2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

DoS-Resilient Multi-Robot Temporal Logic Motion Planning

Xiaowu Sun*† Rohitkrishna Nambiar*† Matthew Melhorn* Yasser Shoukry* Pierluigi Nuzzo**
*Department of Electrical and Computer Engineering, University of Maryland, College Park, MD
**Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA.

***Abstract*— We propose an efficient multi-robot motion planning algorithm for missions captured by linear temporal logic (LTL) specifications, in the presence of bounded disturbances and denial-of-service (DoS) attacks against the communication between robots and base stations. Given an LTL formula ψ, our goal is to construct robot trajectories, and associated control strategies, to satisfy ψ and continuously establish communication paths between robots and base stations despite the DoS attacks and the disturbances on the robot states. Our approach combines and extends results from robust control and efficient motion planning via satisfiability modulo convex programming (SMC). We first compute a feedback controller that rejects the disturbance together with a perturbation of the DoS-free workspace that accounts for the worst-case disturbance scenario. On the perturbed workspace, we formulate the planning problem as a feasibility problem over Boolean and convex constraints, respectively capturing the DoS-resilient mission constraints and the constraints on the nominal, disturbance-free, robot dynamics. Numerical results show the effectiveness of our algorithm in providing DoS-resilient plans that are robust to disturbances and support the execution of complex missions.**

I. Introduction

As multi-robot systems are increasingly being considered for a variety of mission-critical and safety-critical applications (e.g., monitoring, disaster relief, healthcare), accounting for the security implications of these technologies becomes key [1]. In fact, the specific nature of these multi-agent, networked autonomous systems, as well as their complexity, exposes them to a set of unprecedented threats. Attacks may range from passive eavesdropping of the communication channel for data interception, to active communication jamming for disrupting legitimate transmissions, or the injection of malicious robots in the swarm [2], [3], [4]. Devising effective methods to account for these threats since the early stages of the design process, rather than undesirably or expensively retrofitting existing designs, is an open challenge.

A major difficulty for providing security guarantees about these systems stems from the need to reason about the tight integration of discrete abstractions (e.g., high-level tasks, intermittent links) with continuous trajectories and lower-level dynamics [1], [5]. This integration can soon become daunting for complex, high-dimensional systems, since a vast hybrid, discrete/continuous space must be explored while accounting for complex geometries, motion dynamics, safety, and temporal goals. The difficulties are further exacerbated by the uncertainties, as in the majority of real-world scenarios, due to internal noise sources, model errors, and unknown or adversarial environments. In this paper, we address these challenges by focusing on the *resilient multi-robot motion planning problem* for complex missions captured by a high-level formal language, and in the presence of bounded disturbances and denial-of-service (DoS) attacks against the communication between robots and base stations.

Recent work has proposed defense mechanisms for multi-robot systems that can guarantee resilience to communication spoofing attacks [6], [7]. Security mechanisms against communication-jamming attacks have also been studied based on game-theoretic approaches [8], [9], [10] or multi-objective optimization [11]. Differently from these efforts, we consider a mission specified by a linear temporal logic (LTL) [12] formula ψ; we then aim to automatically generate dynamically-feasible robot trajectories, and associated control strategies, that satisfy ψ and guarantee continuous communication between robots and base stations despite the disturbance and the adversarial environment. To do so, we combine and extend results from robust control [13], which separate the concerns of disturbance rejection and trajectory planning, with a satisfiability modulo convex programming (SMC) approach [14], [15], [16], [17], which efficiently reasons about the combination of discrete and convex constraints. Our contributions can be summarized as follows:

- A *novel SMC encoding that enables encapsulating DoS-resilience constraints within the planning problem*, by capturing a notion of communication-based adjacency, in addition to physical adjacency, between workspace locations. The proposed encoding allows leveraging the efficiency of the SMC approach, which was previously applied to solve reach-avoid and LTL motion planning problems, showing more than two orders of magnitude improvement in execution time with respect to state-of-the-art techniques based on the RRT (Rapidly-exploring Random Trees) and EST (Expansive Space Trees) methods on high-dimensional problems [14], [16], [17].
- A *robust LTL motion planning formulation* that allows effectively incorporating disturbances in the robot system states. While LTL has shown to be capable of expressing a rich set of specifications (e.g., safety, progress, response, surveillance, and monitoring) and support algorithmic control synthesis for a variety of applications in robotics and autonomous systems [18], [19], [20], [21], [22], [23], traditional formulations of LTL motion planning do not effectively account for disturbances on the robot trajectories, and tend to become impractical, especially in the presence of adversarial environments.

Numerical results show the effectiveness of our approach

This work was partially sponsored by the NSF award #1837589 and by a collaborative project between the University of Maryland, College Park and the Northrop Grumman University Research Initiative.
† Equally contributing first authors.

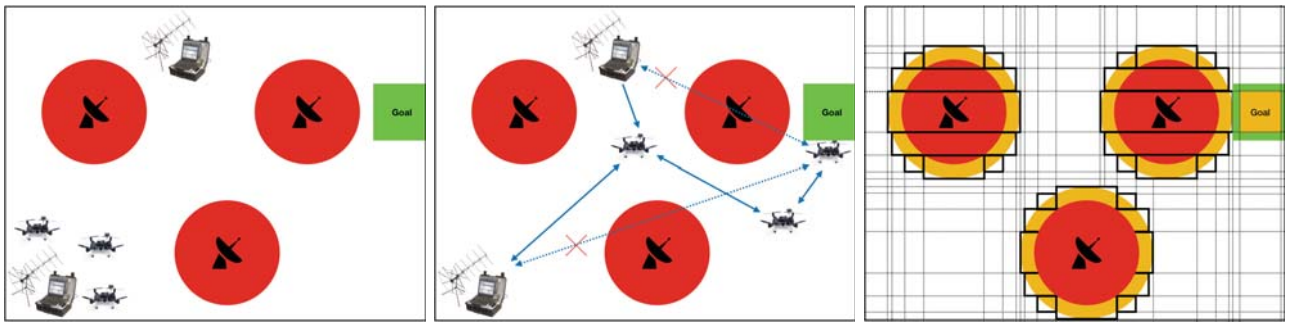

Fig. 1. (Left) Pictorial representation of a workspace that contains a team of three robots, two base stations, and three jamming radars. The mission is to move at least one of the robots to reach the goal location while maintaining communication between all the robots and at least one base station. (Middle) Any communication link that passes through a jamming area is considered under DoS attack. (Right) The workspace is perturbed (yellow) to account for disturbances and a coarse-grain discretization of the free space is computed.

in providing DoS-resilient plans that are robust to disturbances and support the execution of complex missions.

