Conformal Data-driven Control of Stochastic Multi-Agent Systems under Collaborative Signal Temporal Logic Specifications

Eleftherios E. Vlahakis[1], Lars Lindemann[2] and Dimos V. Dimarogonas[1]

***Abstract*— We address control synthesis of stochastic discrete-time linear multi-agent systems under jointly chance-constrained collaborative signal temporal logic specifications in a distribution-free manner using available disturbance samples, which are partitioned into training and calibration sets. Leveraging linearity, we decompose each agent's system into deterministic nominal and stochastic error parts, and design disturbance feedback controllers to bound the stochastic errors by solving a tractable optimization problem over the training data. We then quantify prediction regions (PRs) for the aggregate error trajectories corresponding to agent *cliques*, involved in collaborative tasks, using conformal prediction and calibration data. This enables us to address the specified joint chance constraint via Lipschitz tightening and the computed PRs, and relax the centralized stochastic optimal control problem to a deterministic one, whose solution represents feedforward inputs. To enhance scalability, we decompose the deterministic problem into agent-level subproblems solved in an MPC fashion, yielding a distributed control policy. Finally, we present an illustrative example and a comparison with [1].**

I. Introduction

Multi-agent systems (MAS) arise in various applications, including robotics and cyber-physical systems, where multiple agents collaborate to accomplish tasks jointly. We use signal temporal logic (STL) to define such specifications [2], leveraging Boolean and temporal operators for expressing precisely spatio-temporal constraints [3], [4]. Under stochastic uncertainty, STL specifications are typically formulated as chance constraints, making STL control synthesis challenging, as chance-constrained problems are generally nonconvex and intractable [5]. Existing methods rely on constraint tightening [5]–[7] or analytic techniques [8], [9] to provide probabilistic guarantees. These approaches may be limited to Gaussian settings [8], computationally expensive [6], or rely on concentration inequalities and union bound [1], making them unsuitable for general MAS. In this work, we propose a tractable data-driven approach for STL synthesis of stochastic MAS under individual and collaborative STL tasks with probabilistic guarantees.

Data-driven methods offer flexibility in relaxing chance constraints by leveraging available samples and providing guarantees through statistical tools such as conformal prediction (CP) [10]. Using a calibration dataset of *i.i.d.* samples, CP enables the construction of prediction regions for a test sample with a specified probability, without requiring knowledge of the underlying distribution. CP has recently been applied in control [11], including STL-based runtime verification for single-agent schemes [12], [13]. Single-agent STL control synthesis in the presence of uncontrollable agents has been explored in [14], providing probabilistic guarantees via CP, while a multi-agent reinforcement learning problem is studied in [15] using CP, albeit without provable assurances.

Here, we study control synthesis for stochastic MAS under collaborative STL specifications, formulated as a chance-constrained optimization. We assume: 1) collaborative tasks are assigned to arbitrary groups of agents, called *cliques*; 2) a joint chance constraint is imposed across all cliques with a desired probability level $1-\theta$; and 3) disturbance trajectory samples are available for each agent (Sec. II).

Assuming linear agent dynamics, we decompose each system into a deterministic nominal part driven by a feedforward term and a stochastic error part regulated by a disturbance-feedback (DF) controller. To design DF, we introduce *non-conformity scores*—functions of aggregate error trajectories parameterized by DF gains—that capture deviations from nominal behavior within each clique and return the maximum deviation across cliques. The DF gains are then obtained by minimizing a CVaR-based cost of the empirical distribution of these scores, computed from training samples (Sec. III-C).

Next, we quantify prediction regions (PRs) for clique-level aggregate error trajectories by computing the $(1-\theta)$th quantile of the empirical distribution of the proposed scores evaluated using the remaining (calibration) samples. This provides joint probabilistic guarantees across cliques via CP (Sec. III-D) and enables us to handle the joint chance constraint through Lipschitz tightening and the computed PRs. Consequently, the original centralized chance-constrained problem can be relaxed to a deterministic one, whose solution yields the agents' feedforward input sequences. To improve scalability, we decompose the deterministic problem into agent-level subproblems, solved in an MPC fashion with a shrinking horizon, leading to a distributed controller with feedback and feedforward elements. The method is illustrated and compared to our previous work [1] (Secs. IV and V). All proofs are provided in [16].

II. Problem setup

A. Notation and Preliminaries

The sets of real numbers and nonnegative integers are $\mathbb{R}$ and $\mathbb{N}$, respectively. Let $N, M \in \mathbb{N}$. Then, $\mathbb{N}_{[0,N]} = \{0, 1, \dots, N\}$. Let $x_1, \dots, x_n$ be vectors. Then, $x =$

This work was supported by the Swedish Research Council (VR), the Knut & Alice Wallenberg Foundation (KAW), the Horizon Europe Grant SymAware and the ERC Consolidator Grant LEAFHOUND. [1]Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044, Stockholm, Sweden. Email: {vlahakis,dimos}@kth.se. [2]Automatic Control Laboratory, ETH Zürich, Zürich, 8092, Switzerland. Email: llindemann@ethz.ch.

$(x_1, \ldots, x_n) = [x_1^\intercal \ \cdots \ x_n^\intercal]^\intercal$. We denote by $\boldsymbol{x}(a:b) = (x(a), \ldots, x(b))$ an aggregate vector consisting of $x(t)$, $t \in \mathbb{N}_{[a,b]}$, representing a trajectory. When $x(t)$, $t \in \mathbb{N}_{[a,b]}$, are random vectors, $\boldsymbol{x}(a:b) = (x(a), \ldots, x(b))$ is a random process. Let $x_i(t)$, for $t \in \mathbb{N}_{[0,N]}$ and $i \in \mathbb{N}_{[1,M]}$. Then, $\boldsymbol{x}(0:N) = (x(0), \ldots, x(N))$ denotes an aggregate trajectory when $x(t) = (x_1(t), \ldots, x_M(t))$, $t \in \mathbb{N}_{[0,N]}$. A random variable w following a distribution $\mathscr{D}_w$ is denoted as $w \sim \mathscr{D}_w$ and the expected value of w is $\mathbb{E}(w)$. The probability of event Y is $\Pr\{Y\}$. $Q_\delta(\mathscr{D})$ is the δ-th quantile of a distribution $\mathscr{D}$, i.e., for $Z \sim \mathscr{D}$, $Q_\delta(\mathscr{D}) = \inf\{z : \Pr\{Z \leq z\} \geq \delta\}$. The ceiling operator is $\lceil \cdot \rceil$. The cardinality of a set $\mathcal{V}$ is $|\mathcal{V}|$. The symbol $\otimes$ denotes the Kronecker product.

1) Conformal Prediction: Let $\mathcal{R}^{(0)}, \ldots, \mathcal{R}^{(k)}$ be *i.i.d.* random variables. We refer to $\mathcal{R}^{(\varsigma)}$, $\varsigma \in \mathbb{N}_{[0,k]}$, as *nonconformity scores*. Given $\theta \in (0,1)$, an upper bound $q \in \mathbb{R}$ for $\mathcal{R}^{(0)}$, which we call a test point, can be obtained as follows.

