Graph-Based Controller Synthesis for Safety-Constrained, Resilient Systems

Matija Bucić, Melkior Ornik, and Ufuk Topcu

***Abstract*— Resilience to damage, component degradation, and adversarial action is a critical consideration in design of autonomous systems. In addition to designing strategies that seek to prevent such negative events, it is vital that an autonomous system remains able to achieve its control objective even if the system partially loses control authority. While loss of authority limits the system's control capabilities, it may be possible to use the remaining authority in such a way that the system's control objectives remain achievable. In this paper, we consider the problem of optimal design for an autonomous system with discrete-time linear dynamics where the available control actions depend on adversarial input produced as a result of loss of authority. The central question is how to partition the set of control inputs that the system can apply in such a way that the system state remains within a safe set regardless of an adversarial input limiting the available control inputs to a single partition elements. We interpret such a problem first as a variant of a safety game, and then as a problem of existence of an appropriate edge labeling on a graph. We obtain conditions for existence and a computationally efficient algorithm for determining a system design and a control policy that preserve system safety. We illustrate our results on two examples: a damaged autonomous vehicle and a method of communication over a channel that ensures a minimal running digital sum.**

I. Introduction

Controller's loss of authority over parts of an autonomous system may happen in many scenarios:

(a) *System damage and component degradation.* An autonomous system operating for substantial periods of time in a remote, unknown, or hostile environment will inevitably sustain damage or experience partial system failures over time due to malfunctions. Examples include unmanned aerial vehicles (UAVs) operating over contested territory [1], search-and-rescue robots [2], and rovers performing missions on extraterrestrial surfaces [3].

(b) *Hostile takeover.* In a number of adversarial settings, the adversary will attempt to take over elements of the system and disturb its regular functions. A typical setting is that of attacks on computer networks [4] and power systems [5], [6], where, because of the vastness of the network and heterogeneity and physical distance between system elements, an adversarial agent may be able to penetrate a part of the system. Hostile takeover scenarios also include recent successful attacks resulting in loss of control over UAVs; see, e.g., [7], [8].

(c) *User-responsive systems.* Settings where an automated controller is required to respond to (a priori unknown) user inputs in a particular way necessarily yield a part of the control authority to the user. Such scenarios include resource distribution in parallel computing [9], semantic web service composition [10], and communication protocols [11].

In all of the above settings, it is critical to ensure that the autonomous system can perform its tasks regardless of external control inputs. A standard method of ensuring continued functioning of the system is through imposing redundancy or near-redundancy in design. For instance, critical components in commercial airplanes are duplicated [12], and military UAVs use a combination of different sensing systems for navigation [8]. In the latter example, while these different sensing systems do not work in the same way and, in regular flight regime, serve to complement each other, each system is able to ensure that the UAV can achieve basic control objectives even if complementary systems are not functioning.

Motivated by the above scenarios, our work seeks to investigate how to guarantee continued safe operation of an abstract control system in which some components are no longer under the controller's authority. We focus on systems with linear driftless discrete-time dynamics, and interpret the partial loss of control authority as limitations on the controller's choice of actions, based on adversarial inputs. The control objective that we investigate is *safety*: the system state is required to remain within a particular set throughout the system run. We are interested in (i) designing a control policy, if one exists, that guarantees safety of a predetermined system regardless of the adversarial inputs and (ii) determining a *resilient system design* — the description of all control inputs that the system can apply for a particular adversarial input — which ensures that there will exist a policy satisfying (i).

The work in this paper is closely related to previous research on control of safety-critical systems [13], [14] and safety games [15]–[17]. In particular, as we will show, given a system design, i.e., possible control inputs given an adversarial input, a safe control policy can be interpreted as a winning strategy for a turn-based safety game. This interpretation leads to a computationally efficient algorithm for designing a safe control policy. However, such an algorithm does not directly provide for a computationally feasible procedure of determining whether there *exists* a resilient system design,

M. Bucić is with the Department of Mathematics, ETH Zürich. M. Ornik is with the Institute for Computational Engineering and Sciences, University of Texas at Austin. U. Topcu is with the Institute for Computational Engineering and Sciences and the Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin. Emails: {matija.bucic@math.ethz.ch, mornik@ices.utexas.edu, utopcu@utexas.edu}

This work was funded by grants FA8750-17-C-0087 from the Defense Advanced Research Projects Agency and 200021-175573 from the Swiss National Science Foundation.

as each design corresponds to a different safety game, and searching through all possible games is computationally prohibitive. We address this challenge through a method based on a graph-theoretical interpretation of system design.

The outline of the remainder of this paper is as follows. In Section II we provide a motivation for theoretical framework used in the paper, and formally describe the problems of safe control design and resilient system design under adversarial action. We then interpret such problems within the context of safety games in Section III, resulting in a simple solution to the problem of safe control design. We interpret the problem of resilient system design in a graph-theoretical setting in Section IV, and — using the probabilistic method, as described in [18] — provide a sufficient condition and a necessary condition for its solvability in Section V. Based on the previous section, we provide a computationally efficient algorithm for resilient system design and construction of a safe control policy in Section VI. Section VII illustrates our techniques on two examples: an autonomous vehicle experiencing partial loss of control authority, and design of codes for communication over a channel with a bounded running digital sum.