II. Problem Formulation

In this section, we introduce models for the robots and the adversarial environment. We consider a team of robots that move in a workspace $\mathcal{W} \subset \mathbb{R}^d$, where d can be 2 or 3, corresponding to a 2-dimensional or 3-dimensional workspace, respectively. We use $\|a\|$ to denote the infinity norm of vector a. Given two sets $S_1 \subset \mathbb{R}^n$ and $S_2 \subset \mathbb{R}^n$, the Minkwoski (vector) sum is defined by $S_1 \oplus S_2 \triangleq \{s_1 + s_2 | s_1 \in S_1, s_2 \in S_2\}$, the Pontryagin (geometric) set difference is $S_1 \ominus S_2 \triangleq \{s | s \oplus S_2 \subseteq S_1\}$. For a constant $\alpha \in [0, 1[$ and a set $S \subset \mathbb{R}^n$, we denote by αS the set $\{\alpha s | s \in S\}$. A closed hyperball in $\mathbb{R}^n$ of radius $r \in \mathbb{R}_{\geq 0}$ is denoted by $\mathcal{B}(r) \triangleq \{x \in \mathbb{R}^n | \|x\| \leq r\}$. For two points $w_1, w_2 \in \mathcal{W}$, we denote by $\mathcal{L}(w_1, w_2)$ the set of points that lie on the line connecting w_1 and w_2, i.e., $\mathcal{L}(w_1, w_2) = \{w | w = sw_1 + (1-s)w_2, \ 0 \leq s \leq 1\}$. We formulate the Denial-of-Service (DoS) resilient motion planning problem as follows.

A. Robot, Environment, and Threat Models

We assume that the workspace $\mathcal{W}$ contains a set of N_J adversarial communication-jamming radars. As shown in Fig. 1, each radar has an effective jamming radius causing a DoS for any communication passing through it. We denote by $J_k = \{w \in \mathcal{W} | w \in \{j_k\} \oplus \mathcal{B}_{r_k}\}$ the subset of the workspace affected by the kth jamming radar where $j_k \in \mathcal{W}$ is the position of the jamming radar and $r_k \in \mathbb{R}_{\geq 0}$ is its jamming radius. We suppose that the location j_k and radius r_k of each jamming radar are known, and leave the case of uncertain jamming radar position and radius for future work.

We then consider a team of N_R mobile robots operating in this adversarial environment. The mobile robots obey the following motion models:

$$x^i_{t+1} = f(x^i_t, u^i_t) + \theta^i_t, \qquad x^i_0 = \overline{x}^i_0 \tag{II.1}$$

where $x^i_t \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state of the ith robot at time $t \in \mathbb{N}$, $u^i_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the ith robot input at time t, selected from the space of admissible controls $\mathcal{U}$, $\overline{x}^i_0$ is the ith robot initial state, and $\theta^i_t \in \Theta \subseteq \mathbb{R}^n$ is the bounded disturbance on the ith robot at time t.

Each robot in the team needs to establish communication, at all times, with one or more base stations in a set of N_B stationary base stations (e.g., to receive mission updates). We denote by $B_i \in \mathcal{W}$ the location of the ith base station. The communication between base stations and robots can take place directly (single hop) or indirectly (multi-hop) through other robots. Throughout this paper, we assume a line-of-sight communication model, in which two nodes (robots or base stations) can communicate whenever the straight line connecting them does not pass through a DoS region.

B. Temporal Logic Specification

In addition to maintaining communication with the base stations at all times, the team must perform a mission that is defined over a set of regions of interest. We assume that the regions of interest are polytopes and partition the workspace as $\mathcal{W} = \bigcup_1^r \mathcal{W}_i$, where $\{\mathcal{W}_1, \ldots, \mathcal{W}_r\}$ is a set of non-overlapping regions. For robot R_i, we can associate to each of the above regions a Boolean proposition in the set $\Pi^i = \{\pi^i_1, \ldots, \pi^i_r\}$, where π^i_j evaluates to one (true) if robot R_i is in region $\mathcal{W}_j$ and zero (false) otherwise. We then denote by $h_{\mathcal{W} \to \Pi^i} : \mathcal{W} \to \Pi^i$ the map from each point $w \in \mathcal{W}$ to the proposition $\pi^i_j \in \Pi^i$ that evaluates to one at w for robot R_i. Moreover, a subset of the state variables of each robot, describing its position (coordinates), is also used to describe $\mathcal{W}$. Therefore, we denote as $h_{\mathcal{X} \to \mathcal{W}} : \mathcal{X} \to \mathcal{W}$ the natural projection of the state x^i onto the workspace $\mathcal{W}$, and by $h_{\mathcal{X} \to \Pi^i}$ the map from the state space of robot R_i to the set of propositions Π^i, obtained after projecting the state onto the workspace, i.e., $h_{\mathcal{X} \to \Pi^i}(x^i) = h_{\mathcal{W} \to \Pi^i}(h_{\mathcal{X} \to \mathcal{W}}(x^i))$.

