Verginis is with the Division of Signals and Systems, Department of Electrical Engineering, Uppsala University, Uppsala, Sweden. christos.verginis@angstrom.uu

is a proposition based logic that allows timing constraints on temporal operators but fails to define behaviour of continuous signals in space and time. In [15], a high-level and a low-level task planner are used to satisfy a metric temporal logic specification but are limited to manipulation with a single robotic arm. Compared to [16], which studies prescribed performance control for STL tasks, we consider a richer STL fragment, provide a model-free controller, and deal with conjunctions in a non-hybrid control framework.

The rest of the paper is organized as follows. Section II presents preliminaries and notation used in the rest of the work. The problem formulation along with the system model are given in Section III and in Section IV we design the controller using the PPC method. Section V presents experimental verification of the theory developed and conclusions are laid out in Section VI.

II. Notation and Preliminaries

A. Notation

The set of natural numbers is denoted by $\mathbb{N}$ and the set of real numbers by $\mathbb{R}$. With $n \in \mathbb{N}$, $\mathbb{R}^n$ is the set of n-coordinate real-valued vectors and $\mathbb{R}^n_{\geq 0}$ is the set of real n-vector with non-negative elements; I_n represents the $n \times n$ identity matrix and $0_{n\times m}$ represents the $n \times m$ sized matrix with all zero entries. For $a, b \in \mathbb{R}^3$, $S(a)$ denotes a skew-symmetric matrix defined as $S(a)b = a \times b$, where $\times$ is the vector-cross product operator; C^1 represents the class of continuously differentiable functions. We further use $\mathbb{M} = \mathbb{R}^3 \times \mathbb{T}^3$, where $\mathbb{T}^3$ is the 3D torus, and $\mathbb{B} = \{\top, \perp\}$ (True, False). Finally, we use $A = \text{blckdiag}\{A_1, A_2, \dots, A_N\}$ to denote the block diagonal matrix with matrices $A_i \in \mathbb{R}^{p\times p}$ on its diagonal, where $N, p \in \mathbb{N}$.

B. Signal Temporal Logic (STL)

Let $\mathbf{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be a continuous-time signal. Signal Temporal Logic [4] is a predicate-based logic with the following syntax,

$$\phi = \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_{[a,b]} \phi_2 \tag{1}$$

where ϕ_1, ϕ_2 are STL formulae and μ is a predicate of the form $\mu : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{B}$ defined via a predicate function $p : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ as

$$\mu = \begin{cases} \top & p(\mathbf{x}, t) \geq 0 \\ \perp & p(\mathbf{x}, t) < 0 \end{cases}. \tag{2}$$

The satisfaction relation $(\mathbf{x}, t) \models \phi$ indicates that signal $\mathbf{x}$ satisfies ϕ at time t and is defined recursively as follows:

$$\begin{aligned}
&(\mathbf{x}, t) \models \mu && \Leftrightarrow p(\mathbf{x}, t) \geq 0 \\
&(\mathbf{x}, t) \models \neg\phi && \Leftrightarrow \neg((\mathbf{x}, t) \models \phi) \\
&(\mathbf{x}, t) \models \phi_1 \wedge \phi_2 && \Leftrightarrow (\mathbf{x}, t) \models \phi_2 \wedge (\mathbf{x}, t) \models \phi_2 \\
&(\mathbf{x}, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2 && \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (\mathbf{x}, t_1) \models \phi_2 \\
& && \quad \wedge \forall t_2 \in [t, t_1], (\mathbf{x}, t_2) \models \phi_1.
\end{aligned}$$

A signal $\mathbf{x}$ satisfies the *Until* operator, $(\mathbf{x}, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2$, if ϕ_1 holds at all times before ϕ_2 holds and ϕ_2 holds at some time instance between a and b.

The robust semantics for STL are defined by a real-valued function $\rho^\phi(\mathbf{x}, t)$. In particular, it holds that $(\mathbf{x}, t) \models \phi$ if $\rho^\phi(\mathbf{x}, t) > 0$, where the value of $\rho^\phi(\mathbf{x}, t) = p(\mathbf{x}(t), t)$ indicates the level of satisfaction of the signal $\mathbf{x}(t)$ at time t. The STL syntax we consider here is

$$\psi ::= \top \mid \mu \mid \neg\mu \tag{3a}$$

$$\phi ::= \psi \mid \mathcal{G}_{[a,b]}\psi \mid \mathcal{F}_{[a,b]}\psi \mid \psi_1 \mathcal{U}_{[a,b]} \psi_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \tag{3b}$$

where ψ, ψ_1 and ψ_2 are formulae of class ψ given in (3a), ϕ_1, ϕ_2 are formulae of class ϕ given in (3b) and $a, b \in \mathbb{R}_{\geq 0}$ with $a \leq b$. The following temporal operators can be defined from (1); *eventually*: $\mathcal{F}_{[a,b]}\psi \equiv \top \mathcal{U}_{[a,b]} \psi$, *always*: $\mathcal{G}_{[a,b]}\psi \equiv \neg\mathcal{F}_{[a,b]}\neg\psi$ and *disjunction*: $\phi_1 \vee \phi_2 \equiv \neg(\neg\phi_1 \wedge \neg\phi_2)$.

More details on STL semantics can be found in [8]. In the following, each part of ϕ that corresponds to a different predicate μ (and a predicate function), along with the respective temporal operator, is called a *sub-formula*.

C. Prescribed Performance Control

The concepts and techniques of Prescribed Performance Control (PPC), introduced in [7], are used in this work to achieve an STL task for a cooperatively manipulated object; PPC describes the behavior where a tracking error $e(t)$ evolves strictly within a predefined region that is bounded by certain functions of time, achieving prescribed transient- and steady-state performance. The mathematical expression of prescribed performance is given by the inequality $-\gamma(t) < e(t) < \gamma(t)$, where $\gamma(t)$ is a smooth and bounded decaying function of time satisfying $\lim_{t\to\infty} \gamma(t) > 0$, called *performance function*. We focus on the special case of exponential performance functions $\gamma(t) = (\gamma_0 - \gamma_\infty)\exp(-lt) + \gamma_\infty$, with $\gamma_0 \geq \gamma_\infty$ and $l \in \mathbb{R}_{\geq 0}$ appropriately chosen constants. More specifically, the constant $\gamma_0 = \gamma(0)$ is selected such that $\gamma_0 > |e(0)|$, and the constant $\gamma_\infty = \lim_{t\to\infty} \gamma(t)$ represents the maximum allowable size of the tracking error $e(t)$ at steady state, which may be set arbitrarily small to a value reflecting the resolution of the measurement device, thus achieving practical convergence of $e(t)$ to zero. Moreover, the decreasing rate of $\gamma(t)$, which is affected by the constant l in this case, introduces a lower bound on the required speed of convergence of $e(t)$. Therefore, the appropriate selection of the performance function imposes performance characteristics on the tracking error $e(t)$.

III. Problem Formulation

We begin by providing the model of the cooperative manipulation problem consisting of the dynamics of the agents and the object. We provide now the problem setup. Consider a workspace $\mathcal{W} \subset \mathbb{R}^3$ with N robotic agents rigidly grasping an object as seen in Figure 1, with $\mathcal{N} = \{1, \dots, N\}$. The agent's reference frame is denoted by $\{A_i\}$, the frames of end-effectors as $\{E_i\}$ for $i = 1, 2, \dots, N$, the inertial frame by $\{I\}$ and the frame corresponding to object's center of mass as $\{O\}$. Each agent is assumed to know the fixed distance $p_{i/o}$ from $\{O\}$ to $\{E_i\}$ in the $\{I\}$ frame. The agents are considered to be robotic arms, possibly mounted on mobile platforms that have access to the whole workspace. The arms are assumed to grasp rigidly the object thereby being able to exert forces/torques along every direction.