Lemma 1 *[17, Lemma 1] If $\mathcal{R}^{(0)}, \ldots, \mathcal{R}^{(k)}$ are i.i.d. random variables, then for any $\theta \in (0,1)$, we have*

$$\Pr\left\{\mathcal{R}^{(0)} \leq Q_{1-\theta}\Big(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(k)}, \infty\Big)\right\} \geq 1-\theta. \quad (1)$$

Remark 1 *(i) Assuming $\mathcal{R}^{(1)} \leq \cdots \leq \mathcal{R}^{(k)}$, one can compute $Q_{1-\theta}\big(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(k)}, \infty\big)$ as the pth smallest nonconformity score $\mathcal{R}^{(p)}$, where $p = \lceil (k+1)(1-\theta) \rceil$. $\mathcal{R}^{(p)}$ is finite or $p \in \mathbb{N}_{[1,k]}$ if $k \geq \lceil (k+1)(1-\theta) \rceil$. (ii) The probability* (1) *is defined over the randomness in the test and calibration draws. Conditional coverage guarantees of the form $\Pr\{\mathcal{R}^{(0)} \leq C \mid \mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(k)}\}$ are in general difficult to obtain. Notably, probably approximately correct coverage guarantees $\Pr_c\{\Pr\{\mathcal{R}^{(0)} \leq Q_{1-\hat{\theta}}\big(\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(k)}, \infty\big)\} \geq 1-\theta\} \geq 1-\beta$ can be obtained by setting $\hat{\theta} = \theta - \sqrt{\frac{\ln(1/\beta)}{2k}}$, with the "outer" probability $\Pr_c$ taken over the randomness in the calibration draw $\mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(k)}$, and $\beta \in (0,1)$ [18].*

2) Conditional value at risk (CVaR): For a random variable $\mathcal{R} \sim \mathscr{D}$ and confidence level $(1-\theta)$, $\mathrm{CVaR}_{1-\theta}(\mathcal{R})$ is the expected value of $\mathcal{R}$ in the θ-tail exceeding $Q_{1-\theta}(\mathscr{D})$, i.e., $\mathbb{E}\left(\mathcal{R} \,|\, Q_{1-\theta}(\mathscr{D}) \leq \mathcal{R}\right)$. CVaR is a coherent risk measure that meets essential criteria such as convexity and monotonicity. It is typically optimized using standard convex and linear programming techniques, and can be formulated as:

$$\mathrm{CVaR}_{1-\theta}(\mathcal{R}) = \min_{\eta \in \mathbb{R}} \mathbb{E}\left(\eta + \frac{1}{\theta}(\mathcal{R} - \eta)_+\right), \quad (2)$$

where $(\cdot)_+ = \max\{0, \cdot\}$ [19].

3) Signal temporal logic: We consider STL formulas with standard syntax

$$\varphi := \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 U_{[t_1,t_2]} \phi_2, \quad (3)$$

where $\pi := (\mu(x) \geq 0)$ is a predicate, $\mu(x) : \mathbb{R}^{n_x} \to \mathbb{R}$ is a predicate function of $x \in \mathbb{R}^{n_x}$, and ϕ, ϕ_1, and ϕ_2 are STL formulas, which are built recursively using predicates π, logical operators $\neg$ and $\wedge$, and the *until* temporal operator U, with $[t_1, t_2] \equiv \mathbb{N}_{[t_1,t_2]}$. We omit $\vee$ (*or*), $\Diamond$ (*eventually*) and $\Box$ (*always*) operators from (3) and the sequel, as these may be defined by (3), e.g., $\phi_1 \vee \phi_2 = \neg(\neg\phi_1 \wedge \neg\phi_2)$, $\Diamond_{[t_1,t_2]}\phi = \top U_{[t_1,t_2]}\phi$, and $\Box_{[t_1,t_2]}\phi = \neg\Diamond_{[t_1,t_2]}\neg\phi$.

We denote by $\boldsymbol{x}(t) \models \phi$, $t \in \mathbb{N}$, the satisfaction of ϕ, verified over $\boldsymbol{x}(t) = (x(t), x(t+1), \ldots)$. The validity of a formula ϕ can be verified using Boolean or quantitative semantics: $\boldsymbol{x}(t) \models \phi \iff \rho^\phi(\boldsymbol{x}(t)) > 0$, where the *robustness function* $\rho^\phi : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}$ quantifies how robustly a signal $\boldsymbol{x}(t)$ satisfies a formula ϕ. Based on the Boolean semantics of STL, the horizon of a formula ϕ is roughly the minimum length of a signal that is required to verify ϕ. We refer to [3] and [4] for detailed descriptions of the Boolean and quantitative semantics of STL.

To facilitate the definition of joint STL formulas and to represent groups of agents involved in collaborative tasks, we adopt the concept of *cliques* from graph theory. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, potentially with self-loops and multiple edges, with node set $\mathcal{V}$ of cardinality $M = |\mathcal{V}|$ and edge set $\mathcal{E}$. Let $\nu \subseteq \mathcal{V}$ with $|\nu| \geq 1$, and define $\mathcal{E}_\nu \subseteq \mathcal{E}$ as the set of edges connecting the nodes in ν. Then, $(\nu, \mathcal{E}_\nu)$ is called a *clique* if it forms a complete subgraph of $\mathcal{G}$ [20].

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with clique set $\mathcal{K}$. For simplicity, and with a slight abuse of notation, we denote a clique $(\nu, \mathcal{E}_\nu) \in \mathcal{K}$ simply by ν. Let $\nu \in \mathcal{K}$, with $\nu = (i_1, \ldots, i_{|\nu|})$, and consider vectors $x_{i_j}(t)$, $j \in \mathbb{N}_{[1,|\nu|]}$, with $t \in \mathbb{N}$, Then, $x_\nu(t) = (x_{i_1}(t), \ldots, x_{i_{|\nu|}}(t))$ is an aggregate vector, and $\boldsymbol{x}_\nu(t) = (x_\nu(t), x_\nu(t+1), \ldots)$ is an aggregate trajectory. We denote the satisfaction of a collaborative task ϕ_ν defined over $\boldsymbol{x}_\nu(t)$ by $\boldsymbol{x}_\nu(t) \models \phi_\nu$, or equivalently by $\rho^{\phi_\nu}(\boldsymbol{x}_\nu(t)) > 0$.