We express the specification for a multi-robot mission using linear temporal logic (LTL) [12]. LTL formulas can compactly describe temporal orderings of events along the robots' trajectories and express a rich set specifications (e.g., safety, progress, response, surveillance, and monitoring) to capture complex tasks [18], [19], [20], [21], [22], [23]. Let $\Pi = \bigcup_{i=1}^R \Pi^i$ be the set of propositions associated with the workspace regions for all robots, as defined above. We consider formulas over a set of atomic propositions Σ, where $\sigma(\pi) \in \Sigma$ is a Boolean or pseudo-Boolean predicate over Π. From atomic propositions in Σ, any LTL formula can be generated according to the following grammar:

$$\psi := \sigma \mid \neg\psi_0 \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \psi_1 \ \mathcal{U} \ \psi_2 \mid \psi_1 \ \mathcal{R} \ \psi_2,$$

where ψ_0, ψ_1, ψ_2 are LTL formulas. Based on the above grammar, we can define $false$ and $true$ such that $false = \psi \wedge \neg\psi$ and $true = \neg false$. From the temporal operators *until* ($\mathcal{U}$), and *release* ($\mathcal{R}$), we can derive additional temporal operators, for example, *eventually* ($\Diamond$) and *always* ($\Box$), i.e., $\Diamond\psi = true\ \mathcal{U}\ \psi$, and $\Box\psi = false\ \mathcal{R}\ \psi$. We refer the reader to the literature (e.g., [24]) for the formal semantics of LTL. Defining the atomic propositions as Boolean or pseudo-Boolean predicates over Π allows us to express complex multi-robot behaviors like "either robot R_1 or R_2 must be in $\mathcal{W}_1$," via the proposition $\sigma_1 := \pi_1^1 \vee \pi_1^2$, or "at least one robot must be in $\mathcal{W}_2$" using the proposition $\sigma_2 := \sum_{i=1}^{N_R} \pi_2^i \geq 1$.

C. DoS-Resilient Motion Planning Problem

Despite the power of LTL in capturing complex missions, traditional formulations of LTL motion planning do not account for disturbances on the robot trajectories, which tends to be impractical, especially in the presence of adversarial environments, since disturbances can force some of the robots (or the communication links) to enter the DoS regions, leading to a mission failure. In this paper, we generalize the classical formulations to account for the uncertainty stemming from disturbances as follows.

Definition II.1 (Trajectory)**.** *A system trajectory is a tuple including the following infinite sequences:*

- $X = X_0X_1X_2\ldots$ *is a sequence of sets of system states where* $X_t = (X_t^0, \ldots, X_t^{N_R})$ *includes all possible states of all the robots at time* t*,*
- $\mu = \mu_0\mu_1\mu_2\ldots$ *is a control policy, where* $\mu_t^i : \mathbb{R}^n \to \mathbb{R}^m$ *is the control law for the* i*th robot at time* t *and* $\mu_t = (\mu_t^0, \ldots, \mu_t^{N_R})$ *is the set of control laws for all the robots at time* t*,*
- $\Lambda = \Lambda_0\Lambda_1\Lambda_2\ldots$ *is a sequence of sets of valuations over* Π*, where* $\Lambda_t = \{\lambda_t^i | \lambda_t^i = h_{\mathcal{X}\to\Pi^i}(x_t^i), x_t^i \in X_t^i, 1 \leq i \leq N_R\}$ *is the set of all possible valuations for all the possible states of all the robots at time* t*,*
- $\xi = \{\xi_0\xi_1\xi_2\ldots | \xi_t \in \Xi_t\}$ *is a set of sequences of valuations over* Σ *where* $\Xi_t = \{\sigma_i(\lambda_t) | \lambda_t \in \Lambda_t, \sigma_i \in \Sigma\}$ *represents the truth assignments of all the Boolean and pseudo-Boolean predicates associated with the state set* X_t *and propositions* Λ_t*.*

Robot j is considered in the ith robot's communication neighborhood at time t if there exists a line that connects i and j and does not pass through any jamming area J_k, with $1 \leq k \leq N_J$. However, due to disturbances, the robot states are no longer uniquely defined at each time. Instead, they can take any value within the sets X_t^i and X_t^j. Therefore, we define the set of DoS-free communication neighborhoods as follows.

Definition II.2 (DoS-Free Multi-hop Communication Neighborhood)**.** *Given a system trajectory, the DoS-free,* h*-hop, communication neighborhood* $C_{r,t}^i(h)$ *of the* i*th robot at time* t *can be recursively defined as:*

$$C_{r,t}^i(1) = \{j | j \in \{1,\ldots,N_R\}, \mathcal{L}(h_{\mathcal{X}\to\mathcal{W}}(x_t^i), h_{\mathcal{X}\to\mathcal{W}}(x_t^j)) \cap J_k \neq \emptyset \ \forall\ k \in \{1,\ldots,N_J\}, \forall\ x_t^i \in X_t^i, \forall\ x_t^j \in X_t^j\},$$

$$C_{r,t}^i(h) = \{j | j \in C_{r,t}^k(1), \forall\ k \in C_{r,t}^i(h-1)\}, h > 1.$$

Definition II.3 (DoS-Free Base Station Communication Neighborhood)**.** *Given a system trajectory, the set of base stations* $C_{b,t}^i(h)$ *for which robot* i *can establish a DoS-free,* h*-hop communication at time* t *can be recursively defined as:*

$$C_{b,t}^i(1) = \{j | j \in \{1,\ldots,N_B\}, \mathcal{L}(h_{\mathcal{X}\to\mathcal{W}}(x_t^i), B_j) \cap J_k \neq \emptyset \ \forall\ k \in \{1\ldots,N_J\}, \forall\ x_t^i \in X_t^i\}$$

$$C_{b,t}^i(h) = \{j | j \in C_{b,t}^k(1), \forall\ k \in C_{r,t}^i(h-1)\}, h > 1$$

Definitions II.2 and II.3 require a communication link to be established between two robots (or a robot and the base stations) regardless of the disturbance. Moreover, because of the disturbance, there may not exist a single valuation over the atomic propositions in Σ at each time in a trajectory. We therefore require that all the possible valuations in the trajectory satisfy the LTL specification as follows.