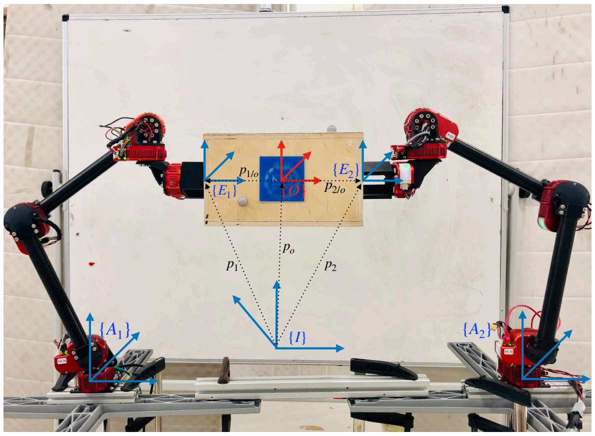


Fig. 1: Two arms rigidly grasping an object at the Smart Mobility Lab (SML), KTH.

We describe next the agents' and object's dynamics.

1) Agent dynamics: Let $q_i \in \mathbb{R}^{n_i}$ denote the joint-space variables of agent $i \in \mathcal{N}$, with $n_i \geq 6$, and $p_i \in \mathbb{R}^3$, $\eta_i \in \mathbb{T}^3$ the position and Euler-angle orientation of agent i's end-effector. Moreover, let $v_i \in \mathbb{R}^6$ be the velocity of agent i's end-effector given by $v_i = [\dot{p}_i^\top, \omega_i^\top]^\top$, where $\omega_i \in \mathbb{R}^3$ is the respective angular velocity; v_i is related to q_i via the agent's kinematic Jacobian matrix $J_i : \mathbb{R}^{n_i} \times \mathbb{R}^{6\times n_i}$ as $v_i = J_i(q_i)\dot{q}_i$. We assume that the agents operate away from kinematic singularities (where $J_i(q_i)$ loses rank), which leads to the well-defined task-space agent dynamics [17]

$$M_i(q_i)\dot{v}_i + C_i(q_i, \dot{q}_i)v_i + g_i(q_i) + w_i(q_i, \dot{q}_i, t) = u_i - \lambda_i \quad (4)$$

where $M_i : \mathbb{R}^{n_i} \to \mathbb{R}^{6\times 6}$ is the task-space positive definite mass matrix, $C_i : \mathbb{R}^{2n_i} \to \mathbb{R}^{6\times 6}$ the task-space Coriolis matrix, $g_i : \mathbb{R}^{n_i} \to \mathbb{R}^6$ the task-space gravity vector, $w_i : \mathbb{R}^{2n_i} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ represents model uncertainties and bounded external disturbances, $u_i \in \mathbb{R}^6$ is the task-space input wrench and $\lambda_i \in \mathbb{R}^6$ is the generalized force vector that agent i exerts on the object. The vector fields $M_i(\cdot)$, $C_i(\cdot)$, $g_i(\cdot)$, are continuously differentiable [17], and we further assume that $w_i(q_i, \dot{q}_i, t)$ is continuous in q_i and $\dot{q}_i$ for each fixed $t \geq 0$, and uniformly bounded in t for each fixed $(q_i, \dot{q}_i) \in \mathbb{R}^{2n_i}$. Nevertheless, in this work we assume that all the dynamic terms $M_i(\cdot)$, $C_i(\cdot)$, $g_i(\cdot)$, $w_i(\cdot)$ are *unknown* and cannot be used in the control design. Similarly, we do not employ any force-related sensor unit and hence λ_i is also unknown to agent i. The only information that agent i has access to is its joint variables $(q_i, \dot{q}_i)$, from which it can derive its end-effector's pose (p_i, η_i) and velocity v_i through the forward kinematics, $i \in \mathcal{N}$. The dynamics (4) can be written in vector form as:

$$M(q)\dot{v} + C(q, \dot{q})v + g(q) + w(q, \dot{q}, t) = u - \lambda, \quad (5)$$

where $v = [v_1^\top, \ldots, v_N^\top] \in \mathbb{R}^{6N}$, $M = \text{diag}\{[M_i]\} \in \mathbb{R}^{6N\times 6N}$, $C = \text{diag}\{[C_i]\} \in \mathbb{R}^{6N\times 6N}$, $g = [g_1^\top, \ldots, g_N^\top]^\top \in \mathbb{R}^{6N}$, $w = [w_1^\top, \ldots, w_N^\top]^\top \in \mathbb{R}^{6N}$, $u = [u_1^\top, \ldots, u_N^\top]^\top \in \mathbb{R}^{6N}$ and $\lambda = [\lambda_1^\top, \ldots, \lambda_N^\top]^\top \in \mathbb{R}^{6N}$.

2) Object dynamics: The pose and velocity of the object's center of mass are denoted by $\mathbf{x}_0 = [p_o^\top, \eta_o^\top]^\top \in \mathbb{M}$ and $v_o = [\dot{p}_o^\top, \omega_o^\top]^\top \in \mathbb{R}^6$, respectively, where $p_o \in \mathbb{R}^3$, $\eta_o \in \mathbb{R}^3$, and $\omega_o \in \mathbb{R}^3$ are the position, Euler-angle orientation, and angular velocity of the object's center of mass. The dynamics of the object are given by

$$\dot{\mathbf{x}}_o = J_{o_r}^{-1}(\mathbf{x}_o)v_o, \quad (6a)$$

$$M_o(\mathbf{x}_o)\dot{v}_o + C_o(\mathbf{x}_o, v_o)v_o + g_o(\mathbf{x}_o) + w_o(t) = \lambda_o \quad (6b)$$

where $M_o : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is the positive-definite inertia matrix, $C_o : \mathbb{M} \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ is the Coriolis matrix, $g_o : \mathbb{M} \to \mathbb{R}^6$ is the gravity vector and $w_o : \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ is a bounded vector of external disturbances. Additionally, $J_{o_r} : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is the object representation Jacobian [3] and $\lambda_o \in \mathbb{R}^6$ is the force vector acting on the object's center of mass. The terms $M_o(\cdot)$, $C_o(\cdot)$, $g_o(\cdot)$ are continuously differentiable and we assume that $w_o(t)$ is bounded. The matrix J_{o_r} and its inverse are well-defined when the object's pitch angle θ_o (second entry of η_o) evolves in the set $(-\frac{\pi}{2}, \frac{\pi}{2})$, which is assumed to be the case. Additionally and similarly to the agents, the object's dynamic terms $M_o(\cdot)$, $C_o(\cdot)$, $g_o(\cdot)$, $w_o(\cdot)$ and force λ_o are considered to be unknown and cannot be used in the control design.

3) Coupled dynamics: The grasping rigidity gives rise to the geometric relations

$$p_i = p_o + p_{i/o}, \quad \eta_i = \eta_o + \eta_{i/o} \quad (7)$$

where $p_{i/o}$ and $\eta_{i/o}$ are the constant position and orientation offsets among the ith end-effector and the object's center of mass. By differentiating (7), we obtain

$$v_i = J_{o_i}(q_i)v_o \quad (8)$$

where $J_{o_i} : \mathbb{R}^{n_i} \to \mathbb{R}^{6\times 6}$ is the object-to-i'th agent Jacobian

$$J_{o_i}(q_i) = \begin{bmatrix} I_3 & S(-p_{i/o}(q_i)) \\ 0_{3\times 3} & I_3 \end{bmatrix} \quad (9)$$

for all $i \in \mathcal{N}$, which is invertible due to the grasp rigidity. Note that, by inverting (7) and (8), agent i can derive the pose and velocity of the object. The coupled dynamics, derived in [3], are given by

$$\tilde{M}(q)\dot{v}_o + \tilde{C}(q, \dot{q})v_o + \tilde{g}(q, \dot{q}) + \tilde{w}(q, t) = G^\top(q)u \quad (10)$$

where $\tilde{M}(q) = M_o(q) + G^\top(q)M(q)G(q)$, $\tilde{C}(q, \dot{q}) = C_o(q, \dot{q}) + G^\top(q)M(q)\dot{G}(q) + G^\top(q)C(q, \dot{q})G(q)$, $\tilde{g}(q, \dot{q}) = g_o(q) + G^\top(q)g(q)$ and $\tilde{w}(q, t) = w_o(t) + G^\top(q)w(q, t)$. $\tilde{M}$ being positive definite, and $G : \mathbb{R}^n \to \mathbb{R}^{6N\times 6}$ is the system's grasp matrix, given by $G(q) = [J_{o_1}^\top(q_1), \ldots, J_{o_N}^\top(q_N)]^\top$. We note that G has full column rank owing to the grasp rigidity.

