Secure Control Under Linear Temporal Logic Constraints

Luyao Niu and Andrew Clark

***Abstract*— Cyber-physical systems must satisfy performance and safety criteria in spite of malicious attacks. In this paper, we investigate the problem of automatically synthesizing a control policy that maximizes the probability of satisfying safety and liveness constraints modeled using Linear Temporal Logic in the presence of an adversary. We develop a Stackelberg game framework, in which the controller chooses a probabilistic policy and the adversary chooses an attack strategy based on observation of that policy. We prove that maximizing the probability of satisfying safety and liveness constraints in this framework is equivalent to a worst-case reachability problem, and propose polynomial-time algorithms for computing the optimal policy. We illustrate our approach via numerical study.**

I. Introduction

Cyber-physical systems (CPS) are expected to perform increasingly complex tasks with autonomy. An emerging approach to CPS design is to formulate performance and safety properties as formal logic constraints and then automatically synthesize a controller satisfying these requirements [1]–[3]. Existing approaches of this type typically map the constraints to a formal language such as linear temporal logic (LTL), which express system goals such as liveness (e.g., "always eventually A") and safety ("always not A") [4]. These constraints are then combined with a model of the system behavior to obtain a finite state abstraction, enabling synthesis of control strategies via model-checking algorithms.

In addition to uncertainties and stochastic errors, CPS will also be subject to malicious attacks, including denial-of-service and injection of false sensor measurements and control inputs [5]. Such attacks will have different characteristics from uncertainties and errors. Unlike stochastic errors, adversaries will exhibit strategic behaviors, and in particular may adapt their strategies to maximize impact against a given controller. At the same time, unlike modeling uncertainties, adversaries will have limited information regarding the system state and control inputs, making techniques such as randomized control strategies potentially effective in mitigating attacks. Due to these factors, control strategies that are designed for safety and performance under errors and uncertainties may be suboptimal against intelligent attacks. At present, however, automatic synthesis of control systems in adversarial scenarios has received little research attention.

In this paper, we investigate the problem of selecting an optimal control strategy for a probabilistic autonomous system to satisfy safety and liveness constraints, which are commenly required for real world CPS, modeled using LTL in the presence of an adversary. We consider adversaries who tamper with control inputs based on the current system state. We assume the information structure to be concurrent Stackelberg game, in which the controller and adversary take actions simultaneously and jointly control the system transitions. Stackelberg games are popular models in security domain [6]–[8] and turn-based Stackelberg games have been used to construct model checkers [9] and compute control strategy [10], however, to the best of our knowledge, control synthesis in concurrent Stackelberg setting has not be investigated. We make the following specific contributions:

- We formulate a model of the interaction between the CPS and attacker via a stochastic game that describes the system dynamics and the effects of inputs from the controller and adversary. We give a heuristic algorithm for abstracting the stochastic game based on given system and adversary models.
- We consider the problem of maximizing the worst-case probability of satisfying an LTL specification consists of liveness and safety properties. For safety and liveness constraints, we demonstrate that this problem is equivalent to a zero-sum stochastic Stackelberg game, in which the controller chooses a policy to maximize the probability of reaching a desired set of states and the adversary chooses a policy to minimize that probability.
- We propose a novel algorithm for computing an optimal stationary policy, defined as a policy that depends only on the current system state. Our approach is based on iteratively computing the best-response at each system state. We prove that our approach converges to a Stackelberg equilibrium and characterize the convergence rate of the algorithm.
- Our approach is illustrated through simulation study, which shows that the control strategy obtained using our proposed approach outperforms existing techniques for satisfying the same specifications on Markov decision processes (MDPs) that do not consider the adversary's presence.

The remainder of this paper is organized as follows. Section II presents related work. Section III gives background on temporal logic, stochatic games and Stackelberg games. Section IV introduces the system model. Section V presents our problem formulation and solution algorithm. Section VI contains an illustrative motion planning case study. Section VII concludes the paper.

L. Niu and A. Clark are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA. {lniu,aclark}@wpi.edu

This work was supported by NSF grant CNS-1656981.

II. Related Work

Temporal logics such as linear temporal logic and computation tree logic (CTL) are widely used to specify and verify system properties [4], especially complex system behaviors [3], [11], [12]. Sensor-based temporal logic motion planning to conduct high level task specifications is studied in [11], [13], [14]. An online motion planning algorithm is proposed for dynamical systems to synthesize controllers or control strategies to satisfy a class of temporal logic specifications, while a receding horizon based framework that incorporates a class of LTL specifications is investigated in [3]. Feedback control design for deterministic system models specified by LTL formulas is studied in [15], while the algorithms for probabilistic system models are investigated in [4]. Control synthesis under PCTL constraints has been investigated in [16]–[18]. These existing works do not consider the impact of malicious attacks.

MDPs are widely used as finite state abstractions to capture the non-determinism and probabilistic behaviors of CPS [1], [2], which enables the application of off-the-shelf model checking algorithms for temporal logic [3], [15], [17], [19]. Robust control of MDP under uncertainties has been extensively studied [20]–[23]. However, robust MDPs are typically used to model uncertainties such as environmental disturbances and modeling errors, which are fundamentally different from malicious attacks.

Stochastic games, as generalization of MDP, model strategic interactions between multiple players. Turn-based two-player stochastic games have been used to construct model checkers [9] and abstraction-refinement framework for model checking [24]–[26]. Unlike the turn-based games studied in [9], [24]–[26], however, we consider a different information structure, in which both players take actions simultaneously at each system state.