B. Multi-agent system

1) Dynamics: We consider a MAS with M agents, with the ith agent following the dynamics

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + w_i(t), \quad (4)$$

where $x_i(t) \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$, and $w_i(t) \in \mathbb{R}^{n_i}$ are the state, input, and disturbance vectors, respectively, $t \in \mathbb{N}$, the initial condition, $x_i(0)$, is known, $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, with the index $i \in \mathcal{V} = \{1, \ldots, M\}$. Recall a clique $\nu = (i_1, \ldots, i_{|\nu|})$, where $i_j \in \mathcal{V}$, with $j \in \mathbb{N}_{[1,|\nu|]}$. By collecting individual state, input, and disturbance vectors, as $x_\nu(t) = (x_{i_1}(t), \ldots, x_{i_{|\nu|}}(t)) \in \mathbb{R}^{n_\nu}$, $u_\nu(t) = (u_{i_1}(t), \ldots, u_{i_{|\nu|}}(t)) \in \mathbb{R}^{m_\nu}$, and $w_\nu(t) = (w_{i_1}(t), \ldots, w_{i_{|\nu|}}(t)) \in \mathbb{R}^{n_\nu}$, respectively, we write the aggregate dynamics of $|\nu|$ agents as

$$x_\nu(t+1) = A_\nu x_\nu(t) + B_\nu u_\nu(t) + w_\nu(t), \quad (5)$$

with $A_\nu = \mathrm{diag}(A_{i_1}, \ldots, A_{i_{|\nu|}})$, $B_\nu = \mathrm{diag}(B_{i_1}, \ldots, B_{i_{|\nu|}})$. When $\nu = (1, \ldots, M)$, the aggregate dynamics of the entire MAS are written as

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad (6)$$

where $A = \mathrm{diag}(A_1, \ldots, A_M)$ and $B = \mathrm{diag}(B_1, \ldots, B_M)$.

2) STL specification: The MAS is subject to

$$\phi = \bigwedge_{\nu \in \mathcal{K}_\phi} \phi_\nu, \tag{7}$$

which is a conjunctive STL formula, where each conjunct ϕ_ν defined over $\boldsymbol{x}_\nu(t)$ follows the syntax in (3) and represents a formula that involves the clique ν, indicating all agents in ν can interact with each other. The set $\mathcal{K}_\phi$ collects all these cliques induced by ϕ, and may include individual agents ($|\nu| = 1$) or group of agents ($1 < |\nu| \leq |\mathcal{V}|$). Let $\pi := (\mu(y) \geq 0)$ be a predicate in ϕ, where $\mu(y) : \mathbb{R}^{n_y} \to \mathbb{R}$. The vector $y \in \mathbb{R}^{n_y}$ can represent either an individual state vector $x_i \in \mathbb{R}^{n_i}$, where $i \in \mathcal{V}$, or an aggregate state vector $x_\nu \in \mathbb{R}^{n_\nu}$ that collects the states of agents in clique $\nu \in \mathcal{K}_\phi$.

3) Disturbance: Let $\boldsymbol{w}_i(0{:}N{-}1) = \{w_i(0), w_i(1), \ldots, w_i(N{-}1)\}$ denote the disturbance sequence for the ith agent. We assume that the joint distribution $\mathscr{D}_{w_i}$ of the random vectors $w_i(t) \in \mathbb{R}^{n_i}$, $t \in \mathbb{N}_{[0,N-1]}$, is unknown, and instead that a disturbance dataset, $\mathcal{D}^{w_i} = \{\boldsymbol{w}_i^{(0)}, \ldots, \boldsymbol{w}_i^{(k)}\}$, of $k{+}1$ samples is available, with $\boldsymbol{w}_i^{(\varsigma)} = (w_i^{(\varsigma)}(0), \ldots, w_i^{(\varsigma)}(N{-}1))$, $\varsigma \in \mathbb{N}_{[0,k]}$. We assume that the disturbance sequence samples in $\mathcal{D}^{w_i}$ are i.i.d., where each sample represents one realization of the process. Note that although $\boldsymbol{w}_i^{(\varsigma)}$, $\varsigma \in \mathbb{N}_{[0,k]}$, are assumed to be independent, the random vectors $w_i(t)$, $t \in \mathbb{N}_{[0,N-1]}$, may be correlated across time for the ith agent. We partition $\mathcal{D}^{w_i}$ into training and calibration sets: $\mathcal{D}^{w_i}_{\text{train}} = \{\boldsymbol{w}_i^{(k_1+1)}, \ldots, \boldsymbol{w}_i^{(k)}\}$ and $\mathcal{D}^{w_i}_{\text{cal}} = \{\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(k_1)}\}$. We also use the grouped datasets $\mathcal{D}^{w_\nu}_{\text{train}}$ and $\mathcal{D}^{w_\nu}_{\text{cal}}$ over cliques.

C. Problem statement

We wish to solve the stochastic optimal control problem

$$\begin{aligned}
&\underset{\substack{\boldsymbol{u}(0:N-1)\\ \boldsymbol{x}(0:N)}}{\text{Min.}} \ \mathbb{E}\left(\sum_{i=1}^{M}\Big(\sum_{t=0}^{N-1}(\ell_i(x_i(t), u_i(t))) + V_{f,i}(x_i(N))\Big)\right)\\
&\text{s.t. } x(t+1) = Ax(t) + Bu(t) + w(t),\ t \in \mathbb{N}_{[0,N)},\\
&\qquad \Pr\{\boldsymbol{x}(0:N) \models \phi\} \geq 1-\theta, \text{ with } x(0) = x_0,
\end{aligned} \tag{8}$$

where $\ell_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$, $V_{f,i} : \mathbb{R}^{n_i} \to \mathbb{R}$, the optimization variables are $\boldsymbol{u}(0{:}N{-}1) = (u(0), \ldots, u(N-1))$, $\boldsymbol{x}(0{:}N) = (x(0), \ldots, x(N))$, with $u(t) = (u_1(t), \ldots, u_M(t))$, $t \in \mathbb{N}_{[0,N-1]}$, and $x(t) = (x_1(t), \ldots, x_M(t))$, $t \in \mathbb{N}_{[0,N]}$, respectively, ϕ is the STL formula in (7), to be satisfied by $\boldsymbol{x}(0{:}N)$ with a probability $1-\theta$, with $\theta \in (0,1)$, x_0 is a known initial condition, and N is the horizon of ϕ. Solving the problem directly is challenging due to 1) the (joint) probabilistic constraint in (8), equivalently written as

$$\Pr\{\boldsymbol{x}_\nu(0:N) \models \phi_\nu,\ \forall \nu \in \mathcal{K}_\phi\} \geq 1-\theta, \tag{9}$$

2) the lack of knowledge about the distribution of the disturbance $w(t)$, and 3) the growing computational complexity with the number of agents. We address challenges 1) and 2) next via a data-driven approach, and 3) thereafter.