The problem at hand is the design of a decentralized control policy for u_i, $i \in \mathcal{N}$, in (4), such that the object satisfies a user-defined STL task. More formally:

Problem 1: Consider a system of N agents rigidly grasping an object with unknown coupled dynamics (6), (10). Let an STL formula ϕ (as in (3)) over the object trajectory $\mathbf{x}_o(t)$. Design a decentralized control policy for u_i, $i \in \mathcal{N}$, in (4), that leads to the satisfaction of φ, i.e., $(\mathbf{x}_o, t) \models \varphi$.

IV. Main Results

This section provides the main results of this work. In Section IV-A, we first provide a mechanism to encode the STL task in performance functions. In Section IV-B, we design a decentralized controller that guarantees evolution of the states inside the performance functions ensuring the satisfaction of the STL task.

A. Encoding STL tasks as PPC constraints

As mentioned in Section II-B, an STL formula ϕ consists of temporal operators along with predicates μ (see (3)), which are encoded via C^1 functions $p(\mathbf{x}_o, t)$. In this work, we exploit the PPC formulation and appropriately map these predicate functions $p(\mathbf{x}_o, t)$ to C^1 functions $h(\mathbf{x}_o, t)$, such that $\mu = \top$ if and only if $-\gamma_h(t) < h(\mathbf{x}_o(t), t) < \gamma_h(t)$, where $\gamma_h(t)$ is a user-defined performance function; that is, the predicate satisfaction is equivalent to the containment of $h(\mathbf{x}_o(t), t)$ in the interval $(-\gamma_h(t), \gamma_h(t))$.

We now describe in more detail the construction of the performance and predicate functions for the fragment of the form (3b). The STL formula at hand can be written as $\phi = \phi_1 \vee \phi_2 \vee \cdots \vee \phi_R$, where each ϕ_i is a conjunction $\phi_i = \phi_{i,1} \wedge \phi_{i,2} \wedge \cdots \wedge \phi_{i,M_i}$, and each $\phi_{i,j}$ is of the form

$$\psi \mid \mathcal{G}_{[a,b]}\psi \mid \mathcal{F}_{[a,b]}\psi, \tag{11}$$

for all $j \in \{1, \ldots, M_i\}$, $i \in \{1, \ldots, R\}$ for positive constants M_i and R and where ψ is a formula of the form (3a). Here, R represents the number of disjunction components and M_i represents the number of conjunction components corresponding to each i-th disjunction component. The until operator $\psi_1 \mathcal{U}_{[a,b]} \psi_2$ is equivalent to the conjunction $\mathcal{F}_{[t_1]}\psi_2 \wedge \mathcal{G}_{[a,t_1]}\psi_1$, with $t_1 \in (a, b]$, and is hence incorporated in the aforementioned formulation.

We consider a bottom-up approach and consider first a formula $\phi_{i,j}$ of the form (11). For each $\phi_{i,j}$, $i \in \{1, \ldots, R\}$, $j \in \{1, \ldots, M_i\}$, if $\phi_{i,j} = \mathcal{G}_{[a_{i,j}, b_{i,j}]}\psi_{i,j}$ or $\phi_{i,j} = \mathcal{F}_{[a_{i,j}, b_{i,j}]}\psi_{i,j}$, then we opt to activate $\phi_{i,j}$ only in the specific time interval $[a_{i,j}, b_{i,j}]$. More specifically, we define smooth switching functions $\beta_{i,j}(t)$ such that

$$\beta_{i,j}(t) = \begin{cases} 0, & t < t^\star_{i,j} - \delta \\ 1, & t \in [t^\star_{i,j}, b_{i,j}] \\ 0, & t > b_{i,j} + \delta, \end{cases} \tag{12}$$

where $\delta > 0$ is the "rise and fall" time, and we set $\beta_{i,j}(t) = 1$ if $\phi_{i,j} = \psi_{i,j}$ (i.e., there are no temporal constraints). The time instants $t^\star_{i,j} \in \mathbb{R}_{\geq 0}$ are chosen a priori as

$$t^\star_{i,j} \in \begin{cases} [a_{i,j}] & \text{if } \phi_{i,j} = \mathcal{G}_{[a_{i,j}, b_{i,j}]}\psi \\ [a_{i,j}, b_{i,j}] & \text{if } \phi_{i,j} = \mathcal{F}_{[a_{i,j}, b_{i,j}]}\psi. \end{cases}$$

In the experiments, $t^\star_{i,j}$ for the eventually($\mathcal{F}$) operator is chosen randomly in the interval $(a_{i,j}, b_{i,j}]$. Next, let the predicates $\mu_{i,j}$ and the respective predicate functions $p_{i,j}(\mathbf{x}_o, t)$ that correspond to $\psi_{i,j}$ of the subformulae (11). We map $p_{i,j}(\mathbf{x}_o, t)$ to C^1 functions $h_{i,j}(\mathbf{x}_o, t)$ such that

$$\mu_{i,j} = \begin{cases} \top, & -\gamma_{h_{i,j}}(t) < \bar{h}_{i,j}(\mathbf{x}_o, t) < \gamma_{h_{i,j}}(t), \\ \bot, & \text{otherwise} \end{cases}$$

which implies

$$p_{i,j}(\mathbf{x}(t), t) \geq 0 \Leftrightarrow -\gamma_{h_{i,j}}(t) < \bar{h}_{i,j}(\mathbf{x}_o(t), t) < \gamma_{h_{i,j}}(t), \tag{13}$$

where

$$\bar{h}_{i,j}(\mathbf{x}_0, t) = \beta_{i,j}(t) h_{i,j}(\mathbf{x}_0, t). \tag{14}$$

Here, $\gamma_{h_{i,j}}(t)$ are appropriately chosen performance functions of the form $\gamma_{h_{i,j}}(t) = (\gamma^0_{h_{i,j}} - \gamma^\infty_{h_{i,j}}) \exp(-l_{i,j} t) + \gamma^\infty_{h_{i,j}}$ (see Section II-C), and the incorporation of $\beta_{i,j}$ in $\bar{h}_{i,j}$ accommodates the local activation of the temporal subformulae of the form $\mathcal{G}_{[a_{i,j}, b_{i,j}]}\psi_{i,j}, \mathcal{F}_{[a_{i,j}, b_{i,j}]}\psi_{i,j}$; intuitively, we aim to force $h_{i,j}(\mathbf{x}_o, t)$ to evolve inside the funnels prescribed by $\gamma_{h_{i,j}}(t)$, such that an appropriate choice of $\gamma_{h_{i,j}}(t)$ will yield the satisfaction of the STL task. Apart from achieving satisfaction of the predicates, the PPC formulation allows for greater control over the rate of convergence and robustness of $h_{i,j}(\mathbf{x}_o, t)$, since the latter are explicitly shaped by the user-defined performance functions $\gamma_{h_{i,j}}(t)$. These functions also accommodate the temporal constraints imposed by the operators $\mathcal{G}_{[a_{i,j}, b_{i,j}]}, \mathcal{F}_{[a_{i,j}, b_{i,j}]}$. For instance, let $\phi_{i,j} = \mathcal{G}_{[5,10]}\Big(\|\mathbf{x}_o(t)\| < \epsilon\Big)$. Then, $p_{i,j}(\mathbf{x}_o, t) = \epsilon - \|\mathbf{x}_o\|$, $h_{i,j}(\mathbf{x}_o, t) = \|\mathbf{x}_o\|$, and we choose $\gamma_{h_{i,j}}$ such that $\gamma_{h_{i,j}}(5) = \epsilon$, implying $\gamma_{h_{i,j}}(t) < \epsilon$, for all $t > 5$. Note that, subformulae of the form $\mathcal{G}_{[a_{i,j}, b_{i,j}]}\psi$ and ψ must be satisfied initially, i.e., at $t = a_{i,j}$ and $t = 0$, respectively. This is formalized in the next assumption:

Assumption 1: For every $\phi_{i,j} = \psi$, it holds that $p_{i,j}(\mathbf{x}_o(0), 0) \geq 0$, and for every $\phi_{i,j} = \mathcal{G}_{[a_{i,j}, b_{i,j}]}\psi$, it holds that $p_{i,j}(\mathbf{x}_o(a_{i,j}), 0) \geq 0$, $j \in \{1, \ldots, M_i\}$, $i \in \{1, \ldots, R\}$.