Several existing works focus on computing Nash equilibria of stochastic games [27]–[29]. In the present paper, however, we consider a Stackelberg setting in which the adversary chooses an attack strategy based on the control policy selected by the system. A stochastic hybrid Stackelberg game with different information structure was presented in [10]. In the work of [10], the authors present an asymmetric information pattern similar to turn-based games that favors the adversary and derive a pure control strategy. However the information pattern in this paper allows the controller to potentially obtain advantage since we allow simultaneous action (see [6] for a simple example) and leads to a more general class of control strategies, i.e., mixed strategy. While stochastic Stackelberg security games have been studied in [7], [8], to the best of our knowledge they do not consider safety or liveness constraints. We also present algorithms for computing Stackelberg equilibria under reachability constraints that are not available in the literature to the best of our knowledge.

Game- and control-theoretic methods for CPS security have attracted recent research interest. Secure estimation and control for CPS is studied in [5], [30]. Game theoretic methods for CPS security and privacy are surveyed in [31]. Efficient algorithms for security games are presented in [6]. Resilient control using game theory is considered in [32].

III. Preliminaries

In this section, we present background on LTL, stochastic games and Stackelberg games.

A. Linear Temporal Logic

In this paper, control specifications are modeled using LTL. An LTL formula consists of [4], [33]

- a set of atomic propositions Π;
- Boolean operators: negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$).;
- temporal operators: next (X) and until ($\mathcal{U}$).

An LTL formula thus is defined inductively as follows:

$$\phi = True \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid \phi_1 \ \mathcal{U} \ \phi_2.$$

Other operators can be defined accordingly. In particular, implication ($\implies$) operator can be descirbed as $\neg\phi \vee \psi$, eventually (F) operator $F\phi$ can be described as $F\phi = True \ \mathcal{U} \ \phi$, and always ($G$) operator $G\phi$ can be described as $G\phi = \neg F \neg\phi$.

The semantics of LTL formulas are defined over infinite words in 2^{Π}. Informally speaking, $G\phi$ is true if and only if ϕ is true for the current time step and all the future time. $F\phi$ is true if ϕ is true at some future time. $X\phi$ is true if and only if ϕ is true in the next time step. A word η satisfying an LTL formula ϕ is denoted as $\eta \models \phi$.

B. Stochastic Games

Stochastic games generalize MDPs and are widely used for control synthesis under temporal logic constraints [34]–[36]. In these games, there are two or more players whose utilities are coupled, either due to players' action jointly controlling transition probabilities or coupling of reward functions. In the following, we present background on finite state/action space stochastic games involving two players, i.e., the controller and adversary.

Definition 1. *(Stochastic Game): A stochastic game $\mathcal{SG}$ is a tuple $\mathcal{SG} = (S, U_C, U_A, Pr, s_0, \Pi, \mathcal{L})$, where $S = \{1, 2, \cdots, n\}$ is a finite set of states, U_C is a set of actions of the controller, U_A is a set of actions of an adversary, $Pr : S \times U_C \times U_A \times S \to [0, 1]$ is a transition function where $Pr(s, u_C, u_A, s')$ is the probability of a transition from state s to state s' when the controllers action is u_C and the adversarys action is u_A. We let $U_C(s)$ and $U_A(s)$ denote the subset of controller actions and adversary actions available at state s, respectively. $s_0 \in S$ is the initial state. Π is a set of atomic propositions. $\mathcal{L} : S \to 2^{\Pi}$ is a labeling function, which maps each state to a set of propositions that are realized to be true in Π.*

We define a control policy for the controller $\mu : S^* \to [0, 1]^{|U_C|}$, which maps a sequence of states $S^* = \{s_0, s_1, \cdots, s_k\}$ to a probability distribution over the set of actions $U_C(s_k)$ available at state s_k. A control policy

τ for the adversary is defined as $\tau : S^* \to [0,1]^{|U_A|}$. A control policy is *stationary* if it is only a function of the current state, i.e., $\mu : s \to [0,1]^{|U_C|}$ is only dependent on the last state of S^*. A stationary policy is said to be *proper* if the probability of reaching the terminal state after finite steps is positive under this policy. We let $\mu(s)$ denote the function $\mu(s) : U_C(s) \to [0,1]^{|U_C|}$ that maps each action u_C to $\mu(s,u_C) \in [0,1]$; an analogous definition holds for adversary policies $\tau(s)$. A pair of stationary policies μ and τ induces a Markov chain whose state set is S and transition probability from state s to s' is $P^{\mu\tau}(s,s') = \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s,u_C)\tau(s,u_A)Pr(s,u_C,u_A,s')$. An infinite sequence $w^{\mu\tau}_{\mathcal{SG}} = s_0 s_1 \cdots$ for a given initial state s_0 on $\mathcal{SG}$ under policies μ and τ is called a *path* on $\mathcal{SG}$ if there exist u_C and u_A such that $\mu(k,u_C) > 0, \tau(k,u_A) > 0$ for all k and $Pr(s_k,u_C,u_A,s_{k+1}) > 0,\ \forall k$. Given a path $w^{\mu\tau}_{\mathcal{SG}}$, a word is generated as $\mathcal{L}(w^{\mu\tau}_{\mathcal{SG}}) = \mathcal{L}(s_0)\mathcal{L}(s_1)\cdots$. Denote the set of all infinite paths generated under policies μ and τ on $\mathcal{SG}$ as $W^{\mu\tau}_{\mathcal{SG}}$. Then the probability of satisfying an LTL formula ϕ under policies μ and τ on $\mathcal{SG}$ is denoted as $Pr^{\mu\tau}_{\mathcal{SG}} = Pr^{\mu\tau}_{\mathcal{SG}}\{w^{\mu\tau}_{\mathcal{SG}} \in W^{\mu\tau}_{\mathcal{SG}} : \mathcal{L}(w^{\mu\tau}_{\mathcal{SG}}) \models \phi\}$.

In the following, we briefly review a subclass of stochastic games denoted Stackelberg games, involving two players [37], [38]. In the Stackelberg setting, player 1 (also called leader) commits to a strategy first. Then player 2 (also known as follower) observes the strategy of the leader and plays its best response. The information structure under Stackelberg setting can be classified into the following two categories.