III. Data-driven uncertainty quantification and feedback design

A. Decomposition of dynamics

Due to the linearity in (4), the state of the ith agent can be decomposed into a deterministic part, $z_i(t)$, and an error term, $e_i(t)$, i.e., $x_i(t) = z_i(t) + e_i(t)$, with initial conditions $z_i(0) = x_i(0)$ and $e_i(0) = 0$. Consider a clique $\nu \in \mathcal{K}_\phi$, where $\nu = (i_1, \ldots, i_{|\nu|})$, and define the aggregate vectors $z_\nu(t) = (z_{i_1}(t), \ldots, z_{i_{|\nu|}}(t))$ and $e_\nu(t) = (e_{i_1}(t), \ldots, e_{i_{|\nu|}}(t))$. We consider a control policy comprising disturbance-feedback [21] and feedforward elements, given by $u_\nu(t) = \sum_{k=0}^{t-1}\Gamma_\nu^{t,k} w_\nu(k) + v_\nu(t)$, where $\Gamma_\nu^{t,k} = \operatorname{diag}(\Gamma_{i_1}^{t,k}, \ldots, \Gamma_{i_{|\nu|}}^{t,k})$, with $\Gamma_{i_j}^{t,k} \in \mathbb{R}^{m_{i_j} \times n_{i_j}}$ for $j \in \mathbb{N}_{[1,|\nu|]}$. Then, the decomposed dynamics of the agents in ν are

$$z_\nu(t+1) = A_\nu z_\nu(t) + B_\nu v_\nu(t), \tag{10a}$$

$$e_\nu(t+1) = A_\nu e_\nu(t) + \sum_{k=0}^{t-1}\Gamma_\nu^{t,k} w_\nu(k) + w_\nu(t). \tag{10b}$$

As the two systems in (10) are decoupled, and in view of Proposition 1 below, we first synthesize the gains $\Gamma_\nu^{t,k}$, $k \in \mathbb{N}_{[0,t-1]}$, for $t \in \mathbb{N}_{[1,N)}$, to obtain tight prediction regions (see Definition 1) for the trajectories of the error systems in (10b), which will be used to handle uncertainty in (8)–(9), and then, design the feedforward elements in (10a) (see Sec. IV). Next, we construct trajectory samples of the error system in (10b) and use them for feedback synthesis and data-driven provable uncertainty quantification.

Definition 1 *Let $\boldsymbol{y}(a{:}b) \in \mathbb{R}^{n(b-a+1)}$ be a random process, with $y(t) \in \mathbb{R}^n$, $t \in \mathbb{N}_{[a,b]}$. We call the ball $\mathbb{B}(q) \subseteq \mathbb{R}^{n(b-a+1)}$ a prediction region (PR) of $\boldsymbol{y}(a{:}b)$ at probability level $1-\theta$, if* $\Pr\{\boldsymbol{y}(a{:}b) \in \mathbb{B}(q)\} \geq 1-\theta$.

Proposition 1 ([1]) *Let $\boldsymbol{x}(0{:}N) = \boldsymbol{z}(0{:}N) + \boldsymbol{e}(0{:}N)$, with $\boldsymbol{x}(0{:}N) = (x(0), \ldots, x(N))$, $\boldsymbol{z}(0{:}N) = (z(0), \ldots, z(N))$ and $\boldsymbol{e}(0{:}N) = (e(0), \ldots, e(N))$. Let $q > 0$ be s.t.* $\Pr\{\boldsymbol{e}(0{:}N) \in \mathbb{B}(q)\} \geq 1-\theta$. *If $\boldsymbol{z}(0{:}N) + \boldsymbol{e}(0{:}N) \models \phi$ for all $\boldsymbol{e}(0{:}N) \in \mathbb{B}(q)$, then* $\Pr\{\boldsymbol{x}(0) \models \phi\} \geq 1-\theta$.

B. Error trajectory samples

Let matrices $\boldsymbol{A}_i$, $\boldsymbol{\Gamma}_i$, and $\boldsymbol{B}_i$, $i \in \mathcal{V}$, be defined as

$$\begin{bmatrix} I_{n_i} & 0 & \cdots & 0\\ A_i & I_{n_i} & \ddots & 0\\ \vdots & \vdots & \ddots & \vdots\\ A_i^{N-1} & A_i^{N-2} & \cdots & I_{n_i}\end{bmatrix}, \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0\\ \Gamma_i^{1,0} & 0 & \cdots & \cdots & 0\\ \Gamma_i^{2,0} & \Gamma_i^{2,1} & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots\\ \Gamma_i^{N-1,0} & \Gamma_i^{N-1,1} & \cdots & \Gamma_i^{N-1,N-2} & 0\end{bmatrix}, \tag{11}$$

and $\boldsymbol{B}_i = \boldsymbol{A}_i(I_N \otimes B_i)$, respectively. Then, for the νth clique, with $\nu = (i_1, \ldots, i_{|\nu|})$, a dataset of aggregate error trajectory samples $\boldsymbol{e}_\nu^{(\varsigma)}(1{:}N) = (e_\nu^{(\varsigma)}(1), \ldots, e_\nu^{(\varsigma)}(N))$ can be constructed as

$$\mathcal{D}^{e_\nu} = \{\boldsymbol{e}_\nu^{(0)}(1{:}N), \ldots, \boldsymbol{e}_\nu^{(k)}(1{:}N)\}, \tag{12a}$$

$$\boldsymbol{e}_\nu^{(\varsigma)}(1{:}N) = (\boldsymbol{A}_\nu + \boldsymbol{B}_\nu\boldsymbol{\Gamma}_\nu)\boldsymbol{w}_\nu^{(\varsigma)}(0{:}N{-}1),\ \varsigma \in \mathbb{N}_{[0,k]}, \tag{12b}$$

with $\boldsymbol{A}_\nu = \mathrm{diag}(\boldsymbol{A}_{i_1}, \ldots, \boldsymbol{A}_{i_{|\nu|}})$, $\boldsymbol{B}_\nu = \mathrm{diag}(\boldsymbol{B}_{i_1}, \ldots, \boldsymbol{B}_{i_{|\nu|}})$, and $\boldsymbol{\Gamma}_\nu = \mathrm{diag}(\boldsymbol{\Gamma}_{i_1}, \ldots, \boldsymbol{\Gamma}_{i_{|\nu|}})$. In the following, we partition $\mathcal{D}^{e_\nu}$ into $\mathcal{D}^{e_\nu}_{\mathrm{train}}$ and $\mathcal{D}^{e_\nu}_{\mathrm{cal}}$. The samples in $\mathcal{D}^{e_\nu}_{\mathrm{train}}$ are generated from the disturbance samples in $\mathcal{D}^{w_\nu}_{\mathrm{train}}$ and are linearly parameterized by the feedback gains $\boldsymbol{\Gamma}_\nu$, which facilitates both feedback synthesis and PR scaling over cliques, as detailed in Sec. III-C. The samples in $\mathcal{D}^{e_\nu}_{\mathrm{cal}}$ are constructed using $\mathcal{D}^{w_\nu}_{\mathrm{cal}}$ and the trained feedback gains, and are used in Sec. III-D to obtain PRs with probabilistic guarantees.