***Conjunctions**:* Consider now the formula $\phi_i = \phi_{i,1} \wedge \phi_{i,2} \wedge \cdots \wedge \phi_{i,M_i}$ where each $\phi_{i,j}$ is of the form (11), $i \in \{1, \ldots, R\}$. Let $\bar{h}_{i,j}(\mathbf{x}_o, t)$ be the predicate function corresponding to each $\phi_{i,j}$, constructed as in (14). We accommodate the aforementioned conjunction by defining the vector $\bar{h}_i(\mathbf{x}_o, t) = \begin{bmatrix} \bar{h}_{i,1} & \cdots & \bar{h}_{i,M_i} \end{bmatrix}^\top$; all $\bar{h}_{i,j}, j \in \{1, \ldots, M_i\}$, must satisfy (13) to guarantee satisfaction of ϕ_i. The following example illustrates the construction of $\bar{h}_i$.

Example 1: Consider the STL formula $\phi_i = \mathcal{G}_{[5,10]}(\|\mathbf{x}_o - A\|_p < 0.05) \wedge \mathcal{F}_{[5,15]}(\|\mathbf{x}_o - B\|_p < 0.05) \wedge \mathcal{G}_{[10,20]}\Big(\phi_o - \pi < \frac{\pi}{6}\Big) \wedge (\|\mathbf{x}_o - C\|_p < 1)\mathcal{U}_{[20,25]}(\|\mathbf{x}_o - C\|_p < 0.05)$ where A, B and C are some desired positions for $\mathbf{x}_o$. The formula ϕ_i can be broken down into four subtasks, namely, 1) $\mathcal{G}_{[5,10]}(\|\mathbf{x}_o - A\|_p < 0.05)$ which requires the object x_o between 5s and 10s to match configuration A within a margin of 0.05; 2) $\mathcal{F}_{[5,15]}(\|\mathbf{x}_o - B\|_p < 0.05)$ which requires the object x_o between 5s and 15s to eventually match configuration B within a margin of 0.05; 3) $\mathcal{G}_{[10,20]}\Big(\phi_o - \pi < \frac{\pi}{6}\Big)$ requires the object to assume a roll angle of π between 10s and 20s within a margin of $\pi/6$; and finally 4) $(\|\mathbf{x}_o - C\|_p < 1)\mathcal{U}_{[20,25]}(\|\mathbf{x}_o - C\|_p < 0.05)$ requires the object to assume configuration C with an accuracy of 1 until the object gets close enough to C within the margin 0.05 in the interval $[20, 25]$s. The function $\bar{h}_i(\mathbf{x}_o, t) = [\bar{h}_{i,1}(\mathbf{x}_o, t), \bar{h}_{i,2}(\mathbf{x}_o, t), \bar{h}_{i,3}(\mathbf{x}_o, t), \bar{h}_{i,4}(\mathbf{x}_o, t), \bar{h}_{i,5}(\mathbf{x}_o, t)]^\top$, corresponding to ϕ_i, is of the form,

$$\bar{h}_i(\mathbf{x}_o, t) = \begin{bmatrix} \beta_{i,1}(t)\ h_{i,1}(\mathbf{x}_o) \\ \beta_{i,2}(t)\ h_{i,2}(\mathbf{x}_o) \\ \beta_{i,3}(t)\ h_{i,3}(\mathbf{x}_o) \\ \beta_{i,4}(t)\ h_{i,4}(\mathbf{x}_o) \\ \beta_{i,5}(t)\ h_{i,5}(\mathbf{x}_o) \end{bmatrix} = \begin{bmatrix} \beta_{i,1}(t)\ \|\mathbf{x}_o - A\|_p \\ \beta_{i,2}(t)\ \|\mathbf{x}_o - B\|_p \\ \beta_{i,3}(t)\ (\phi_o - \pi) \\ \beta_{i,4}(t)\ \|\mathbf{x}_o - C\|_p \\ \beta_{i,5}(t)\ \|\mathbf{x}_o - C\|_p \end{bmatrix},$$

where $M_i = 5$ and $\beta_{i,j}(t)$ are switching functions of the form (12), $j \in \{1, \dots, 5\}$.

***Disjunctions**:* Finally, we consider disjunctions of the form $\phi = \phi_1 \vee \phi_2 \vee \dots \vee \phi_R$ where each ϕ_i is a conjunction of subformulas of the form (11), $i \in \{1, \dots, R\}$. As discussed before, each ϕ_i corresponds to a predicate function $h_i(\mathbf{x}_o, t) \in \mathbb{R}^{M_i}$; Since satisfying ϕ_i for any i, satisfies ϕ, we use a cost metric to decide on initially choosing such a ϕ_i. More specifically, we define the cost function $J_i(h_i(\mathbf{x}_o, t_0))$, which captures the robust satisfaction of the STL subtask ϕ_i, with t_0 being the first time instant appearing in the subformulas $\phi_{i,j},\ i \in \{1, \dots, R\},\ j \in \{1, \dots, M_i\}$. One such example of J_i is $J_i(h(\mathbf{x}_o, t_0)) = \frac{1}{M_i} \sum_{j \in \{1,\dots,M_i\}} \left| \frac{h_{i,j}(\mathbf{x}_o, t_0)}{\gamma_{i,j}(t_0)} \right|$, which chooses a ϕ_i that is, on average, furthest from the funnel boundaries at t_0. We select then the subtask ϕ_ℓ that satisfies $\ell = \arg\min_i J_i(h_i(\mathbf{x}_o(t_0), t_0))$. We do this automatically by defining the variable

$$k_\ell(t) = \begin{cases} 1 & \text{if} \quad \ell = \arg\min_i J_i(h(\mathbf{x}_o(t_0), t_0)) \\ 0 & \text{otherwise.} \end{cases}$$

and finally selecting the predicate function for ϕ as $\widetilde{h} = [\widetilde{h}_1^\top, \dots, \widetilde{h}_R^\top]^\top$, with

$$\widetilde{h}(\mathbf{x}_o, t) = \bar{K}(t) \left[\bar{h}_1(\mathbf{x}_o, t) \quad \cdots \quad \bar{h}_R(\mathbf{x}_o, t) \right]^\top$$

where $\bar{K}(t) = \text{blckdiag}\{[k_i I_{M_i}]_{i \in \{1,\dots,R\}}\}$. An example of this strategy is shown below.

Example 2: Consider the STL formula, $\phi = \Big[\mathcal{G}_{[5,10]}(\|\mathbf{x}_o - A\|_p < 1) \wedge \mathcal{F}_{[10,15]}(\|\mathbf{x}_o - B\|_p < 0.05)\Big] \vee \Big[\mathcal{F}_{[5,15]}(\|\mathbf{x}_o - C\|_p < 0.05)\Big]$ where A, B and C are desired attainable positions for $\mathbf{x}_o$. The predicate function corresponding to ϕ is

$$\widetilde{h}(\mathbf{x}_o, t) = \left[\begin{array}{cc:c} k_1(t) & 0 & 0 \\ 0 & k_1(t) & 0 \\ \hdashline 0 & 0 & k_2(t) \end{array}\right] \left[\begin{array}{c} \beta_1(t)\ \|\mathbf{x}_o - A\|_p \\ \beta_2(t)\ \|\mathbf{x}_o - B\|_p \\ \hdashline \beta_3(t)\ \|\mathbf{x}_o - C\|_p \end{array}\right].$$

Remark 1: In the aforementioned procedure on the derivation of k_ℓ for the disjunction sub-formulae, the minimization involves the time instant t_0 (see $J_i(h(\mathbf{x}_o(t_0), t_0))$. For instance, the algorithm might choose the index $\ell = \arg\min_i J_i(h(\mathbf{x}_o(t_0), t_0))$ at $t = t_0$, choosing to satisfy ϕ_ℓ, but there might exist t' and ℓ' such that $\ell' = \arg\min_i J_i(h(\mathbf{x}_o(t_0), t)) \neq \ell$, for all $t \geq t'$. We can extend the proposed algorithm to handle such cases by considering a time-varying index $\ell(t) = \arg\min_i J_i(h(\mathbf{x}_o(t_0), t))$. To retain the differentiability of $\widetilde{h}$, we can replace the $\min()$ operator with a differentiable approximation (see, e.g., [16].). We note that this stage of planning occurs offline, based on the STL formula that is available to the agents.