- *Turn-based games*: Only one player is allowed to take action at each time step.
- *Concurrent games*: All the players take actions simultaneously at each time step.

Unlike [9], [24]–[26], which are turn-based, in this paper, we focus on concurrent games.

The concept of Stackelberg equilibrium is defined formally in the following.

Definition 2. *(Stackelberg Equilibrium): Denote the utility that the leader gained in a stochastic game $\mathcal{SG}$ under leader follower strategy pair (μ,τ) as $\mathcal{Q}_{\mathcal{SG}}(\mu,\tau)$. A pair of leader follower strategy (μ,τ) is a Stackelberg equilibrium if leader's strategy μ is optimal given that the follower observes its strategy and plays its best response.*

$$(\mu,\tau) = \underset{\mu' \in M, \tau' \in \mathcal{BR}(\mu')}{\operatorname{argmax}} \mathcal{Q}_{\mathcal{SG}}(\mu',\tau'), \tag{1}$$

where M is the set of all admissible policies of the controller and $\mathcal{BR}(\mu')$ denotes the best response to palyer 1 strategy μ' palyed by player 2.

IV. System Model

In this section, we present our system model. Consider a finite state discrete time system with dynamics as

$$x(t+1) = f(x(t),u_C(t),u_A(t),\vartheta(t))\ \forall t = 0,1,\cdots, \tag{2}$$

where $x(t)$ is the system state, $u_C(t)$ and $u_A(t)$ are the control inputs from the controller and adversary, respectively, and $\vartheta(t)$ is stochastic disturbance.

In this work, we focus on the scenarios where a strategic adversary can manipulate the control input of the system by tampering with control inputs. For instance, the control input of the system consists of the original input from the controller $u_C(t)$ and the false data injected by the adversary $u_A(t)$, i.e., $u(t) = u_C(t) + u_A(t)$.

We adopt the concurrent Stackelberg setting in our problem, which is widely used in security domains to model the interaction between the defender and attacker. Here, we let the controller be the leader and the adversary be the follower. The contoller first commits to a mixed strategy by considering the potential presence of the adversary. The adversary then can observe how the controller behaves over time and then plays its best response to maximize its own utility.

A specification modeling liveness and safety constraints is given as an LTL formula ϕ over a set of atomic propositions Π in form of $\phi = GF\pi \wedge \psi$, where the livenesss constraint $GF\pi$ requires that the system satisfy a property $\pi \in \Pi$ infinitely often and the safety constraint ψ requires that the system remain within a certain safe region. The region is partitioned into a finite set of sub-regions with respect to the atomic proposition set Π. Assume each system state x can be mapped to a sub-region. Hence, we can generate a stochastic game as an abstraction for the dynamical system (2).

To generalize the proposed approach in this paper, motivated by [21], [39], we use the following algorithm to generate a finite state/action stochastic game from continuous system. The difference is that we need to consider the presence of adversary. For each sub-region X_i and pair of (control, adversary) inputs (u_C,u_A), we randomly select K samples in X_i and adversary and control inputs that map to u_C and u_A. We compute the state X_j that the system transitions to, the transition probability $Pr(X_i,u_C,u_A,X_j)$ can be updated as the fraction of samples located in sub-region X_j. To approximate the transition probability, Monte Carlo simulation or particle filter can be used [21], [39], [40].

V. Problem Formulation - Maximizing Satisfaction Probability

In this section, we formulate the problem of maximizing the probability of satisfying a given LTL specification in the presence of an adversary. We first present the problem formulation, and then give solution algorithms for computing the optimal control policy. Due to space constraint, the proofs for propositions and lemmas can be found in [41].

A. Problem Statement

For a system with the presence of an adversary modeled as (2), the problem is defined as follows. Given a stochastic game $\mathcal{SG}$ and an LTL specification ϕ, compute a control policy μ that maximizes the probability of satisfying the specification ϕ under any adversary policy τ, i.e.,

$$\max_{\mu} \min_{\tau} Pr^{\mu\tau}_{\mathcal{SG}}(\phi). \tag{3}$$

We assume that $\phi = GF\pi \wedge \psi$, letting V denote the set of states that satisfy π and W denote the set of states that

Algorithm 1 Algorithm for constructing a stochastic game approximation of a continuous system.

procedure CREATE_STOCHASTIC_GAME($X_1, \ldots, X_n$)
  Input: Set of sub-regions $X_1, \ldots, X_n$
  Output: Stochastic game $\mathcal{SG} = (S, U_C, U_A, Pr, s_0, \Pi, \mathcal{L})$
  Initialize K
  $S = \{X_1, \ldots, X_n\}$ and $\mathcal{L}$ is determined accordingly
  Generate control primitive sets $U_C = \{u_{C_1}, u_{C_2} \cdots, u_{C_\Xi}\}$ and $U_A\{u_{A_1}, u_{A_2} \cdots, u_{A_\Gamma}\}$
  for $i = 1, \ldots, n$ **do**
    for All $u_C \in U_C$ and $u_A \in U_A$ **do**
      for $k = 1, \ldots, K$ **do**
        $x \leftarrow$ sampled state in X_i
        $\hat{u}_C, \hat{u}_A \leftarrow$ sampled inputs from u_C, u_A
        $j \leftarrow$ region containing $f(x, \hat{u}_C, \hat{u}_A, \vartheta)$
        Invoke particle filter to approximate transition probabilities Pr between sub-region i and j for all i and j.
      end for
    end for
  end for
end procedure

violate ψ (unsafe states). While the initial state is encoded in the stochastic game $\mathcal{SG}$ by Definition 1, for ease of exposition we make dependence on the initial state explicit by letting $Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s)$ denote the probability of satisfying ϕ using control and adversary policies μ and τ, respectively, when the initial state is s.