C. Data-driven PR scaling and disturbance feedback design

Here, we design the feedback gains $\boldsymbol{\Gamma}_\nu$, $\nu \in \mathcal{K}_\phi$, using the training datasets $\mathcal{D}^{e_\nu}_{\mathrm{train}}$. In view of the joint chance constraint in (9), we also introduce scaling parameters C_ν, $\nu \in \mathcal{K}_\phi$, to weigh uncertainty across cliques, adjusting the size of PRs for the aggregate trajectories of the systems in (10b). To this end, we define nonconformity scores $E^{(\varsigma)}(C, \boldsymbol{\Gamma})$, $\varsigma \in \mathbb{N}_{[k_1+1,k]}$, parameterized by $\boldsymbol{\Gamma} = \{\boldsymbol{\Gamma}_\nu\}_{\nu \in \mathcal{K}_\phi}$ and $C = \{C_\nu\}_{\nu \in \mathcal{K}_\phi}$,

$$E^{(\varsigma)}(C, \boldsymbol{\Gamma})) = \max_{\nu \in \mathcal{K}_\phi} \left(C_\nu \| \boldsymbol{e}^{(\varsigma)}_\nu(1{:}N) \| \right), \tag{13}$$

with $\boldsymbol{e}^{(\varsigma)}_\nu(1{:}N)$ as defined in (12b). Motivated by [22] and our previous work [23, Sec. IV] for single-agent settings, the feedback gains in $\boldsymbol{\Gamma}$ and weights in C can be synthesized by minimizing $Q_{\hat{\theta}}\big(E^{(k_1+1)}(C, \boldsymbol{\Gamma}), \ldots, E^{(k)}(C, \boldsymbol{\Gamma})\big)$, requiring $0 \le C_\nu \le 1$, $\nu \in \mathcal{K}_\phi$, and $\sum_{\nu \in \mathcal{K}_\phi} C_\nu = 1$, with $\hat{\theta} = (1 + \frac{1}{k - k_1 - 1})(1 - \theta)$ (see [11, Sec. 2.1] for details). To handle the complexity of this optimization, arising from both the quantile's encoding and the multi-agent setup, we propose a tractable CVaR-based over approximation, becoming tighter as the number of samples increases [19], as follows:

$$P := \underset{\eta \ge 0, Y^{(\varsigma)}, C, \boldsymbol{\Gamma}}{\text{Minimize}} \ \eta + \frac{1}{q} \sum_{\varsigma = k_1 + 1}^{k} (Y^{(\varsigma)} - \eta)_+ \ \text{s. t.} \tag{14a}$$

$$Y^{(\varsigma)} \ge C_\nu \| \boldsymbol{e}^{(\varsigma)}_\nu(1{:}N) \|, \ \forall \nu \in \mathcal{K}_\phi, \ \varsigma \in \mathbb{N}_{(k_1,k]}, \tag{14b}$$

$$0 \le C_\nu \le 1, \ \nu \in \mathcal{K}_\phi, \ \sum_{\nu \in \mathcal{K}_\phi} C_\nu = 1, \tag{14c}$$

where $q = (k - k_1 - 1)(1 - \hat{\theta})$, and the variables $Y^{(\varsigma)}$ are introduced to address the max operator in (13) by the constraints in (14b). Next, we denote by $P(C)$ the optimization in (14) for fixed weights in C and by $P(\boldsymbol{\Gamma})$ the optimization in (14) for fixed feedback gains in $\boldsymbol{\Gamma}$. Note that both $P(C)$ and $P(\boldsymbol{\Gamma})$ are convex. Algorithm 1 summarizes an efficient procedure, known as a block-coordinate descent algorithm, for solving (14) iteratively, which is a widely used heuristic performing well in practice for a small number of iterations.

Algorithm 1 Iterative procedure for solving (14)

```
1: Set C^0={C_ν=0 for |ν|>1, C_i=1/M for i∈V}
2: for τ in 1 : τ_max do
3:     Solve P(C^{τ−1}) and return Γ^τ
4:     Solve P(Γ^τ) and return C^τ.
5: return (C* ← C^{τ_max}, Γ* ← Γ^{τ_max})
```

D. Error prediction regions

Given the feedback gains in $\boldsymbol{\Gamma}^* = \{\boldsymbol{\Gamma}^*_\nu\}_{\nu \in \mathcal{K}_\phi}$ and weights in $C^* = \{C^*_\nu\}_{\nu \in \mathcal{K}_\phi}$, obtained via Algorithm 1, we now derive PRs for the aggregate error trajectories in cliques $\nu \in \mathcal{K}_\phi$ with the desired confidence level as follows.

Proposition 2 *Let $\boldsymbol{\Gamma}^*$ and C^* contain the feedback gains and weights, respectively, obtained via Algorithm 1. Construct dataset $\mathcal{D}^{e_\nu}_{\mathrm{cal}}$ as in (12), using $\mathcal{D}^{w_\nu}_{\mathrm{cal}}$ and $\boldsymbol{\Gamma}^*$, and compute $E^{(\varsigma)}(C^*, \boldsymbol{\Gamma}^*) = \max_{\nu \in \mathcal{K}_\phi} \left(C^*_\nu \| \boldsymbol{e}^{(\varsigma)}_\nu(1{:}N) \| \right)$, for $\varsigma \in \mathbb{N}_{[0,k_1]}$, and $q = Q_{1-\theta}\big(E^{(1)}(C^*, \boldsymbol{\Gamma}^*), \ldots, E^{(k_1)}(C^*, \boldsymbol{\Gamma}^*), \infty\big)$. Then,*

$$\Pr\left\{ \boldsymbol{e}^{(0)}_\nu(1{:}N) \in \mathbb{B}(q/C^*_\nu), \ \forall \nu \in \mathcal{K}_\phi \right\} \ge 1 - \theta. \tag{15}$$

IV. STL Control Synthesis

Motivated by Prop. 1, we will relax (8) into a deterministic problem by tightening STL constraints via our data-driven uncertainty quantification in Proposition 2 and solve it through a scalable decomposition.

A. STL Tightening

We first tighten the robustness function of the formula in (7) using the Lipschitz constants of its subformulas and the PRs from Prop. 2. The following two lemmas lead to the relaxed formulation in Theorem 1.

Lemma 2 *[24, Prop. 1] For any STL formula ϕ with Lipschitz continuous predicate functions, the robustness function $\rho^\phi(\boldsymbol{z}(0{:}N) + \boldsymbol{e}(0{:}N))$ is Lipschitz continuous, with Lipschitz constant L_ϕ obtained as the maximum Lipschitz constant of the predicate functions appearing in ϕ.*

Lemma 3 *Let trajectory $\boldsymbol{x}(0{:}N) = \boldsymbol{z}(0{:}N) + \boldsymbol{e}(0{:}N)$, with $\Pr\{\boldsymbol{e}(0{:}N) \in \mathbb{B}(q/C)\} \ge 1 - \theta$, and consider STL formula ϕ. Let L_ϕ be the Lipschitz constant of the robustness function $\rho^\phi(\boldsymbol{z}(0{:}N) + \boldsymbol{e}(0{:}N))$. If $\rho^\phi(\boldsymbol{z}(0{:}N)) \ge L_\phi \frac{q}{C}$, then $\Pr\left\{\rho^\phi(\boldsymbol{x}(0{:}N)) \ge 0\right\} \ge 1 - \theta$.*

B. Relaxation of the chance-constrained synthesis problem

We now present a data-driven relaxation of (8).