B. Control Design

In this section, we present the decentralized control design that guarantees compliance with the funnel of the previous section and consequently, satisfaction of the STL task. Before proceeding, we impose the following required assumptions.

Assumption 2: The object's pose does not result in a singular $J_{o_r}(x_o(t))$, i.e. $\theta_o(t) \in \Big(-\frac{\pi}{2}, \frac{\pi}{2}\Big)$, for all $t \geq 0$.

Assumption 3: The function $\widetilde{h}(\mathbf{x}_o, t)$ satisfies the following properties:

- $\widetilde{h}(\mathbf{x}_o, t)$ is cont. differentiable in $\mathbb{M} \times \mathbb{R}_{\geq 0}$ and $\widetilde{h}$, $\frac{\partial \widetilde{h}}{\partial \mathbf{x}_o}$, and $\frac{\partial \widetilde{h}}{\partial t}$ are uniformly bounded in t for all $\mathbf{x}_o \in \mathbb{M}$.
- For every $c_1 > 0$, there exists a $c_2 > 0$ such that $\{\mathbf{x}_o \in \mathbb{M} : \|\widetilde{h}(\mathbf{x}_o, t)\| \leq c_1\} \subset \{\mathbf{x}_o \in \mathbb{M} : \|\mathbf{x}_o\| \leq c_2\}$.

Assumption 2 is required [17] for the controllability of the object's pose (see (6a)). Assumption 3 provides simple differentiability and boundedness conditions for $\widetilde{h}$ and that boundedness of $\widetilde{h}(\mathbf{x}_o, t)$ implies the boundedness of $\mathbf{x}_o$.

Next, let ℓ be the index of the chosen subformulae to be satisfied in the disjunction operator, i.e., $k_\ell = 1$ and $k_i = 0$, for all $i \in \{1, \dots, R\} \backslash \{\ell\}$. Then, the task to be satisfied is given by the function $\widetilde{h}_\ell(\mathbf{x_o}, t) = \bar{h}_\ell(\mathbf{x_o}, t) = [\bar{h}_{\ell,1}, \dots, \bar{h}_{\ell,M_\ell}]^\top$. We provide now the control design.

Step I-a. Choose the performance functions $\gamma_h(t) = \text{diag}\{\gamma_{h_{\ell,1}}, \gamma_{h_{\ell,2}}, \dots, \gamma_{h_{\ell,M_\ell}}\}$ and $\gamma_{h_{\ell,j}}(t) = (\gamma^0_{h_{\ell,j}} - \gamma^\infty_{h_{\ell,j}}) \exp(-l_{\ell,j} t) + \gamma^\infty_{h_{\ell,j}}$, with $\gamma^0_{h_{\ell,j}} > \bar{h}_{\ell,j}(\mathbf{x}_o(0), 0)$; $l_{\ell,j} \in \mathbb{R}_{>0}$ and $\gamma^\infty_{h_{\ell,j}}$ are chosen such that, when $-\gamma_{h_{\ell,j}}(t) < \bar{h}_{\ell,j}(\mathbf{x}_o, t) < \gamma_{h_{\ell,j}}(t)$ holds, the task is satisfied, for $j \in \{1, 2, \dots, M_\ell\}$. In the special case that $\phi_{\ell,j}$ has the form $\|\mathbf{x}_o - A\| < z$, for some $\mathbf{x}_o \in \mathbb{M}$, $z > 0$, we set $\gamma^\infty_{h_{\ell,j}} = z$.

Step I-b. Define the normalised errors $\xi_h \in \mathbb{R}^L$ by

$$\xi_h = [\xi_{h_{\ell,1}}, \dots, \xi_{h_{\ell,M_\ell}}]^\top = \gamma_h^{-1}(t) \bar{h}_\ell(\mathbf{x}_o, t) \qquad (15)$$

and design the reference velocity v_r as

$$v_r(\xi_h, t) = -g_s J_{o_r}(\mathbf{x}_o) \frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top \gamma_h^{-1}(t) r_h(\xi_h) \varepsilon_h(\xi_h) \qquad (16)$$

where g_s is a positive constant and the signals $\varepsilon_h : (-1,1)^{M_\ell} \to \mathbb{R}^{M_\ell}$ and $r_h : (-1,1)^{M_\ell} \to \mathbb{R}^{M_\ell \times M_\ell}$ are $\varepsilon_h(\xi_h) = [\varepsilon_{h_{\ell,1}}(\xi_{h_{\ell,1}}), \dots, \varepsilon_{h_{\ell,M_\ell}}(\xi_{h_{\ell,M_\ell}})]^\top$, $r_h(\xi_h) = \text{diag}\{[r_{h_{\ell,j}}(\xi_{h_{\ell,j}})]_{j \in \{1,\dots,M_\ell\}}\}$, with

$$\varepsilon_{h_{\ell,j}} = \ln\Big(\frac{1 + \xi_{h_{\ell,j}}}{1 - \xi_{h_{\ell,j}}}\Big),\ r_{h_{\ell,j}} = \frac{2}{1 - \xi^2_{h_{\ell,j}}},\ j \in \{1, \dots, M_\ell\}.$$

Step II follows the lines presented in [3].

Step II-a. Define the velocity error $e_v \in \mathbb{R}^6$ as,

$$e_v = [e_{v_1}, \dots, e_{v_6}]^\top = v_o - v_r(\xi_h, t), \qquad (17)$$

and velocity performance functions $\gamma_v(t) = \text{diag}\{\gamma_{v_1}, \gamma_{v_2}, \dots, \gamma_{v_6}\}$ where $\gamma_{v_n}(t) = (\gamma^0_{v_n} -$

$\gamma_{v_i}^{\infty})\exp(-l_{v_n}t) + \gamma_{v_n}^{\infty}$, with parameters $\gamma_{v_n}^0 > |e_{v_n}(t_0)|$, $\gamma_{v_i}^{\infty} \in (0, \gamma_{v_n}^0)$ and $l_{v_n} > 0$, $\forall n \in \{1, \ldots, 6\}$.

Step II-b. Define the normalised velocity errors $\xi_v \in \mathbb{R}^6$:

$$\xi_v = [\xi_{v_1}, \ldots, \xi_{v_6}]^\top = \gamma_v^{-1}(t) e_v, \tag{18}$$

where $\gamma_v = \mathrm{diag}\{[\gamma_{v_m}]_{m\in\{1,\ldots,6\}}\}$, and design the decentralized control law as

$$u(\xi_h, \xi_v, t) = \begin{bmatrix} u_1(\xi_h, \xi_v, t) \\ \vdots \\ u_N(\xi_h, \xi_v, t) \end{bmatrix} = -C_g G^\star(q) \gamma_v^{-1}(t) r_v(\xi_v) \varepsilon_v(\xi_v) \tag{19}$$

where $G^\star(q) = [J_{o_1}^{-1}(q_1), \ldots, J_{o_N}^{-1}(q_N)] \in \mathbb{R}^{6N\times 6}$, $C_g = g_v \mathrm{diag}\{[c_i I_6]_{i\in\mathcal{N}}\} \in \mathbb{R}^{6N\times 6N}$ and the signals $\varepsilon_v : (-1,1)^6 \to \mathbb{R}^6$ and $r_v : (-1,1)^6 \to \mathbb{R}^{6\times 6}$ are $\varepsilon_v(\xi_v) = [\varepsilon_{v_1}(\xi_{v_1}), \ldots, \varepsilon_{v_6}(\xi_{v_6})]^\top$, $r_v(\xi_v) = \mathrm{diag}\{[r_{v_n}(\xi_{v_n})]_{n\in\{1,\ldots,6\}}\}$, with

$$\varepsilon_{v_n} = \ln\Big(\frac{1+\xi_{v_n}}{1-\xi_{v_n}}\Big),\ r_{v_n} = \frac{2}{1-\xi_{v_n}^2},\ n \in \{1, \ldots, 6\}.$$

Note that the STL planning, i.e., the design of performance functions in Step I-a and Step II-a is performed offline, whereas the control input (19) is applied online by the agents.