Problem II.4 (Centralized DoS-Resilient Motion Planning)**.** *Given a set of* N_R *robots whose individual dynamics are governed by* (II.1)*, a set of* N_B *stationary base stations, a mission specification captured by the LTL formula* ψ*, synthesize a system trajectory that satisfies the following constraints:*

Initial state constraint*:* $x_0^i = \overline{x}_0^i, \quad \forall\ i \in \{1,\ldots,N_R\}$,
State constraints*:* $X_t^i \subseteq \mathcal{X}, \quad \forall\ t \in \mathbb{N}, \forall\ i \in \{1,\ldots,N_R\}$,
Input constraints*:* $\mu_t^i(x_t^i) \in \mathcal{U}, \quad \forall\ x_t^i \in X_t^i, \forall\ t \in \mathbb{N}, \forall\ i \in \{1,\ldots,N_R\}$,
Dynamics constraints*:* $f(x_t^i, \mu_t^i(x_t^i)) \oplus \Theta \subseteq X_{t+1}^i$, $\forall\ x_t^i \in X_t^i, \forall\ t \in \mathbb{N}, \forall\ i \in \{1,\ldots,N_R\}$,
LTL constraints*:* $(\xi', 0) \models \psi \quad \forall\ \xi' \in \xi$,
Collision avoidance constraints*:* $\forall\ t \in \mathbb{N}, \forall\ i,j \in \{1,\ldots,N_R\}, i \neq j,\ \|h_{\mathcal{X}\to\mathcal{W}}(x_t^i) - h_{\mathcal{X}\to\mathcal{W}}(x_t^j)\| \geq \epsilon$*, with* $\epsilon \in \mathbb{R}_{>0}$, $\forall\ x_t^i \in X_t^i, \forall\ x_t^j \in X_t^j$,
DoS resilience constraints: $\cup_{h=1}^{N_R} C_{b,t}^i(h) \neq \emptyset, \quad \forall\ t \in \mathbb{N}, \forall\ i \in \{1,\ldots,N_R\}$.

III. Satisfiability Modulo Convex Programming (SMC)-based Motion Planning

We resort to the Satisfiability Modulo Convex Programming (SMC) framework [14], [16], [17] to devise a motion planning algorithm that solves Problem II.4. SMC-based motion planning is an iterative method that relies on encoding the planning problem, for a fixed horizon L, as a monotone SMC formula φ over Boolean and convex constraints, respectively capturing the mission constraints and the robot physical constraints.

As shown in Alg. 1, our motion planner consists of three steps. First, as an offline step, we generate a perturbation of the workspace by inflating the DoS jamming areas to account for the worst-case disturbance scenario. Details on how to compute the perturbed workspace and the associated guarantees are given in Sec. IV. Next, we translate both the LTL mission specification and the DoS-resilience constraints into a conjunction of Boolean constraints over the workspace propositions. Details on the generation of these constraints are provided in Sec. V.

Algorithm 1 SMC-BASED MOTION PLANNER

Input: $\mathcal{P} := \langle \mathcal{W}, J, B, \Pi, \Sigma, f, \Theta, \overline{x}_0, \mathcal{X}, \mathcal{U}, \epsilon, \psi \rangle$
Input: Disturbance rejection factor β

Step 1: Compute the tube set and the workspace perturbation
$(\Omega, \mu_\Omega) :=$ COMPUTE-RCI$(f, \Theta, \mathcal{X}, \beta\mathcal{U})$
$(J^*, \mathcal{W}^*_{\psi,-}, \mathcal{W}^*_{\psi,+}) :=$ PERTURB$(J, \Omega, \mathcal{W})$
$(\mathcal{W}^*, Adj_p, Adj_c) :=$PARTITION$(\mathcal{W}, J^*, \mathcal{W}^*_{\psi,-}, \mathcal{W}^*_{\psi,+})$

Step 2: Use SMC to plan the nominal trajectory
Initialize horizon: $L := 1$;
while Trajectory is not found **do**
  $|[\mathcal{P}, L]|_D :=$ ENCODE-DIS-PLAN($\mathcal{W}^*$,B, Π, Σ, Adj_p, Adj_c, ψ, L)
  $|[\mathcal{P}, L]|_C :=$ ENCODE-CON-PLAN($\mathcal{W}^*$, f, $\overline{x}_0$, $\mathcal{X}$, $(1-\beta)\mathcal{U}$, ϵ, L)
  (STATUS, z, v) := SMC.SOLVE($|[\mathcal{P}, L]|_D, |[\mathcal{P}, L]|_C$);
  if STATUS == UNSAT **then**
    Increase horizon: $L := L + 1$;

Step 3: Trajectory Tracking
At each time step, apply the input $u_t = v_t + \mu_\Omega(x_t - z_t)$

To solve these constraints, SMC uses an efficient Boolean satisfiability (SAT) solver to find a candidate sequence of workspace regions that satisfies the mission and DoS constraints while ignoring the robot dynamics, input, and state constraints. A convex solver is then used to check the feasibility of the candidate path. If both the Boolean and the convex constraints are satisfied, a valid trajectory is returned, consisting of the proposed plan and the corresponding *nominal* state and control input trajectories. Otherwise, the proposed high-level sequence is marked as infeasible and new candidate plans are generated until either a feasible one is found, or no trajectory is feasible for the current horizon length L. A prominent feature of SMC is the generation of compact infeasibility certificates, i.e., "succinct explanations" that can capture the root causes for the infeasibility of a plan and rule out the largest possible number of invalid plans for the SAT solver to accelerate the search. This iterative procedure showed to be more than two orders of magnitude faster than state-of-the-art sampling based techniques for high-dimensional state spaces [16].

Finally, we compute a feedback control law that can track the *nominal* trajectory generated using the SMC approach and the perturbed workspace. This control law will be used to address disturbances during system operation. Details on the computation of the feedback law are provided in Sec. VI.