Theorem 1 *Let L_{ϕ_ν}, $\nu \in \mathcal{K}_\phi$, be the Lipschitz constants of the robustness functions of formulas ϕ_ν, $\nu \in \mathcal{K}_\phi$. Let also test aggregate error trajectories $\boldsymbol{e}^{(0)}_\nu(0{:}N)$, $\nu \in \mathcal{K}_\phi$, with $e^{(0)}_\nu(0) = 0$, satisfy (15), and aggregate deterministic trajectories $\boldsymbol{z}_\nu(0{:}N)$, $\nu \in \mathcal{K}_\phi$, with $z_\nu(0) = x_\nu(0)$, driven by feedforward controllers $\boldsymbol{v}_\nu(0{:}N-1)$, $\nu \in \mathcal{K}_\phi$, be solution to:*

$$\underset{\substack{\boldsymbol{v}(0:N-1) \\ \boldsymbol{z}(0:N)}}{\text{Minimize}} \ \sum_{i=1}^{M} \left(\sum_{t=0}^{N-1} (\ell_i(z_i(t), v_i(t))) + V_{f,i}(z_i(N)) \right)$$

$$\text{subject to } z(t+1) = Az(t) + Bv(t), \ t \in \mathbb{N}_{[0,N)},$$

$$\rho^{\phi_\nu}(\boldsymbol{z}_\nu(0:N)) \ge L_{\phi_\nu} \frac{q}{C^*_\nu}, \ \nu \in \mathcal{K}_\phi. \tag{16}$$

Then, $\Pr\left\{ \boldsymbol{x}^{(0)}_\nu(0:N) \models \phi_\nu, \ \forall \nu \in \mathcal{K}_\phi \right\} \ge 1 - \theta$, with $\boldsymbol{x}^{(0)}_\nu(0:N) = \boldsymbol{z}_\nu(0:N) + \boldsymbol{e}^{(0)}_\nu(0:N)$, $\nu \in \mathcal{K}_\phi$.

C. Distributed control synthesis

To handle STL constraints in (16), one may use existing methods based on integer programming or log-sum-exp approximations [25]. We defer details to an extended version. Next, we propose an agent-level decomposition of (16) and an iterative procedure to manage multi-agent complexity.

1) Agent-level decomposition: For agent i participating in at least one clique, i.e., $i \in \nu$, with $\nu \in \mathcal{K}_\phi$, let $\mathcal{T}_i = \{\nu \in \mathcal{K}_\phi \mid i \in \nu\}$ be the set of cliques containing i. Then, a multi-agent STL formula equivalent to the original one in (7) is defined as $\hat{\phi} = \bigwedge_{i\in\mathcal{V}} \hat{\phi}_i$, where $\hat{\phi}_i = \bigwedge_{\nu_i\in\mathcal{T}_i} \phi_{\nu_i}$. Next, we denote $\varrho^{\phi_{\nu_i}}(\boldsymbol{z}_{\nu_i}(0{:}N)){=}\rho^{\phi_{\nu_i}}(\boldsymbol{z}_{\nu_i}(0{:}N)){-}L_{\phi_{\nu_i}}\frac{q}{C_{\nu_i}}$, $\nu_i \in \mathcal{T}_i$, and introduce the following problems for the ith agent

$$P_i^0 := \underset{\boldsymbol{v}_i^0, \boldsymbol{z}_i^0}{\text{Minimize}} \ \mathcal{L}_i(\boldsymbol{z}_i^0, \boldsymbol{v}_i^0) \text{ subject to} \tag{17a}$$
$$z_i^0(k+1){=}A_i z_i^0(k){+}B_i v_i^0(k),\ k \in \mathbb{N}_{[0,N)}, \tag{17b}$$
$$\varrho^{\phi_i}(\boldsymbol{z}_i^0) \geq 0,\ \text{with } z_i^0(0) = x_i(0), \tag{17c}$$

$$P_i^t := \underset{\boldsymbol{v}_i^t, \boldsymbol{z}_i^t}{\text{Minimize}} \ \mathcal{L}_i(\boldsymbol{z}_i^t, \boldsymbol{v}_i^t) - \Omega_i \mu_{\nu_t}^t \text{ subject to} \tag{18a}$$
$$z_i^t(k+1) = A_i z_i^t(k) + B_i v_i^t(k),\ k \in \mathbb{N}_{[t,N)}, \tag{18b}$$
$$\varrho^{\phi_i}(\boldsymbol{z}_i^t) \geq 0,\ \text{with } z_i^t(t) = x_i(t), \tag{18c}$$
$$\varrho^{\phi_{\nu_t}}(\boldsymbol{z}_{\nu_t}^t){\geq}\mu_{\nu_t}^t,\ \nu_t{=}\underset{\nu\in\mathcal{T}_i,\,|\nu|>1}{\text{argmin}}\{\varrho^{\phi_\nu}(\boldsymbol{z}_\nu^{t-1})\}, \tag{18d}$$
$$\mu_{\nu_t}^t \geq \min\left(0, \varrho^{\phi_{\nu_t}}(\boldsymbol{z}_{\nu_t}^{t-1})\right), \tag{18e}$$
$$\varrho^{\phi_\nu}(\boldsymbol{z}_\nu^t){\geq}\min\left(0, \varrho^{\phi_\nu}(\boldsymbol{z}_\nu^{t-1})\right), \forall \nu{\in}\mathcal{T}_i\backslash\{\nu_t, i\} \tag{18f}$$

where $\Omega_i \gg 0$, $z_i^t(\tau)$ denotes the prediction of $x_i(\tau)$ carried out at time t, $\varrho^{\phi_\nu}(\boldsymbol{z}_\nu^t)$ is the robustness function of the formula ϕ_ν, $\nu \in \mathcal{T}_i$, evaluated over the trajectory $\boldsymbol{z}_\nu^t$, and

$$\boldsymbol{z}_\nu^t{=}(x_\nu(0), ..., x_\nu(t{-}1), z_\nu^t(t), ..., z_\nu^t(N)), \tag{19}$$
$$\boldsymbol{v}_i^t{=}(v_i(0), ..., v_i(t{-}1), v_i^t(t), ..., v_i^t(N{-}1)), \tag{20}$$
$$\mathcal{L}_i(\boldsymbol{z}_i^t, \boldsymbol{v}_i^t){=}\sum_{k=0}^{N-1} \ell_i(z_i^t(k), v_i^t(k)){+}V_{f,i}(z_i^t(N)). \tag{21}$$

P_i^0 and P_i^t decompose the centralized problem in (16): P_i^0 is solved by all agents at $t = 0$, while for $t \geq 1$, P_i^t is solved by a subset of agents. Details are in [1], where these subsets are scheduled offline in a centralized way, which we now extend to distributed coordination.