Remark 2: The control protocol guarantees the funnel containment by enforcing the normalized errors $\xi_{h_{\ell,j}}$, ξ_{v_i} to remain strictly within $(-1,1)$, which is equivalent to guaranteeing the boundedness of the transformed signals $\varepsilon_{h_{\ell,j}}$, ε_{v_i}, respectively. These signals, which are used in the control design, operate similarly to reciprocal barrier functions [18]. In particular, they admit higher negative or positive values as $\bar{h}_{\ell,j}(\mathbf{x}_o, t)$ and e_{v_i} approach the funnel boundaries, enforcing in that way the funnel containment while implicitly counteracting the unknown dynamic terms.

The correctness of the proposed control scheme is proven in the following theorem.

Theorem 1: Consider N agents rigidly grasping an object with coupled dynamics (10) subject to an STL formula ϕ of the form (3). Assume that $\lambda_{\min}\Big(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top\Big) \geq \kappa > 0$, for all $t \geq 0$. Then the decentralized control (19) guarantees $|\bar{h}_{\ell,j}(\mathbf{x}_o(t), t)| < \gamma_{\ell,j}(t)$, for all $t \geq 0$, and the boundedness of all closed loop signals.

Proof: Consider first the dynamics of the normalised errors $\dot{\xi}_h = \gamma_h^{-1}(t)(\dot{\bar{h}}_\ell(\mathbf{x}_o, t) - \dot{\gamma}_h(t)\xi_h)$, $\dot{\xi}_v = \gamma_v^{-1}(t)(\dot{e}_v - \dot{\gamma}_v(t)\xi_v)$, which, after using (10), (17), (16) and (19), become

$$\begin{aligned} \dot{\xi}_h &= f_h(\xi_h, t) \\ &= -g_s \frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} \gamma_h^{-1}(t)\gamma_h^{-1}(t) r_h(\xi_h)\varepsilon_h(\xi_h) + \\ &\gamma_h^{-1}(t)\Big(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} J_{o_r}^{-1}(\mathbf{x}_o)\gamma_v(t)\xi_v + \frac{\partial \bar{h}_\ell}{\partial t} - \dot{\gamma}_h(t)\xi_h\Big), \end{aligned} \tag{20a}$$

$$\begin{aligned} \dot{\xi}_v &= f_v(\xi_h, \xi_v, t) \\ &= \gamma_v^{-1}(t)\Big(\tilde{M}^{-1}(q)\Big[-g_v\gamma_v^{-1}(t) r_v(\xi_v)\varepsilon_v(\xi_v) - \\ &\tilde{C}(q,\dot{q})\mathbf{v}_o - \tilde{h}(q,\dot{q}) - \tilde{w}(q,t)\Big] - \dot{v}_r(\xi_h, t) - \dot{\gamma}_v(t)\xi_v\Big), \end{aligned} \tag{20b}$$

which, by using (15)-(19), can be written compactly as

$$\dot{\xi} = f(\xi, t) = \begin{bmatrix} f_h(\xi, t) \\ f_v(\xi, t) \end{bmatrix} \tag{21}$$

where $\xi = [\xi_h^\top, \xi_v^\top] \in \mathbb{R}^{M_\ell+6}$. Define next the open and nonempty set $\Omega_\xi = \Omega_{\xi_h} \times \Omega_{\xi_v} \subset \mathbb{R}^{M_\ell+6}$ with $\Omega_{\xi_h} = (-1,1)^{M_\ell}$ and $\Omega_{\xi_v} = (-1,1)^6$. We now proceed in two steps: We first prove that there exists a maximal solution $\xi : [t_0, \tau_{\max}) \to \Omega_\xi$ to (21) and then show that $\tau_{\max} = \infty$ which then completes the proof. By choosing $\gamma_h(t)$ and $\gamma_v(t)$ as discussed in Steps I-b, II-a, respectively, and owing to Assumptions 1 and 2, we ensure that $\xi(t_0) \in \Omega_\xi$. In view of (10), (15)-(19), and Assumptions 2, 3, one can conclude that $f(\xi, t)$ is continuous in t and locally Lipschitz in ξ over Ω_ξ. Therefore, [19, Theorem 54] dictates the existence of a maximal solution $\xi : [t_0, \tau_{\max}) \to \Omega_\xi$. Thus,

$$\xi_{h_{\ell,j}}(t) = \frac{\bar{h}_{\ell,j}(\mathbf{x}_o, t)}{\gamma_{h_{\ell,j}}(t)} \in (-1,1), \tag{22a}$$

$$\xi_{v_n}(t) = \frac{e_{v_n}(t)}{\gamma_{v_n}(t)} \in (-1,1), \tag{22b}$$

$\forall j \in \{1, \ldots, M_\ell\}$, $n \in \{1, \ldots, 6\}$, $t \in [t_0, \tau_{\max})$ from which we conclude $\bar{h}_{\ell,j}(\mathbf{x}_o, t)$ and $e_{v_n}(t)$ are bounded by $\gamma_{h_{\ell,j}}(t)$ and $\gamma_{v_n}(t)$ respectively, for $j \in \{1, \ldots, M_\ell\}$, $n \in \{1, \ldots, 6\}$, $t \in [t_0, \tau_{\max})$. Consider now the positive definite and radially unbounded function $V_h(\varepsilon_h) = \frac{1}{2}\varepsilon_h^\top \varepsilon_h$. Its time derivative reads $\dot{V}_h = \varepsilon_h^\top(\xi_h) r_h(\xi_h)\dot{\xi}_h$, and, after using (20a),

$$\begin{aligned} \dot{V}_h &\leq -g_s \Big\|\varepsilon_h^\top(\xi_h) r_h(\xi_h)\gamma_h^{-1}(t)\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}\Big\|^2 + \\ &\varepsilon_h^\top(\xi_h) r_h(\xi_h)\gamma_h^{-1}(t)\Big(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} J_{o_r}^{-1}(\mathbf{x}_o)\gamma_v(t)\xi_v + \frac{\partial \bar{h}_\ell}{\partial t} - \dot{\gamma}_h(t)\xi_h\Big), \end{aligned}$$