IV. ROBUST CONTROLLED INVARIANT SETS AND WORKSPACE PERTURBATION

Given the robot dynamics (II.1), a feedback controller that rejects the disturbance θ and forces the trajectories governed by (II.1) to evolve inside the state constraint set $\mathcal{X}$ can be characterized by the notion of robust controlled invariant set contained inside $\mathcal{X}$ [25]. A set $\Omega \subseteq \mathcal{X}$ is a *robust controlled invariant (RCI) set* for the system (II.1) if there exists a feedback controller $\mu_\Omega : \Omega \to \mathcal{U}$ such that, for every $x_t \in \Omega$, the following holds:

$$f(x_t, \mu_\Omega(x_t)) + \theta_t \in \Omega, \qquad \forall\ \theta_t \in \Theta, \forall\ t \in \mathbb{N}.$$

In other words, if the system state starts in Ω, then it will stay in Ω in spite of the disturbance. Moreover, when f is piecewise affine, we can effectively separate the goals of disturbance rejection and trajectory planning [26], [27], [28].

Given a design parameter $\beta \in [0, 1[$, a disturbance-free state trajectory $z_0, z_1, \ldots$, and an open-loop control trajectory $v_0, v_1, \ldots$ such that, for all t, $z_{t+1} = f(z_t, v_t)$ and $v_t \in (1-\beta)\mathcal{U}$, we can find a robust controlled invariant set Ω_β and a corresponding feedback law μ_{Ω_β} that ensure $\mu_{\Omega_\beta}(x_t) \in \beta\mathcal{U}$ and $x_t \in z_t \oplus \Omega_\beta$ for all t. In other words, the RCI set Ω can be regarded as a "tube," regulated by μ_{Ω_β}, around a nominal (disturbance-free) trajectory $z_0, z_1, \ldots$, determined by $v_0, v_1, \ldots$. In what follows, we will restrict our attention to piecewise affine robot dynamics for which algorithms that synthesize polytopic RCI sets are already available in the literature [29], [13], [26]. For simplicity, we also drop the subscript β from the RCI notation.

In our case, rejecting disturbances translates into designing a tube that lies entirely in the jamming-free region of the workspace. We call such a tube an *Ω-perturbation (inflation)* of the nominal trajectory. We then observe that computing an Ω-perturbed trajectory that lies in the jamming-free space can be rather translated into the problem of computing a nominal (ideal) trajectory that lies in a modified space in which the jamming area and the workspace regions are, instead, perturbed. To derive this perturbation of the space, we proceed as follows.

Given the LTL formula ψ, we denote by $\mathcal{W}_{\psi,+}$ the set of (jamming-free) workspace regions whose corresponding atomic propositions appear asserted (without negation) in ψ, and by $\mathcal{W}_{\psi,-}$ the set of workspace regions whose corresponding atomic propositions are negated in ψ. We assume that a region can be either asserted or negated in ψ, and therefore $\mathcal{W}_{\psi,+}$ and $\mathcal{W}_{\psi,-}$ are disjoint sets. We then "inflate" by Ω the jamming areas and the workspace regions $\mathcal{W}^-_\psi$ that need to be avoided, and "shrink" by Ω the workspace regions $\mathcal{W}^+_\psi$ which must be traversed. Formally, we obtain:

$$J^* = \{J_k \oplus \Omega \mid k \in \{1, \ldots, N_J\}\}$$
$$\mathcal{W}^*_{\psi,-} = \{\mathcal{W}' \oplus \Omega \mid \mathcal{W}' \in \mathcal{W}_{\psi,-}\}$$
$$\mathcal{W}^*_{\psi,+} = \{\mathcal{W}' \ominus \Omega \mid \mathcal{W}' \in \mathcal{W}_{\psi,+}\}.$$

V. SYNTHESIS OF DOS-FREE NOMINAL TRAJECTORIES

As pictorially shown in Fig. 1, we start by over-approximating the Ω-perturbed jamming areas J^*_k using a set of polyhedra, which originates a coarse, multi-resolution, discretization of the free space. Unlike grid-based methods, where the workspace is discretized using a grid (or mesh) of (small) uniform resolution, the coarse-grained abstraction used in this paper avoids state explosion. This decomposition procedure is similar to the ones previously proposed for triangular [30] or polygonal [31] representations. We denote by $\mathcal{W}^*_1, \mathcal{W}^*_2, \ldots \mathcal{W}^*_{r^*}$ the set of regions obtained after discretization, r^* being the total number of regions.

Based on this partition, we compute two adjacency functions, denoted by Adj_p and Adj_c that correspond, respectively, to the physical adjacency and communication adjacency relations between the regions. In particular, two

regions $\mathcal{W}_i^*$ and $\mathcal{W}_j^*$ are said to be physically adjacent, written $Adj_p(\mathcal{W}_i^*, \mathcal{W}_j^*) = 1$, if the polyhedra $\mathcal{W}_i^*$ and $\mathcal{W}_j^*$ share one facet; otherwise, we write $Adj_p(\mathcal{W}_i^*, \mathcal{W}_j^*) = 0$. Similarly, $\mathcal{W}_i^*$ and $\mathcal{W}_j^*$ are communication adjacent if we can connect any point of $\mathcal{W}_i^*$ with any point of $\mathcal{W}_j^*$ without passing through a jamming area, that is,

$$Adj_c(\mathcal{W}_i^*, \mathcal{W}_j^*) = \begin{cases} 1 & \text{if } \mathcal{L}(w_i, w_j) \notin J_k^* \\ & \quad \forall\ k \in \{1, \dots, N_J\}, \\ & \quad \forall\ (w_i, w_j) \in \mathcal{W}_i^* \times \mathcal{W}_j^* \\ 0 & \text{otherwise} \end{cases}$$

We use these notions of adjacency to encode the mission and DoS-resilience constraints as follows.