In the stochastic game we formulated, the players involved are the controller (leader) and the adversary (follwer). Since the objectives of the palyers are contradictive, the stochastic game is thus viewed as a zero-sum stochastic two-player Stackelberg game. In this game, the controller first chooses a randomized policy μ while taking into account the presence of the adversary and its response, and the adversary observes empirical μ and selects a policy τ to minimize $Pr^{\mu\tau}_{\mathcal{SG}}(\phi)$. We rely on the solution concept of Stackelberg equilibrium to solve the problem defined above. The policies μ and τ that achieve the max-min value of (3) can be interpreted as an equilibrium in a zero-sum Stackelberg game between the controller and adversary. We restrict our attention to the class of stationary policies, leaving the general case for future work. We have the following preliminary lemma.

Lemma 1. *Let* $v_s = \max_\mu \min_\tau Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s)$. *Then*

$$v_s = \max_{\mu} \min_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} \mu(s, u_C)\tau(s, u_A)v_{s'} \cdot Pr(s, u_C, u_A, s') \quad (4)$$

The proof is omitted due to space constraints. Our approach to computing the solution to (3) is as follows. We first characterize and compute the set of states $\mathcal{G}$ that are guaranteed to satisfy ϕ, regardless of the adversary's actions. We then prove that the max-min probability of (3) is equivalent to maximizing (over μ) the worst-case probability (over the set of adversary policies τ) of reaching $\mathcal{G}$. Finally, we present an efficient algorithm for computing a policy μ that maximizes the worst-case probability of reaching $\mathcal{G}$.

Definition 3. *A state s is defined to be* accepting *if there exists a stationary control policy μ such that, for any stationary adversary policy, the specification ϕ is guaranteed to be satisfied with probability 1 when the system starts in state s.*

We denote the accepting states as $\mathcal{G}$. The following preliminary results that give properties of the set $\mathcal{G}$.

Lemma 2. *Let $R = S \setminus \mathcal{G}$ and $s \in S$. Suppose that, for any control policy μ, there exists an adversary policy τ such that s reaches R with nonzero probability. Then $s \in R$.*

We now describe a procedure for computing the set of accepting states. Algorithms 2 and 3 provide two subroutines that are used. Algorithm 2 computes the sets of states and actions that result in transitions to a given set of states R with positive probability. Algorithm 3 computes the set of states that are reachable to a set V under any adversary policy. The main procedure for computing $\mathcal{G}$ is given as Algorithm 4.

Algorithm 4 initially uses Algorithm 2 to remove from $\mathcal{G}$ all states that can be driven to an unsafe state with nonzero probability by any adversary strategy, as well as all actions that may cause transitions to unsafe states.

Algorithm 4 then computes a collection of states such that the system is guaranteed to reach V infinitely often without reaching an unsafe state. The approach is to iteratively remove actions from $U_C(s)$ that can cause a transition to a state in $S \setminus \mathcal{G}$. Nodes are removed from $\mathcal{G}$ if all actions cause a transition to $S \setminus \mathcal{G}$ with positive probability.

Algorithm 2 Computing states and actions that transition to $S \setminus \mathcal{G}$ with nonzero probability under any control policy.

procedure COMPUTE_UNSAFE($\mathcal{SG}$, R, $\tilde{\mathbf{U}}_\mathbf{C}$)
  Input: Stochastic game $\mathcal{SG}$, set of states R, set of feasible actions $\tilde{U}_C(s)$ for each state s
  Output: Set of states R' that reach R with positive probability, updated set of actions $\tilde{U}'_C(s)$
  Initialization: $R' \leftarrow R$, $\tilde{U}'_C(s) \leftarrow \tilde{U}_C(s)\ \forall s$, $x \leftarrow 1$
  while $x == 1$ **do**
    $x \leftarrow 0$
    for $s \in S \setminus R'$ **do**
      $\tilde{\mathbf{U}}'_\mathbf{C} \leftarrow \tilde{U}'_C(s) \setminus \{u_C : \sum_{u_A \in U_A(s)} \sum_{s' \in R'} Pr(s, u_C, u_A, s') > 0\}$
      if $\tilde{U}'_C(s) == \emptyset$ **then**
        $R' \leftarrow R' \cup \{s\}$, $x \leftarrow 1$
      end if
    end for
  end while
  return $(R', \tilde{\mathbf{U}}'_\mathbf{C})$
end procedure

Algorithm 3 Computing states that transition to V with positive probability under any adversary policy.

procedure COMPUTE_REACHABLE($\mathcal{SG}$, V, Q, $\tilde{\mathbf{U}}_\mathbf{C}$)
  Input: Stochastic game $\mathcal{SG}$, set of states to reach V, set of initial states Q, set of feasible actions $\tilde{U}_C(s)$ for each state s
  Output: Set of states T that are reachable to V with positive probability under any adversary strategy
  Initialization: $T \leftarrow \emptyset, x \leftarrow 1$
  while $x == 1$ **do**
    $x \leftarrow 0$
    for $s \in Q \setminus T$ **do**
      for $u_A \in U_A(s)$ **do**
        $Z_{s,u_A} \leftarrow \{s' : \sum_{u_C \in \tilde{U}_C(s)} Pr(s, u_C, u_A, s') > 0\}$
      end for
      if $Z_{s,u_A} \cap (T \cup V) \neq \emptyset \ \forall u_A \in U_A(s)$ **then**
        $T \leftarrow T \cup \{s\}$, $x \leftarrow 1$
      end if
    end for
  end while
  return T
end procedure

Algorithm 4 Computing accepting states of stochastic game.