Note next that $\gamma_v(t)$ and $\dot{\gamma}(t)$ are bounded for all $t \geq t_0$, and $\|\xi_h\| \leq \sqrt{M_\ell}$, $\|\xi_v\| \leq \sqrt{6}$, for all $t \in [t_0, \tau_{\max})$ from (22). In view of Assumption 3, the boundedness of $\bar{h}_{\ell,j}(\mathbf{x}_o, t)$ from (22) implies the boundedness of $\mathbf{x}_o$ and the boundedness of $\frac{\partial h_\ell(\mathbf{x}_o,t)}{\partial \mathbf{x}_o}$, $\frac{\partial h_\ell(\mathbf{x}_o,t)}{\partial t}$. Moreover, Assumption 2 and the continuity of $J_{o_r}^{-1}$ implies its boundedness. Note that (22) imply that the aforementioned bounds are independent of $\tau_{\max}$. Hence, $\dot{V}_h$ reduces to

$$\begin{aligned} \dot{V}_h \leq &- g_s\lambda_{\min}\Big(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top\Big)\|\varepsilon_h^\top(\xi_h) r_h(\xi_h)\gamma_h^{-1}(t)\|^2 + \\ &\|\varepsilon_h^\top(\xi_h) r_h(\xi_h)\gamma_h^{-1}(t)\|\bar{B}_h, \end{aligned}$$

where $\bar{B}_h$ is a constant independent of $\tau_{\max}$, satisfying $\bar{B}_h \geq \Big\|\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} J_{o_r}^{-1}(\mathbf{x}_o)\gamma_v(t)\xi_v + \frac{\partial \bar{h}_\ell}{\partial t} - \dot{\gamma}_h(t)\xi_h\Big\|$. Therefore, one concludes that $\dot{V}_h < 0 \Leftrightarrow \|\varepsilon_h^\top(\xi_h) r_h(\xi_h)\gamma_h^{-1}(t)\| > \frac{\bar{B}_h}{g_s\kappa}$, which, by noting $r_{h_m} > 2$, is equivalent to,

$$\dot{V}_h < 0 \Leftrightarrow \|\varepsilon_h(\xi_h)\| > \frac{\bar{B}_h \max\{\gamma_{h_{\ell,j}}^0\}_{j\in(1,\ldots,M_\ell)}}{g_s\kappa}.$$

Therefore, it holds that $\|\varepsilon_h(\xi_h)\| \leq \bar{\varepsilon}_h$, where

$$\bar{\varepsilon}_h = \max\Big\{\|\varepsilon_h(\xi_h(0))\|, \frac{\bar{B}_h \max\{\gamma_{h_{\ell,j}}^0\}_{j\in(1,\ldots,M_\ell)}}{g_s\kappa}\Big\}$$

$\forall t \in [t_0, \tau_{\max})$, and by invoking the inverse logarithmic function,

$$-1 < \frac{\exp(-\bar{\varepsilon}_h) - 1}{\exp(-\bar{\varepsilon}_h) - 1} = \underline{\xi}_h \leq \xi_{h_{\ell,j}}(t) \leq \bar{\xi}_h = \frac{\exp(\bar{\varepsilon}_h) - 1}{\exp(\bar{\varepsilon}_h) - 1} < 1. \quad (23)$$

Hence, $v_r(\xi_h, t)$ in (16) and consequently $\mathbf{v}_o = \gamma_v(t)\xi_v + v_r(\xi_h, t)$ remain bounded for all $t \in [0, \tau_{\max})$. Proceeding in a similar manner with the function $V_v = \frac{1}{2}\varepsilon_v^\top \varepsilon_v$, we conclude that $\|\varepsilon_v(\xi_v)\| \leq \bar{\varepsilon}_v$, where

$$\bar{\varepsilon}_v = \max\left\{\|\varepsilon_v(\xi_v(0))\|, \frac{\bar{B}_v \max\{\gamma^0_{h_n}\}_{n\in(1,\ldots,6)}}{g_v \lambda_{\min}(\tilde{M})}\right\}$$

$\forall t \in [t_0, \tau_{\max})$, where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue; by invoking the inverse logarithmic function, we obtain

$$-1 < \frac{\exp(-\bar{\varepsilon}_v) - 1}{\exp(-\bar{\varepsilon}_v) - 1} = \underline{\xi}_v \leq \xi_{h_n}(t) \leq \bar{\xi}_v = \frac{\exp(\bar{\varepsilon}_v) - 1}{\exp(\bar{\varepsilon}_v) - 1} < 1, \quad (24)$$

for all $n \in \{1, \ldots, 6\}$, $t \in [t_0, \tau_{\max})$, and hence the boundedness of the proposed control law (19). More details regarding this step are provided in [3, Sec. 4.1]. What remains to be shown is that $\tau_{\max} = \infty$. Towards this end, note from (23), (24) that $\xi(t) \in \Omega'_\xi$, where Ω'_ξ is a compact subset of Ω_ξ. Therefore, by invoking [19, Prop. C.3.6], we conclude the forward completeness of the solution and $\tau = \infty$. ■

Remark 3: The satisfaction of $\lambda_{\min}\left(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} \frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top\right) \geq \kappa > 0$ is related to the number and structure of the STL sub-tasks encoded in $\bar{h}_\ell$. More specifically, it requires the matrix $\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} \frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top \in \mathbb{R}^{M_\ell \times M_\ell}$ to have full rank M_ℓ, which imposes the requirement on the rank of $\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} \in \mathbb{R}^{M_\ell \times 6}$ to be M_ℓ. The latter leads to the following necessary conditions. Firstly, the number M_ℓ of sub-tasks must not be larger than six, i.e., $M_\ell \leq 6$. Intuitively, the number of tasks the robotic system can execute simultaneously is restricted by the number of degrees of freedom in the task space, which is six. The overall number of sub-tasks can be increased by defining $\bar{h}_\ell$ to encode more tasks that do not overlap in time; we omit the formal description, however, for ease of exposition. Secondly, $\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}$ must have linearly independent rows, which translates to the requirement of linearly independent sub-tasks encoded in $\bar{h}_\ell$. One can identify tasks that do not satisfy this assumption and remove potential conflicting dependencies a priori. It should be stressed, however, that condition $\lambda_{\min}\left(\frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o} \frac{\partial \bar{h}_\ell}{\partial \mathbf{x}_o}^\top\right) \geq \kappa > 0$ is only a sufficient condition for the theoretical analysis of the proposed control algorithm; practically, the latter may guarantee the satisfaction of the STL tasks even if the condition is not satisfied.

V. Experiments

We consider $N = 2$ six degree-of-freedom HEBI-Robotics arms[1] rigidly grasping a wooden box of dimensions $29cm \times 30.5cm \times 17cm$, and weight $2.05kg$, as depicted in Figure 1. The robots get the object's position ($\mathbf{x}_o$) and velocity (v_o) from the coupled object-agent kinematics as mentioned in Section III.3. The arms lack an on-board computer and thus communicate the states $(\mathbf{x}_o, v_o)$ to a central computer that calculates the reference velocity and the control command. The central computer is an eight-core 1.6 GHz CPU with 16 GB of RAM, and the communication frequency is 150Hz. The arms share the load equally i.e. $c_1 = c_2 = 0.5$. The experiments were performed by cancelling an estimate of gravity forces on the arms and object.

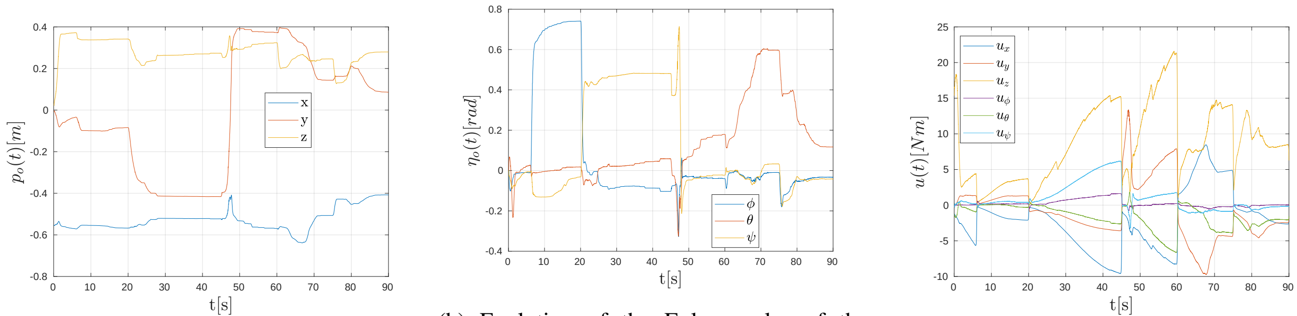
The object's task is encoded in the formula $\phi := \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5$ where $\phi_1 = (\|\mathbf{x}_o - A\| < 2.2)\mathcal{U}_{[0,20]}(\|\mathbf{x}_o - B\| < 4)$, $\phi_2 = \mathcal{G}_{[20,45]}(\|\mathbf{x}_o - C\| < 2.5)$, $\phi_3 = \mathcal{F}_{[45,60]}(\|\mathbf{x}_o - D\| < 2.2)$, $\phi_4 = \mathcal{F}_{[60,75]}(\|\mathbf{x}_o - E\| < 3)$ and $\phi_5 = \mathcal{F}_{[75,90]}(\|\mathbf{x}_o - F\| < 3)$, with $A = [-0.7, 0, 0.4, 0, 0, 0]^\top$, $B = [-0.7, 0, 0.4, \pi/4, 0, 0]^\top$, $C = [-0.7, -0.5, 0.35, 0, 0, \pi/4]^\top$, $D = [-0.7, 0.5, 0.4, 0, 0, 0]^\top$, $E = [-0.35, 0, 0.35, 0, \pi/8, 0]^\top$ and $F = [-0.5, 0, 0.35, 0, 0, 0]^\top$, corresponding to desired setpoints for the object's pose $\mathbf{x}_o = [p_o^\top, \eta_o^\top]^\top$. More specifically, ϕ requires the object to assume a sequence of poses in the respective time intervals, as dictated by ϕ_i, $i \in \{1, \ldots, 5\}$. The STL tasks are encoded as follows,