A. Encoding Mission and DoS-Resilience Constraints

For each robot, region, and time, we introduce a Boolean variable $\pi_{j,t}^i$ which evaluates to one if and only if robot i is in region $\mathcal{W}_j^*$ at time t. Similarly, for each base station and region, we introduce a Boolean variable κ_j^i which evaluates to one if and only if base station i is in region $\mathcal{W}_j^*$, since base stations are stationary and their locations do not change with time. We use these decision variables along with the physical adjacency function Adj_p to translate the high-level, discrete planning constraints into a conjunction of Boolean constraints using the Bounded Model Checking (BMC) encoding technique for LTL model checking. We refer to the literature [24] for details on the Boolean encoding of LTL specifications. In the remainder of this section, we report the encoding of the communication constraints.

We introduce a set of Boolean variables of the form $r_t(i, j, h)$, each evaluating to one whenever robot i can establish an h-hop DoS-free communication with robot j, and zero otherwise. Similarly, a Boolean variable $b_t(i, j, h)$ evaluates to one whenever robot i can establish an h-hop DoS-free communication with base station j. We then capture the communication constraints as follows.

Adjacency Constraints. We encode single-hop communication adjacency as the conjunction of the following constraints:

$$\forall t \in \{0, \dots, L\}, \forall i, j \in \{1, \dots, N_R\}:$$
$$r_t(i, j, 1) \leftrightarrow \bigvee_{k=1}^{r^*} \left(\pi_{k,t}^i \wedge \left(\bigvee_{k' \in \mathcal{N}_c(i)} \pi_{k',t}^j \right) \right), \quad \text{(V.1)}$$

where $\mathcal{N}_c(i) = \{j \in \{1, \dots r^*\} | Adj_c(\mathcal{W}_i^*, \mathcal{W}_j^*) = 1\}$ is the set of indexes marking the regions that are communication adjacent (neighbors) to region $\mathcal{W}_i^*$. Similarly, for the base stations, we obtain

$$\forall t \in \{0, \dots, L\}, \forall i \in \{1, \dots, N_R\}, \forall j \in \{1, \dots, N_B\}:$$
$$b_t(i, j, 1) \leftrightarrow \bigvee_{k=1}^{r^*} \left(\kappa_k^i \wedge \left(\bigvee_{k' \in \mathcal{N}_c(i)} \pi_{k',t}^j \right) \right). \quad \text{(V.2)}$$

Transitivity Constraints. To encode multi-hop communication, we generate the following constraints:

$$\forall t \in \{0, \dots, L\}, \forall i, j \in \{1, \dots, N_R\}, \forall h \in \{1, \dots, N_R\}:$$
$$\bigvee_{k=1}^{N_R} \bigvee_{\substack{h_1, h_2 \in \{1, \dots, N_R\} \\ h_1 + h_2 = h}} (r_t(i, k, h_1) \wedge r_t(k, j, h_2)) \leftrightarrow r_t(i, j, h)$$

and conjoin them with the following ones:

$$\forall t \in \{0, \dots, L\}, \forall i \in \{1, \dots, N_R\}, \forall j \in \{1, \dots, N_B\},$$
$$\forall h \in \{1, \dots, N_R\}:$$
$$\bigvee_{k=1}^{N_R} \bigvee_{\substack{h_1, h_2 \in \{1, \dots, N_R\} \\ h_1 + h_2 = h}} (r_t(i, k, h_1) \wedge b_t(k, j, h_2)) \leftrightarrow b_t(i, j, h).$$

DoS-Resilience Constraints. Finally, the constraints below ensure that each robot is connected with at least one base station, by either a single or multi-hop communication link:

$$\forall i \in \{1, \dots, N_R\}: \qquad \bigwedge_{t=0}^{L} \bigvee_{j=1}^{N_B} \bigvee_{h=1}^{N_R} b_t(i, j, h). \quad \text{(V.3)}$$

B. Nominal Trajectory Planning

As discussed in Sec. III, we use a SAT solver to find a high-level, candidate sequence of regions that satisfy the Boolean formula encoding the LTL specification and the DoS-resilience constraints. It is possible to represent this trajectory, which is infinite in general, with a finite sequence of the form $\rho = (\rho_0 \rho_1 \dots \rho_{k-1})(\rho_k \dots \rho_L)^\omega$, consisting of a prefix $\rho_0 \rho_1 \dots \rho_{k-1}$ and a loop sequence $\rho_k \dots \rho_L$ that repeats indefinitely, as denoted by the superscript ω [24].

Given the system piecewise affine dynamics f, the state and control constraint sets $\mathcal{X}$ and $(1 - \beta)\mathcal{U}$, the robot initial state $\overline{x}_0$, the high-level candidate path ρ, the margin ϵ for collision avoidance, and the Ω-perturbed workspace regions associated with ρ, checking the feasibility of the candidate path, generating the nominal state trajectory $z_0^i, z_1^i, \dots$ for each robot, or providing succinct infeasibility certificates, whenever such state trajectories do not exist, can all be cast as convex programs [14], [16].

VI. Tracking of the Nominal Trajectory

The final step is to compute the control law for tracking the nominal trajectory $z_0, z_1, \dots$ by summing the nominal open-loop control input v_t and the feedback control law $\mu_\Omega(x_t - z_t)$ for all $t \in \mathbb{N}$. Algorithm 1 summarizes the proposed SMC-based robust motion planning procedure. Its correctness guarantees are stated below.

Theorem VI.1 (Correctness of Algorithm 1)**.** *Algorithm 1 is sound, that is, all trajectories resulting from its execution are solutions of Problem II.4.*

Proof Sketch. Soundness of Alg. 1 directly follows from the separation between disturbance rejection and trajectory planning (see, e.g., [26, Theorem 5.4]), the soundness of the SMC-based motion planning algorithm (for the nominal trajectory) [16, Theorem 4.2], the construction of the perturbed sets $J^*, \mathcal{W}_{\psi,-}^*, \mathcal{W}_{\psi,+}^*$, and the soundness of the DoS-resilience encoding in (V.3), i.e., the fact that, if (V.3) holds, then there exists a communication path between each robot and at least one of the base stations at each time. ∎