2) Distributed MPC-like implementation: Let

$$\boldsymbol{z}_i^t(x_i(t), \boldsymbol{v}_i^t){=}(x_i(0), ..., x_i(t), z_i^t(t+1), ..., z_i^t(N)), \tag{22}$$

denote a trajectory where the last $N{-}t$ states are generated by the last $N{-}t$ inputs of $\boldsymbol{v}_i^t$ starting from $x_i(t)$. For $t > 1$, the ith agent computes $\varrho^{\phi_{\nu_i}}(\boldsymbol{z}_{\nu_i}^{t-1})$, $\nu_i \in \mathcal{T}_i$, and estimates the robustness function of $\hat{\phi}_i$ as $r_i^t = \min_{\nu_i\in\mathcal{T}_i}\left(\varrho^{\phi_{\nu_i}}(\boldsymbol{z}_{\nu_i}^{t-1})\right)$. Agent-$i$ communicates r_i^t and $\boldsymbol{z}_i^t(x_i(t), \boldsymbol{v}_i^{t-1})$ to all agents in ν_i, $\nu_i \in \mathcal{T}_i$, and receives their corresponding information. Agent-i either solves P_i^t if $r_i^t = \min_{j\in\nu_i,\,\nu_i\in\mathcal{T}_i}\left(r_j^t\right)$ or retains its input sequence from $t{-}1$, that is $\boldsymbol{v}_i^t{=}\boldsymbol{v}_i^{t-1}$. Alg. 2 summarizes this distributed strategy, and Prop. 3 highlights its benefits.

Proposition 3 *Suppose each agent-i solves P_i^0 at $t{=}0$ and executes Alg. 2 from $t{\geq}1$. Let $\mathcal{O}_t{\subset}\mathcal{V}$ collect indices of agents solving P_i^t at $t{\geq}1$, and assume P_i^0, $i \in \mathcal{V}$, P_i^t, with $i{\in}\mathcal{O}_t$ for $t{\geq}1$, are feasible. It holds: 1) if $i, j{\in}\mathcal{O}_t$ for some $1{\leq}t{\leq}N$, then $\mathcal{T}_i{\cap}\mathcal{T}_j{=}\emptyset$, and 2) collaborative tasks ϕ_ν, $\nu \in \mathcal{K}_\phi$, are minimally violated or satisfied with probability at least $1{-}\theta$.*

Algorithm 2 Distributed STL control of agent-i

for t in $1 : N$ **do**
  Compute $r_i^t = \min_{\nu_i\in\mathcal{T}_i}\left(\varrho^{\phi_{\nu_i}}(\boldsymbol{z}_{\nu_i}^{t-1})\right)$
  Communicate r_i^t, $\boldsymbol{z}_i^t(x_i(t), \boldsymbol{v}_i^{t-1})$ to $j \in \nu_i$, $\nu_i \in \mathcal{T}_i$
  Receive r_j^t, $\boldsymbol{z}_j^t(x_i(t), \boldsymbol{v}_i^{t-1})$ from $j \in \nu_i$, $\nu_i \in \mathcal{T}_i$
  if $r_i^t < r_j^t$ for all $j \in \nu_i$, $\nu_i \in \mathcal{T}_i$ **then**
    Solve P_i^t and store $(\boldsymbol{v}_i^t, \boldsymbol{z}_i^t)$
  else
    Update $\boldsymbol{v}_i^t \leftarrow \boldsymbol{v}_i^{t-1}$ and $\boldsymbol{z}_i^t \leftarrow \boldsymbol{z}_i^t(x_i(t), \boldsymbol{v}_i^{t-1})$
  Apply $u_i(t) = \sum_{k=0}^{t-1} \Gamma_i^{t,k} w_i(k) + v_i^t(t)$

V. Example

We revisit the 10-agent setup from [1, Sec. IV], where agents with single-integrator dynamics are assigned individual and collaborative STL tasks under Gaussian disturbances $w_i(t) \sim \mathcal{N}(0, 0.05I_2)$, $i \in \mathbb{N}_{[1,10]}$. We refer to [1, Sec. IV] for detailed description. Each agent must visit regions T_i and G_i within $\mathbb{N}_{[10,50]}$ and $\mathbb{N}_{[70,100]}$, respectively, while avoiding obstacles in $\mathcal{X}$. Collaborative tasks require clique members to rendezvous within $\mathbb{N}_{[0,100]}$. We recall the clique set $\mathcal{K}_\phi = \{(1), \dots, (10), (1,2,3), (1,5), (3,4), (4,5), (5,6), (4,7), (6, 8), (6,9), (7,8), (8,10), (9,10)\}$. The STL specification spans 100 steps and must be satisfied with probability 95%.

PR-scaling and feedback design: We generate 100 training and 100 calibration disturbance sequences (length 100) and run Alg. 1 for $\tau_{\max} = 4$ to compute $(C^*, \boldsymbol{\Gamma}^*)$ using GUROBI. Average solve times for $P(C)$ and $P(\boldsymbol{\Gamma})$ are under 1 sec and 20 min, respectively (Intel i7, 32 GB RAM). Using the calibrated disturbance data and Prop. 2, we compute PRs $\mathbb{B}_\infty(q/C_\nu^*)$ with $0.7398 \leq q/C_\nu^* \leq 0.8985$, $\nu \in \mathcal{K}_\phi$.

Distributed STL synthesis: To meet the 95% satisfaction target, we tighten the underlying robustness functions using Thm. 1 and Prop. 2, with Lipschitz constants ranging from 0.046 to 1 across cliques (Lemma 2). We solve P_i^0 for all i and apply Alg. 2, where agents iteratively solve P_i^t, $t \in \mathbb{N}_{[1,100]}$, exchanging estimates of robustness functions within cliques. Fig. 1 shows results from one experiment. Out of 100 runs, 97 satisfied the STL formula. Fig. 2 (right) shows agent selection frequency during coordination. Our code is available at [26].