1: **procedure** COMPUTE_ACCEPTING_STATES($\mathcal{SG}$, R, V)
2: **Input**: Stochastic game $\mathcal{SG}$, set of unsafe states R, set of states V that must be reached infinitely often
3: **Output:** Set of accepting states $\mathcal{G}$ and set of actions $\tilde{U}_C(s)$ for all $s \in \mathcal{G}$
4: **Initialization:** $R' \leftarrow R$, $\mathcal{G} \leftarrow \emptyset$, $V' \leftarrow V$, $\tilde{U}_C(s) \leftarrow U_C(s)$, $x \leftarrow 0$
5: $(R', \tilde{\mathbf{U}}_\mathbf{C}) \leftarrow$ Compute_Unsafe($\mathcal{SG}$, R', $\tilde{\mathbf{U}}_\mathbf{C}$)
6: **while** $x == 0$ **do**
7: $T \leftarrow$ Compute_Reachable($\mathcal{SG}$, V, S, $\tilde{\mathbf{U}}_\mathbf{C}$)
8: $T' \leftarrow \emptyset$
9: **while** $T' \neq T$ **do**
10: $T' \leftarrow T$
11: $R' \leftarrow R' \cup (S \setminus T)$
12: $(R', \tilde{\mathbf{U}}_\mathbf{C}) \leftarrow$ Compute_Unsafe($\mathcal{SG}$, R', $\tilde{\mathbf{U}}_\mathbf{C}$)
13: $T \leftarrow T \setminus R'$.
14: $T \leftarrow$ Compute_Reachable($\mathcal{SG}$, V, T, $\tilde{\mathbf{U}}_\mathbf{C}$)
15: **end while**
16: $V' \leftarrow V \cap T$
17: **if** $V' == V$ **then**
18: $x \leftarrow 1$
19: **else**
20: $V \leftarrow V'$
21: **end if**
22: **end while**
23: $\mathcal{G} \leftarrow T$
24: **return** $(\mathcal{G}, \tilde{\mathbf{U}}_\mathbf{C})$
25: **end procedure**

The correctness of the algorithm is shown as follows. We first have the following preliminary lemmas that describe the behavior of Algorithms 2 and 3.

Lemma 3. *Suppose that the set of states R given as input to Algorithm 2 is disjoint from $\mathcal{G}$. Then (i) for the set of actions $\tilde{\mathbf{U}}_\mathbf{C}$ returned by the algorithm and any policy μ that selects an action from $U_C(s) \setminus \tilde{U}_C(s)$ with positive probability at any node s, there exists an adversary policy τ such that $Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s) < 1$, and (ii) the set of states R' returned by Algorithm 2 is disjoint from $\mathcal{G}$.*

Lemma 4. *Let T denote the set returned by Algorithm 3. There exists a stationary policy μ over the action space $\tilde{U}_C$ that ensures that a state in V is reached with positive probability under any stationary adversary policy when the initial state is in T. If $s \notin T$, then for any policy μ, there exists a stationary adversary policy τ that ensures that the set V is not reached under any stationary policy over $\tilde{U}_C$.*

We now prove the correctness of Theorem 1.

Theorem 1. *Let $\tilde{\mathcal{G}}$ denote the set returned by Algorithm 4. Then $\tilde{\mathcal{G}} = \mathcal{G}$.*

Proof. To prove the correctness of Algorithm 4, we first show that no accepting states will be removed. Then we show that the set of states from which the system can be driven to unsafe states under some adversary policy is removed.

First, we show that $\tilde{\mathcal{G}} \subseteq \mathcal{G}$. Let $\tilde{\mathbf{U}}_\mathbf{C}$ be the set of actions returned by Algorithm 4. Let μ be a stationary policy that selects each action in $\tilde{U}_C(s)$ with positive probability. For any adversary policy, the induced Markov chain results in a graph with an edge between two states if there is a positive transition probability between them. By Line 14 and Lemma 4, each state in the graph is connected to a state in V', where V' is a set of states in V, and each state in V' is connected to at least one other state in V'. Furthermore, by Line 7 and Lemma 3, no state in the set $\tilde{\mathcal{G}}$ is connected to any state outside $\tilde{\mathcal{G}}$. Hence, under the policy μ, if the system is initially at a state in $\mathcal{G}$, then it will remain in $\mathcal{G}$ for all time; furthermore, at each time, connectivity to V' implies that the system will eventually reach a state in V'. Thus the condition $GF\pi$ is satisfied with probability 1. Furthermore, by Line 7 and Lemma 3, no state in $\mathcal{G}$ is connected to any state in W, ensuring that ψ is satisfied with probability 1.

To complete the proof, we show that $\mathcal{G} \subseteq \tilde{\mathcal{G}}$, or equivalently, $(S \setminus \tilde{\mathcal{G}}) \subseteq (S \setminus \mathcal{G})$. We prove by induction, noting that the result holds trivially initially. We have that a state is removed from $\tilde{\mathcal{G}}$ either at Line 13 or Line 14 of the algorithm. If a state s is removed because $s \in R'$ at Line 13, then by Lemma 1 and induction, the set R' is disjoint from $\mathcal{G}$, and hence $s \notin \mathcal{G}$. Furthermore, if s is removed from Line 14, then by Lemma 4, there exists an adversary policy that prevents the system starting from s from reaching $V \cap \tilde{\mathcal{G}}$, and hence from reaching $V \cap \mathcal{G}$ by the inductive assumption that $\mathcal{G} \subseteq \tilde{\mathcal{G}}$. Hence $s \notin \mathcal{G}$. Since these are the only two scenarios in which a state is removed from $\tilde{\mathcal{G}}$, we have that $(S \setminus \tilde{\mathcal{G}}) \subseteq (S \setminus \mathcal{G})$ and hence $\mathcal{G} \subseteq \tilde{\mathcal{G}}$. □

B. Computing the Optimal Policy

Our proposed solution is based on the following.