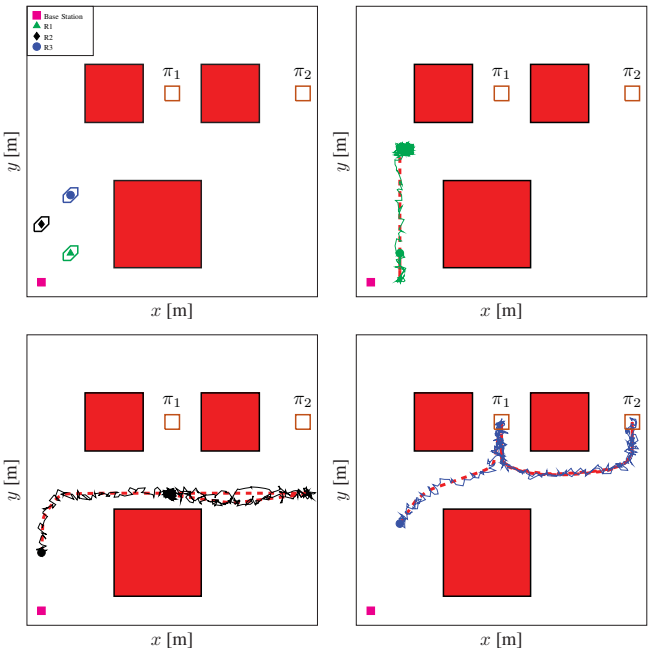

Fig. 2. Workspace showing the initial position of the robots, the base stations, and the jamming areas (red boxes) along with the three trajectories subject to $\big(\Box\Diamond(\pi_1^3 = 1)\big) \wedge \big(\Box\Diamond(\pi_2^1 + \pi_2^2 + \pi_2^3 = 1)\big)$. Actual trajectories (green for $R1$, black for $R2$, and blue for $R3$) are plotted on top of the nominal trajectories (dashed red).

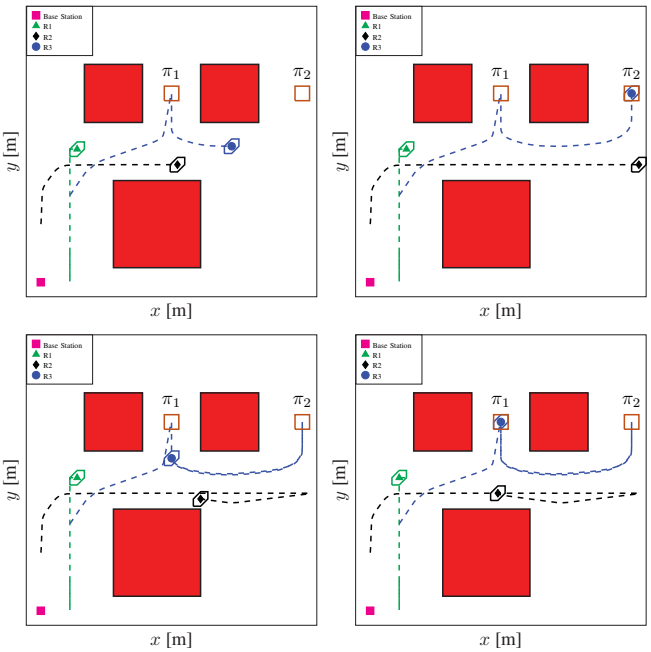

Fig. 3. Snapshots of the three robots at different times along with the nominal trajectories and the corresponding RCI sets, subject to $\big(\Box\Diamond(\pi_1^3 = 1)\big) \wedge \big(\Box\Diamond(\pi_2^1 + \pi_2^2 + \pi_2^3 = 1)\big)$.

VII. Results

We implemented Alg. 1 in PYTHON on top of the SATEX solver [15], using Z3 [32] as a SAT solver and CPLEX [33] as a convex optimization solver. All the experiments were executed on an Intel Core i7 2.3-GHz processor with 16 GB of memory.

To illustrate the capabilities of our algorithm in a multi-robot scenario under generic LTL specifications, we consider a team of 3 robots, $R1$, $R2$, $R3$, and one base station operating in the workspace represented in Fig. 2 (top left). We assume robot dynamics captured by chains of integrators, one chain for each coordinate of the workspace, and a sampling time of 0.5 s. The upper bound on the disturbance is 0.2 m on the robot position (coordinates) and zero on the higher-order states. Red boxes denote the DoS areas of the three jamming radars. Initial positions are shown in Fig. 2 (top left). The mission is specified by the LTL formula $\psi := \big(\Box\Diamond(\pi_1^3 = 1)\big) \wedge \big(\Box\Diamond(\pi_2^1 + \pi_2^2 + \pi_2^3 = 1)\big)$ which requires that $R3$ visit region π_1 infinitely often, and that any of the robots visit location π_2 infinitely often.

Figure 2 shows the nominal trajectories $z_0^i z_1^i z_2^i \ldots$, $i \in \{1, 2, 3\}$, for the double integrator case (dashed red lines) along with the actual robots' trajectories $x_0^i x_1^i x_2^i \ldots$ (green for $R1$, black for $R2$, and blue for $R3$) for a realization of the disturbance from a random uniform distribution over the set of admissible disturbances. Figure 3 reports snapshots of the three robots at different times along with the nominal trajectories and the corresponding RCI sets. The planner strategically positions $R1$ to guarantee communication with the base station at all times. As $R3$ approaches the first goal, $R2$ is positioned to operate as an intermediate hub between $R3$ and $R1$, thus creating a 2-hop link between the base station and $R3$. Similarly, when $R3$ reaches the second goal, $R2$ is also moved to provide the necessary communication path for $R3$. The overall computation of the RCI set, the nominal trajectory, and the feedback law took around 3 s, 534 s, and 20 ms, respectively.

TABLE I

EXECUTION TIME FOR THE WORKSPACE IN FIG. 2.