Comparison with [1]: The method in [1] relies on a union-bound-based uncertainty quantification providing conservative PRs for the error trajectories, which limits the specified satisfaction probability to 70%. In contrast, the proposed data-driven approach yields considerably tighter PRs with high confidence, depending on the number of

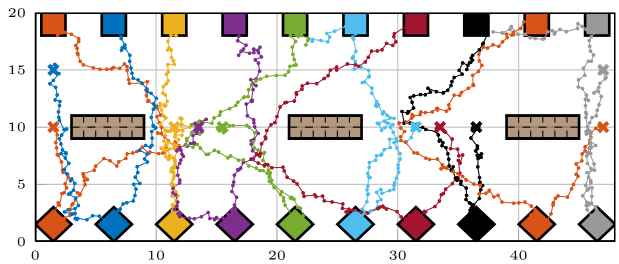

Fig. 1: Simulation under Alg. 2 with $x_i(0)$ as crosses, T_i areas as diamonds, G_i areas as boxes, and 3 brown obstacles.

available samples. Fig. 2 (left) compares the 70% l_2 PR (red circle) from [1] with the data-driven 95% l_∞-based PR obtained using 10^4 new calibration samples and a tightened 97% quantile, guaranteeing over 99.9999% confidence $1-\beta$ (see Rem. 1). Unlike the offline design in [1], the new method enables fully distributed coordination via Alg. 2.

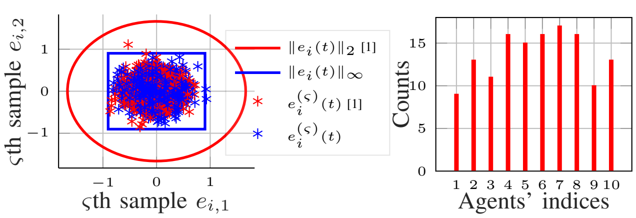

Fig. 2: Left: Red circle shows the PR for $\boldsymbol{e}_i(t)$ from [1], computed via a union bound with 70% probability; blue box denotes the proposed 95% ℓ_∞-based PR with confidence $1-\beta \gg 99.99\%$ (see Prop. 2 and Rem. 1). Right: Histogram of agents during the experiment of Fig. 1 under Alg. 2.

VI. Conclusion

We propose distribution-free control synthesis for stochastic linear MAS under chance-constrained collaborative STL specifications. Exploiting linearity, we decompose each agent's dynamics into deterministic and error components, following a disturbance-feedback (DF) and feedforward control policy. We design DF via CVaR-based optimization on training samples and use conformal prediction on calibration data to quantify prediction regions for error trajectories, enabling us to address chance constraints through Lipschitz tightening. This relaxation yields a centralized deterministic problem, whose solution provides the feedforward inputs. To improve scalability, we decompose it into agent-level MPC subproblems, resulting in a distributed control architecture. Compared with [1], the approach provides tighter uncertainty bounds and enhances scalability via distributed coordination.

References

[1] E. E. Vlahakis, L. Lindemann, P. Sopasakis, and D. V. Dimarogonas, "Probabilistic tube-based control synthesis of stochastic multi-agent systems under signal temporal logic," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 1586–1592.

[2] D. Sun, J. Chen, S. Mitra, and C. Fan, "Multi-agent motion planning from signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3451–3458, 2022.

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

[4] A. Donzé and O. Maler, "Robust Satisfaction of Temporal Logic over Real-Valued Signals," in *8th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS, 2010*, Klosterneuburg, Austria, 2010, pp. 92–106.

[5] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, "Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3324–3331, 2019.

[6] S. Safaoui, L. Lindemann, D. V. Dimarogonas, I. Shames, and T. H. Summers, "Control Design for Risk-Based Signal Temporal Logic Specifications," *IEEE L-CSS*, vol. 4, no. 4, pp. 1000–1005, 2020.

[7] L. Lindemann, G. J. Pappas, and D. V. Dimarogonas, "Reactive and Risk-Aware Control for Signal Temporal Logic," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5262–5277, 2022.

[8] D. Sadigh and A. Kapoor, "Safe Control under Uncertainty with Probabilistic Signal Temporal Logic," in *Proceedings of Robotics: Science and Systems*, jun 2016.

[9] S. Jha, V. Raman, D. Sadigh, and S. A. Seshia, "Safe Autonomy Under Perception Uncertainty Using Chance-Constrained Temporal Logic," *Journal of Automated Reasoning*, vol. 60, no. 1, pp. 43–62, 2018.

[10] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer Science & Business Media, 2005.

[11] L. Lindemann, Y. Zhao, X. Yu, G. J. Pappas, and J. V. Deshmukh, "Formal verification and control with conformal prediction," 2024. [Online]. Available: https://arxiv.org/abs/2409.00536

[12] X. Qin, Y. Xia, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification of cyber-physical systems using surrogate models and conformal inference," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, 2022, pp. 116–126.

[13] L. Lindemann, X. Qin, J. V. Deshmukh, and G. J. Pappas, "Conformal prediction for stl runtime verification," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, ser. ICCPS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 142–153.

[14] X. Yu, Y. Zhao, X. Yin, and L. Lindemann, "Signal temporal logic control synthesis among uncontrollable dynamic agents with conformal prediction," 2024. [Online]. Available: https://arxiv.org/abs/2312.04242

[15] T. Kuipers, R. Tumu, S. Yang, M. Kazemi, R. Mangharam, and N. Paoletti, "Conformal off-policy prediction for multi-agent systems," 2024. [Online]. Available: https://arxiv.org/abs/2403.16871

[16] E. E. Vlahakis, L. Lindemann, and D. V. Dimarogonas, "Conformal data-driven control of stochastic multi-agent systems under collaborative signal temporal logic specifications," 2025. [Online]. Available: https://arxiv.org/abs/2504.04615

[17] R. J. Tibshirani, R. F. Barber, E. J. Candès, and A. Ramdas, "Conformal prediction under covariate shift," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] V. Vovk, "Conditional validity of inductive conformal predictors," in *Proc. Asian Conf. Mach. Learn.*, vol. 25, 2012, pp. 475–490.

[19] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at risk," *Journal of Risk*, vol. 3, pp. 21–41, 2000.

[20] J. Orlin, "Contentment in graph theory: Covering graphs with cliques," *Indagationes Mathematicae*, vol. 80, no. 5, pp. 406–424, 1977.

[21] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

[22] M. Cleaveland, I. Lee, G. J. Pappas, and L. Lindemann, "Conformal prediction regions for time series using linear complementarity programming," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, pp. 20 984–20 992, 2024.

[23] E. E. Vlahakis, L. Lindemann, P. Sopasakis, and D. V. Dimarogonas, "Conformal prediction for distribution-free optimal control of linear stochastic systems," *IEEE L-CSS*, vol. 8, pp. 2835–2840, 2024.

[24] A. B. Kordabad, E. E. Vlahakis, L. Lindemann, S. Gros, D. V. Dimarogonas, and S. Soudjani, "Data-driven distributionally robust control for interacting agents under logical constraints," 2025, [Online]: https://arxiv.org/abs/2503.09816.

[25] A. Donzé and V. Raman, "BluSTL: Controller synthesis from signal temporal logic specifications," in *ARCH14-15: 1st & 2nd Int. Workshop on Appl. Verification for Cont. & Hybrid Syst.*, vol. 34. EasyChair, 2015, pp. 160–168.

[26] E. E. Vlahakis, https://github.com/lefterisvl83/CDC25, 2025.