Proposition 1. *For any stationary control policy μ and initial state s, the minimum probability over all stationary adversary policies of satisfying the LTL formula is equal to the minimum probability over all stationary policies of reaching $\mathcal{G}$.*

Proposition 1 implies that the problem of maximizing the worst-case success probability can be mapped to the following equivalent problem. Define a stochastic game $\mathcal{SG}'$ as follows. For all nodes in $\mathcal{G} \cup R$, remove all outgoing edges and add a self-transition, which occurs with probability 1 regardless of the actions taken. The transition probabilities and action spaces of all other nodes are unchanged. The problem (3) is then equivalent to

$$\max_{\mu} \min_{\tau} Pr^{\mu\tau}_{\mathcal{SG}'}(\text{reach } \mathcal{G}) \tag{5}$$

The solution to (3) can be obtained from the solution to (5) by following the optimal policy μ^* for (5) at all states not in $\mathcal{G}$, and following the policy of choosing each strategy in $\tilde{\mathbf{U}}_{\mathbf{C}}$ returned by Algorithm 4 with positive probability for all states in $\mathcal{G}$.

Our approach for solving (5) is to first compute a value vector $\mathbf{v} \in \mathbb{R}^{|S|}$, where

$$v_s = \max_{\mu} \min_{\tau} Pr(\text{reach } \mathcal{G}|s).$$

The optimal policy can then be obtained from $\mathbf{v}$ by choosing the distribution μ that solves the optimization problem of Eq. (4) at each state s. Algorithm 5 gives an algorithm for computing $\mathbf{v}$. The idea of the algorithm is to initialize $\mathbf{v}$ to be zero except on $\mathcal{G}$, and then greedily update v_s at each iteration by computing the optimal Stackelberg policy at each state.

Algorithm 5 Algorithm for a control strategy that maximizes the probability of satisfying ϕ.

1: **procedure** MAX_REACHABILITY($\mathcal{SG}'$, $\mathcal{G}$)
2: **Input**: Stochastic game $\mathcal{SG}'$, set of states to be reached $\mathcal{G}$
3: **Output:** Vector $\mathbf{v} \in \mathbb{R}^{|S|}$, where $v_s = \max \min Pr(\text{reach } \mathcal{G}|s_0 = s)$
4: **Initialization:** $\mathbf{v}^0 \leftarrow 0$, $v^1_s \leftarrow 1$ for $s \in \mathcal{G}$, $v^1_s \leftarrow 0$ otherwise, $k \leftarrow 0$
5: **while** $\max\{|v^{k+1}_s - v^k_s| : s \in S\} > \delta$ **do**
6: $k \leftarrow k + 1$
7: **for** $s \notin \mathcal{G}$ **do**
8: $v^{k+1}_s \leftarrow$
9: $\max_\mu \min_\tau \Big\{ \sum_{s'} \sum_{u_C \in U_c(s)} \sum_{u_A \in U_A(s)} v_{s'} \mu(u_C)$
10: $\tau(u_A) Pr(s, u_C, u_A, s') \}$
11: **end for**
12: **end while**
13: **return** $\mathbf{v}$
14: **end procedure**

We note that the computation of Lines 8-10 can be performed in polynomial time by solving a linear program [42].

Corrolary 1. *The complexity of computing the optimal control policy is $O(|S|^2|U_C||U_A|)$.*

The following theorem shows that Algorithm 5 guarantees convergence to a Stackelberg equilibrium.

Theorem 2. *There exists $\mathbf{v}^\infty$ such that for any $\epsilon > 0$, there exists δ and K such that $||\mathbf{v}^k - \mathbf{v}^\infty||_\infty < \epsilon$ for $k > K$. Furthermore, $\mathbf{v}^\infty$ satisfies the conditions of $\mathbf{v}$ in Lemma 1.*

Proof. We first show that, for each s, the sequence $v^k_s : k = 1, 2, \ldots,$ is bounded and monotone. Boundedness follows from the fact that, at each iteration, v^k_s is a convex combination of the states of its neighbors, which are bounded above by 1. To show monotonicity, we induct on k. Note that $v^1_s \geq v^0_s$ and $v^2_s \geq v^1_s$ since $v^1_s = 0$ for $s \notin \mathcal{G}$ and $v^k_s \equiv 1$ for $s \in \mathcal{G}$.

Let μ^k_s denote the optimal control policy at state s and step k. We have that

$$\begin{aligned} v^{k+1}_s &\geq \min_\tau \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} v^k_{s'} \mu^k_s(u_C) \quad (6) \\ &\quad \cdot \tau(u_A) Pr(s, u_C, u_A, s') \\ &\geq \min_\tau \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} v^{k-1}_{s'} \mu^k_s(u_C) \quad (7) \\ &\quad \cdot \tau(u_A) Pr(s, u_C, u_A, s') \\ &= v^k_s \quad (8) \end{aligned}$$

Eq. (6) follows because the value of v^{k+1}_s, which corresponds to the maximizing policy, dominates the value achieved by the particular policy μ^k_s. Eq. (7) holds by induction, since $v^k_{s'} \geq v^{k-1}_{s'}$ for all s'. Finally, (8) holds by construction of μ^k_s. Hence v^k_s is monotone in k.

We therefore have that v^k_s is a bounded monotone sequence, and hence converges by the monotone convergence theorem. Let $\mathbf{v}^\infty$ denote the vector of limit points, so that we can select δ sufficiently small (to prevent the algorithm from terminating before convergence) and K large in order to satisfy $||\mathbf{v}^k - \mathbf{v}^\infty||_\infty < \epsilon$.