# robots	# states	RCI [s]	One Base Station			Two Base Stations		
			# Bool variables	SMC [s]	μ_Ω [ms]	# Bool variables	SMC [s]	μ_Ω [ms]
2	4	2.878	36	92.99	10.2	108	175.64	11.5
	6	3.265	42	223.04	13.4	126	199.81	13.6
	8	3.780	42	88.98	18.5	126	1175.96	18.8
3	4	2.878	108	210.06	10.2	240	411.73	11.5
	6	3.265	126	347.89	13.4	280	474.17	13.6
	8	3.780	126	818.69	18.5	280	1328.92	18.8
4	4	2.878	240	565.31	10.2	450	647.16	11.5
	6	3.265	280	645.01	13.4	525	2685.97	13.6
	8	3.780	280	1597.51	18.5	525	2373.67	18.8

Table I reports the execution time of the three steps in Alg. 1 for a basic reach-avoid specification as the number of robots and the number of integrators (per robot) in the chain, hence the number of state variables, increase in the presence of one and two base stations. The table also reports the number of Boolean variables needed to encode the DoS-resilience constraints.

VIII. Conclusions

Our numerical results show the capability of the proposed satisfiability modulo convex programming (SMC)-based algorithm of guaranteeing robust plans that are resilient to denial-of-service attacks, while supporting the specification and execution of complex multi-robot missions expressed in linear temporal logic. Future work includes the extension of our framework to support networking models that can also account for the communication quality.

REFERENCES

[1] F. Higgins, A. Tomlinson, and K. M. Martin, "Threats to the swarm: Security considerations for swarm robotics," *International Journal on Advances in Security*, vol. 2, no. 2 & 3, 2009.

[2] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, pp. 1–39, 2016.

[3] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[4] M. J. Mears, "Cooperative electronic attack using unmanned air vehicles," in *Proc. American Control Conference*. IEEE, 2005, pp. 3339–3347.

[5] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI Communications*, no. Preprint, pp. 1–12.

[6] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus, "Guaranteeing spoof-resilient multi-robot networks," *Autonomous Robots*, vol. 41, no. 6, pp. 1383–1400, 2017.

[7] V. Renganathan and T. Summers, "Spoof resilient coordination for distributed multi-robot systems," in *Int. Symp. Multi-Robot and Multi-Agent Systems*. IEEE, 2017, pp. 135–141.

[8] Y. Xu, G. Ren, J. Chen, Y. Luo, L. Jia, X. Liu, Y. Yang, and Y. Xu, "A One-Leader Multi-Follower Bayesian-Stackelberg Game for Anti-Jamming Transmission in UAV Communication Networks," *IEEE Access*, vol. 6, pp. 21 697–21 709, 2018.

[9] X. Lu, D. Xu, L. Xiao, L. Wang, and W. Zhuang, "Anti-jamming communication game for UAV-aided VANETs," in *IEEE Global Communications Conf.*, Dec 2017, pp. 1–6.

[10] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[11] Q. Qingwen, D. Wenfeng, L. Meiqing, and Y. Yang, "Cooperative jamming resource allocation of UAV swarm based on multi-objective DPSO," in *Proc. Chinese Control and Decision Conference*, June 2018, pp. 5319–5325.

[12] A. Pnueli, "The temporal logic of programs," in *FOCS*, 1977, pp. 46–57.

[13] S. V. Rakovic and M. Baric, "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1599–1614, 2010.

[14] Y. Shoukry, P. Nuzzo, I. Saha, A. Sangiovanni-Vincentelli, S. Seshia, G. Pappas, and P. Tabuada, "Scalable lazy SMT-based motion planning," in *Proc. IEEE Conf. Decision and Control*, 2016, pp. 6683–6688.

[15] Y. Shoukry, P. Nuzzo, A. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "SMC: Satisfiability modulo convex optimization," in *Proc. Int. Conf. Hybrid Systems: Computation and Control*, Apr. 2017.

[16] Y. Shoukry, P. Nuzzo, A. Balkan, I. Saha, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "Linear temporal logic motion planning for teams of underactuated robots using satisfiability modulo convex programming," in *Proc. IEEE Conf. Decision and Control*, 2017, pp. 1132–1137.

[17] Y. Shoukry, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "SMC: Satisfiability modulo convex programming," vol. 106, no. 9, pp. 1655–1679, Sep. 2018.

[18] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Trans. Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.

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

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

[21] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Trans. Automatic Control*, vol. 57, no. 11, pp. 2817–2830, 2012.

[22] P. Nuzzo, H. Xu, N. Ozay, J. Finn, A. Sangiovanni-Vincentelli, R. Murray, A. Donze, and S. Seshia, "A contract-based methodology for aircraft electric power system design," *IEEE Access*, vol. 2, pp. 1–25, 2014.

[23] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe LTL specifications," in *Int. Conf. Intelligent Robots and Systems*, 2014, pp. 1525–1532.

[24] A. Biere, K. Heljanko, T. Junttila, T. latvala, and V. Schuppan, "Linear encoding of bounded LTL model checking," *Logical Methods in Computer Science*, vol. 2, no. 5:5, pp. 1–64, 2006.

[25] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 604–613, 1972.

[26] S. Sadraddini and C. Belta, "Formal guarantees in data-driven model identification and control synthesis," in *Proc. Int. Conf. Hybrid Systems: Computation and Control*. ACM, 2018, pp. 147–156.

[27] B. Schürmann and M. Althoff, "Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems," in *Proc. American Control Conference*. IEEE, 2017, pp. 2522–2529.

[28] C. Fan, U. Mathur, S. Mitra, and M. Viswanathan, "Controller synthesis made real: reach-avoid specifications and linear dynamics," in *International Conference on Computer Aided Verification*. Springer, 2018, pp. 347–366.

[29] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3665–3670, 2017.

[30] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 61–70, 2007.

[31] X. C. Ding, M. Kloetzer, Y. Chen, and C. Belta, "Automatic deployment of robotic teams," *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 75–86, 2011.

[32] L. De Moura and N. Björner, "Z3: An efficient SMT solver," in *Proc. Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems*, 2008, pp. 337–340.

[33] (2012, Feb.) IBM ILOG CPLEX Optimizer. [Online]. Available: www.ibm.com/software/integration/optimization/cplex-optimizer/