$$\bar{h}(\mathbf{x}_o, t) = [\bar{h}_1(\mathbf{x}_o, t)), \ldots, \bar{h}_6(\mathbf{x}_o, t))]^\top =$$
$$\begin{bmatrix} \left(\frac{1}{1+\exp(-20(t+0.1))} - \frac{1}{1+\exp(-20(t-6.1))}\right)\|\mathbf{x}_o - A\| \\ \left(\frac{1}{1+\exp(-20(t-6.1))} - \frac{1}{1+\exp(-20(t-20.1))}\right)\|\mathbf{x}_o - B\| \\ \left(\frac{1}{1+\exp(-20(t-20.1))} - \frac{1}{1+\exp(-20(t-45.1))}\right)\|\mathbf{x}_o - C\| \\ \left(\frac{1}{1+\exp(-20(t-45.1))} - \frac{1}{1+\exp(-20(t-60.1))}\right)\|\mathbf{x}_o - D\| \\ \left(\frac{1}{1+\exp(-20(t-60.1))} - \frac{1}{1+\exp(-20(t-75.1))}\right)\|\mathbf{x}_o - E\| \\ \left(\frac{1}{1+\exp(-20(t-75.1))} - \frac{1}{1+\exp(-20(t-90.1))}\right)\|\mathbf{x}_o - F\| \end{bmatrix}.$$

The object is initially fixed at $[-0.7, 0, 0, 0, 0, 0]^\top$. The parameters of the performance functions are chosen as $\gamma^0_h = \begin{bmatrix}5.5 & 69.32 & 466.5 & 8.8\times10^5 & 2.3\times10^8 & 2.3\times10^{10}\end{bmatrix}$, $\gamma^\infty_h = \begin{bmatrix}2.2 & 4 & 2.5 & 2.2 & 3 & 3\end{bmatrix}$, $l = \begin{bmatrix}0.35 & 0.35 & 0.2 & 0.25 & 0.3 & 0.3\end{bmatrix}$ and the control gains as $g_s = \text{diag}\{10, 10, 20, 10, 10, 10\}$ and $g_v = \text{diag}\{5, 5, 10, 5, 5, 5\}$.

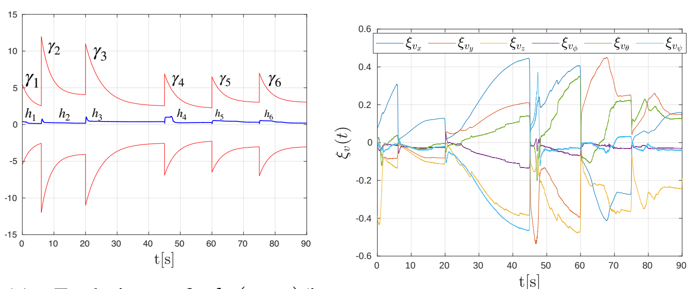
The experimental results are shown in Figure 2, 3a and Figure 3b. Figure 2a, 2b show the position and orientation, respectively, of the object executing the STL formula ϕ. Figure 2c depicts the control input of one of the robots, which is identical to the second robot's. Figure 3a illustrates the evolution of the predicate functions $h_i(\mathbf{x}_o, t)$, along with the respective performance functions that correspond to the 6 sub-formulas in ϕ. One can easily conclude that ϕ is satisfied. Note that, for better visualization, the activation functions and the nonactive $h_i(\mathbf{x}_o, t)$ are not shown in Figure 3a. Figure 3b depicts the evolution of $\xi_v(t)$ which respects $\xi_v \in (-1, 1)$ and is thus inside the prescribed velocity funnels. Compared to existing results [16], the experiments here can handle a richer STL fragment and define tasks to exploit all the 6 degrees-of-freedom of the system. Additionally, the dynamics are assumed to be unknown and we avoid any approximations when considering *disjunctions*.

[1]The video of the experiments can be found here: https://youtu.be/CfoWh5PEYPo



(a) Evolution of the object's position $p_o(t)$ (b) Evolution of the Euler angles of the object $\eta_o(t)$ (c) Control input to the robots $u(t)$

Fig. 2: Results pertaining to an experimental run of the setup in Fig. 2 satisfying ϕ



(a) Evolution of $h_i(\mathbf{x}_o, t)$(in blue) and performance functions $\gamma_i(t)$(in red) for $i = 1, \ldots, 6$. (b) Evolution of normalised velocity errors $\xi_v(t)$

Fig. 3: Experimental evolution of the funnels and velocity error

VI. Conclusion

We provide an algorithm for cooperative manipulation subject to STL tasks. In particular, we translate the STL tasks to PPC constraints, and we provide a decentralized control protocol for the robotic arms manipulating the object. We validate our results by performing experiments on two manipulator arms grasping an object. The flexibility in the design of performance functions provides the user with greater control over accuracy and robustness of the system. Future work consists of considering task infeasibilities and rolling contacts, as well as experiments using mobile bases.

References

[1] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 94–103, 2002.

[2] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[3] C. K. Verginis and D. V. Dimarogonas, "Timed abstractions for distributed cooperative manipulation," *Autonomous Robots*, vol. 42, no. 4, pp. 781–799, Apr. 2018.

[4] O. Maler and D. Nickovic, "Monitoring Temporal Properties of Continuous Signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, ser. Lecture Notes in Computer Science, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer, 2004, pp. 152–166.

[5] A. Linard and J. Tumova, "Active learning of signal temporal logic specifications," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 779–785.

[6] 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 *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 81–87.

[7] C. P. Bechlioulis and G. A. Rovithakis, "A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems," *Automatica*, vol. 50, no. 4, pp. 1217–1226, Apr. 2014.

[8] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 757–762, 2019.

[9] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.

[10] J. T. Wen and K. Kreutz-Delgado, "Motion and force control of multiple robotic manipulators," *Automatica*, vol. 28, no. 4, pp. 729–743, 1992.

[11] H.-C. Lin, J. Smith, K. K. Babarahmati, N. Dehio, and M. Mistry, "A projected inverse dynamics approach for multi-arm cartesian impedance control," *International Conference on Robotics and Automation (ICRA)*, pp. 1–5, 2018.

[12] S. Erhart, D. Sieber, and S. Hirche, "An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 315–322, 2013.

[13] A.-N. Ponce-Hinestroza, J.-A. Castro-Castro, H.-I. Guerrero-Reyes, V. Parra-Vega, and E. Olguỳn-Dỳaz, "Cooperative redundant omnidirectional mobile manipulators: Model-free decentralized integral sliding modes and passive velocity fields," *International Conference on Robotics and Automation (ICRA)*, pp. 2375–2380, 2016.

[14] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 346–352.

[15] S. Saha and A. A. Julius, "Task and motion planning for manipulator arms with metric temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 379–386, 2017.

[16] L. Lindemann, C. K. Verginis, and D. V. Dimarogonas, "Prescribed performance control for signal temporal logic specifications," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 2997–3002.

[17] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008, vol. 200.

[18] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[19] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer-Verlag New York, 1998, oCLC: 1165444109.