We now show that $\mathbf{v}^\infty$ is a Stackelberg equilibrium. Since v^k_s converges, it is a Cauchy sequence and thus for any $\epsilon > 0$, there exists K such that $k > K$ implies that $|v^k_s - v^{k+1}_s| < \epsilon$. By construction, this is equivalent to

$$\left| v^k_s - \max_\mu \min_\tau \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} \left[v^{k-1}_{s'} \mu(u_C) \cdot \tau(u_A) Pr(s, u_C, u_A, s') \right] \right| < \epsilon,$$

and hence $\mathbf{v}^\infty$ is within ϵ of a Stackelberg equilibrium for every $\epsilon > 0$. □

While this approach guarantees asymptotic convergence to a Stackelberg equilibrium, there is no guarantee on the rate

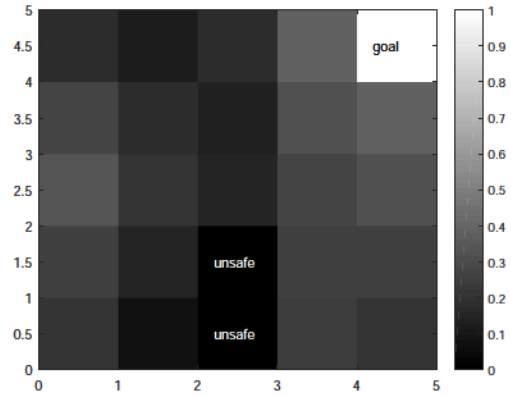

Fig. 1: Probability of satisfying LTL specification under optimal policy.

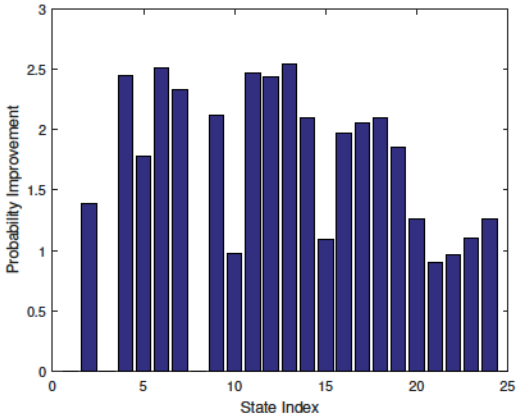

Fig. 2: Comparison of the probability of satisfying the LTL specification using proposed and baseline approaches. The proposed approach improves the satisfaction probability for all initial states.

of convergence. By modifying Line 8 of the algorithm so that v_s^{k+1} is updated if

$$\max_{\mu}\min_{\tau}\Bigg\{\sum_{s'}\sum_{u_C\in U_c(s)}\sum_{u_A\in U_A(s)}[v_{s'}\mu(u_C) \cdot \tau(u_A)Pr(s,u_C,u_A,s')]\Bigg\} > (1+\epsilon)v_s^k \quad (9)$$

and is constant otherwise, we derive the following result on the termination time.

Proposition 2. *The ϵ-relaxation of (9) converges to a value of* $\mathbf{v}$ *satisfying* $(1+\epsilon)\mathbf{v} > \mathbf{v}^{\infty}$ *within*

$$n\max_s\left\{\frac{\log\left(\frac{1}{x_s^0}\right)}{\log(1+\epsilon)}\right\}$$

iterations, where x_s^0 is the smallest positive value of x_s^k for $k = 0, 1, \ldots$.

VI. Case Study

In this section, we present a case study to demonstrate our proposed method. We consider the reach-avoid problem of a robot following standard discrete time unicycle model as follows:

$$\begin{aligned} x(t+1) &= x(t) + (u_{Cx}(t) + u_{Ax}(t) + \vartheta_v(t))\cos(\theta) \\ y(t+1) &= y(t) + (u_{Cy}(t) + u_{Ay}(t) + \vartheta_v(t))\sin(\theta) \\ \theta(t+1) &= \theta(t) + (u_{C\omega}(t) + u_{A\omega}(t) + \vartheta_\omega(t)), \end{aligned}$$

where the control input from the controller $u_C(t) = [u_{Cx}(t), u_{Cy}(t), u_{C\omega}(t)]^T$, the control input from the adversary $u_A(t) = [u_{Ax}(t), u_{Ay}(t), u_{A\omega}(t)]^T$ and $\vartheta(t) = [\vartheta_v(t), \vartheta_\omega(t)]^T$ is the stochastic disturbance.

Suppose the robot is moving in a $5m \times 5m$ region that is further divided into 25 sub-regions. For each sub-region, its location can be uniquely determined by the first two states of the robot $[x(t), y(t)]$. We set each sub-region to be a state in the stochastic game abstracted from the system. The actions available at each state include moving to a neighbor and staying in the same state. For each state, a labeling function $\mathcal{L}$ is used to map the state to one or several atomic propositions in proposition set $\Pi = \{\text{initial}, \text{unsafe}, \text{goal}\}$. In particular, the 25th state (top right) is mapped to goal and all the states are safe except the third and eighth states are mapped to unsafe. The detailed label for each state is shown in Fig. 1.

Let the robot be required to navigate from the initial state to the goal state while avoiding all unsafe states. Thus, we have the specification for the robot described in LTL formula as $\phi = GF\text{goal} \wedge G\neg\text{unsafe}$.

We first show the probability of satisfying the LTL specification starting from different state using our approach in Fig. 1. As we observe in Fig. 1, the only state that has probability 1 is the goal state, while the unsafe states have 0 probability of satisfying the specification. Other states are colored accordingly. In order to evaluate the benefit of incorporating resilience to adversaries, we compare the result obtained using our proposed method (Algorithm 5) and the result obtained using the method without considering the presence of the adversary. In particular, we calculate the optimal policy for the system without considering the potential impact of adversary, denoted μ_2, and set it as the baseline. Then we implement the policy of Algorithm 5, denoted μ_1. The comparison of the probability of satisfying the LTL specification starting from each state is shown in Fig. 2, where the vertical axis is equal to

$$(Pr_{\mathcal{SG}}^{\mu_1\tau_1}(\phi|s) - Pr_{\mathcal{SG}}^{\mu_2\tau_2}(\phi|s))/Pr_{\mathcal{SG}}^{\mu_2\tau_2}(\phi|s).$$

where τ_1 and τ_2 are the optimal adversary responses to μ_1 and μ_2, respectively. We see that the overall performance is improved under our proposed secure control method.

VII. Conclusion

In this paper, we investigated the problem of satisfying LTL safety and liveness specifications in the presence of malicious adversaries. We formulated a stochastic Stackelberg game for selecting a stationary control policy that maximizes the probability of satisfying the constraints in the presence

of an adversary who observes the policy and chooses a strategy to minimize the satisfaction probability. We mapped the problem to an equivalent reachability game, and proposed a deterministic polynomial-time algorithm for computing an optimal control strategy. Future work will consider more general LTL specifications and non-stationary control and adversary strategies.

REFERENCES

[1] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, and J. Kuchar, "Collision avoidance for unmanned aircraft using markov decision processes," in *AIAA guidance, navigation, and control conference*, 2010, p. 8040.

[2] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proceedings of Robotics: Science and Systems*, 2016.

[3] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning for dynamical systems," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2009, pp. 5997–6004.

[4] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.

[5] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2016.

[6] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 895–902.

[7] Z. Yin and M. Tambe, "A unified method for handling discrete and continuous uncertainty in Bayesian Stackelberg games," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012, pp. 855–862.

[8] Y. Vorobeychik and S. P. Singh, "Computing stackelberg equilibria in discounted stochastic games." in *AAAI*, 2012.

[9] T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games." in *TACAS*, vol. 13. Springer, 2013, pp. 185–191.

[10] J. Ding, M. Kamgarpour, S. Summers, A. Abate, J. Lygeros, and C. Tomlin, "A stochastic games framework for verification and control of discrete time stochastic hybrid systems," *Automatica*, vol. 49, no. 9, pp. 2665–2674, 2013.

[11] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Where's Waldo? sensor-based temporal logic motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 3116–3121.

[12] S. Karaman and E. Frazzoli, "Sampling-based motion planning with deterministic μ-calculus specifications," in *IEEE Conference on Decision and Control (CDC)*, 2009, pp. 2222–2229.

[13] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2689–2696.

[14] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Motion planning with dynamics by a synergistic combination of layers of planning," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 469–482, 2010.

[15] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.

[16] C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski, "Controller synthesis for probabilistic systems," in *Exploring New Frontiers of Theoretical Informatics*. Springer, 2004, pp. 493–506.

[17] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 396–409, 2012.

[18] A. Nikou, J. Tumova, and D. V. Dimarogonas, "Probabilistic plan synthesis for coupled multi-agent systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 766–10 771, 2017.

[19] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of markov decision processes with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[20] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[21] E. M. Wolff, U. Topcu, and R. M. Murray, "Robust control of uncertain markov decision processes with temporal logic specifications," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 3372–3379.

[22] J. Fu and U. Topcu, "Computational methods for stochastic control with metric interval temporal logic specifications," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 7440–7447.

[23] X. C. D. Ding, S. L. Smith, C. Belta, and D. Rus, "Ltl control in uncertain environments with probabilistic satisfaction guarantees," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3515–3520, 2011.

[24] M. Kattenbelt and M. Huth, "Verification and refutation of probabilistic specifications via games," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 4. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[25] T. Quatmann, C. Dehnert, N. Jansen, S. Junges, and J.-P. Katoen, "Parameter synthesis for markov models: Faster than ever," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2016, pp. 50–67.

[26] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker, "A game-based abstraction-refinement framework for markov decision processes," *Formal Methods in System Design*, vol. 36, no. 3, pp. 246–280, 2010.

[27] P.-Y. Chen, S.-M. Cheng, and K.-C. Chen, "Smart attacks in smart grid communication networks," *IEEE Communications Magazine*, vol. 50, no. 8, 2012.

[28] C. Y. Ma, N. S. Rao, and D. K. Yau, "A game theoretic study of attack and defense in cyber-physical systems," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 708–713.

[29] Y. Li, L. Shi, P. Cheng, J. Chen, and D. E. Quevedo, "Jamming attacks on remote state estimation in cyber-physical systems: A game-theoretic approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2831–2836, 2015.

[30] S. Zonouz, K. M. Rogers, R. Berthier, R. B. Bobba, W. H. Sanders, and T. J. Overbye, "SCPSE: Security-oriented cyber-physical state estimation for power grid critical infrastructures," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1790–1799, 2012.

[31] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 25, 2013.

[32] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Systems*, vol. 35, no. 1, pp. 46–65, 2015.

[33] E. M. Clarke, O. Grumberg, and D. Peled, *Model Checking*. MIT press, 1999.

[34] L. S. Shapley, "Stochastic games," *Proceedings of the national academy of sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[35] A. Condon, "The complexity of stochastic games," *Information and Computation*, vol. 96, no. 2, pp. 203–224, 1992.

[36] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer Science & Business Media, 2012.

[37] H. Von Stackelberg, *Marktform und gleichgewicht*. J. springer, 1934.

[38] D. Fudenberg and J. Tirole, "Game theory, 1991," *Cambridge, Massachusetts*, vol. 393, p. 12, 1991.

[39] M. Lahijanian, S. B. Andersson, and C. Belta, "A probabilistic approach for control of a stochastic system from ltl specifications," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 2236–2241.

[40] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[41] L. Niu and A. Clark, "Technical report: Secure control under linear temporal logic constraints," https://www.dropbox.com/s/4o848y0miyjucv0/ACC_18.pdf?dl=0, 2017.

[42] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, 2006, pp. 82–90.