Notation. The symbol $\mathbb{N}$ denotes all strictly positive integers, $\mathbb{N}_0$ denotes all nonnegative integers, and $\mathbb{Z}$ denotes all integers. For $m \in \mathbb{N}$, $[m]$ denotes the set $\{1, \dots, m\}$. For a set $\mathcal{X}$, $|\mathcal{X}|$ denotes its cardinality, and $2^{\mathcal{X}}$ the set of all its subsets. For an event B within a particular probability distribution, $\Pr(B)$ denotes the probability of B occurring. For a graph $G = (V, E)$ and vertex $v \in V$, $\deg_G(v)$ denotes the (outgoing, if the graph is directed) degree of v, and $\mathrm{mindeg}(G)$ denotes the minimal (outgoing) degree of any vertex in V. If G, H are graphs, $G \subseteq H$ signifies that G is an induced subgraph of H. Vector e_i denotes the standard basis vector consisting solely of zeros, except for a 1 in the i-th position. Symbol $\|v\|_\infty$ denotes the max-norm of a vector $v \in \mathbb{R}^n$, and $\|v\|_1$ denotes the 1-norm of a vector v.

II. Problem Statement

Consider a system operating with discrete-time dynamics

$$x(t+1) = x(t) + u(t) \tag{1}$$

for all times $t \in \mathbb{N}_0$, where $x(0) \in \mathbb{Z}^n$ and $u \in \mathcal{U} \subseteq \mathbb{Z}^n$, with a finite $\mathcal{U}$. While model (1) is simple, our use of it is supported by its wide presence in robotic exploration (see, e.g., [19]–[21], and the references therein) as well as its use in communication over a channel [22]. As we will discuss in subsequent sections, (1) yields a straightforward graph-theoretical interpretation of system motion which may lead to generalizations for more complex models.

To provide motivation for the problems that we will pose, let us assume that dynamics (1) represent an autonomous system controlled by actuators A_1, A_2, …, A_p. The control effort $u(t)$ is then given as $u(a_1(t), \dots, a_p(t))$, where $a_i(t) \in \mathcal{A}_i$ is the setting of actuator A_i at time t, and $\mathcal{U} = \{u(a_1, \dots, a_p) \mid a_i \in \mathcal{A}_i, i = 1, \dots, p\}$.

We are interested in the scenario where the controller experiences loss of authority over some of the actuators, say $A_1, \dots, A_r$. Thus, the choice of $a_1(t), \dots, a_r(t)$ is not made by the controller, and any control actuation $u(t)$ needs to chosen in the set $U(a_1(t), \dots, a_r(t)) = \{u(a_1(t), \dots, a_r(t), a_{r+1}, \dots, a_p) \mid a_i \in \mathcal{A}_i, i \geq r+1\}$. We assume that we do not possess any prior knowledge about the inputs $a_1(t), \dots, a_r(t)$; these may be subjects to adversarial choices.

The control objective that we consider is *safety*. That is, we want to ensure that $x(t) \in S$ for all $t \geq 0$, where $S \subseteq \mathbb{Z}^n$ is a predetermined set with $x(0) \in S$. We are interested in two questions:

(i) For given sets $U(a_1, \dots, a_r)$, determine, if it exists, a control policy that guarantees system safety regardless of choices $a_1(t), \dots, a_r(t)$.
(ii) Design sets $U(a_1, \dots, a_r)$ so that the above control policy exists.

The latter question corresponds to designing the abilities and role of each actuator in such a way that the system is resilient to loss of authority over some of the actuators.

If the system can exhibit perfect redundancy, i.e., $U(a_1, \dots, a_r) = \mathcal{U}$ for every $a_1 \in \mathcal{A}_1$, …, $a_r \in \mathcal{A}_r$, questions (i) and (ii) are simple. However, redundancy is often undesirable due to cost, weight, or resource consumption [23]. Thus, we assume that it is impossible to execute exactly the same control with two different actuations. Under this assumption, $\{U(a_1, \dots, a_r) \mid a_1 \in \mathcal{A}_1, \dots, a_r \in \mathcal{A}_r\}$ is a partition of $\mathcal{U}$. For the sake of simpler notation, we denote $\mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_r = [m]$ for some $m \in \mathbb{N}$.

Questions (i) and (ii) are now formulated as follows.

Restricted partition control problem (RPCP): Let $S \subseteq \mathbb{Z}^n$ and $x(0) \in S$. Let $\mathcal{U} \subseteq \mathbb{Z}^n$ be finite, and $U : [m] \to 2^{\mathcal{U}}$ such that $\{U(1), \dots, U(m)\}$ is a partition of $\mathcal{U}$. Does there exist a function $\hat{u} : \cup_{i=1}^{\infty} [m]^i \to \mathcal{U}$ such that

(i) $\hat{u}(d_1, \dots, d_k) \in U(d_k)$ for all $d_1, \dots, d_k \in [m]$, and
(ii) for every $d : \mathbb{N}_0 \to [m]$, if $x(t)$ is the solution of (1) with $u(t) = \hat{u}(d(0), \dots, d(t))$, then $x(t) \in S$ for all $t \in \mathbb{N}_0$?

Free partition control problem (FPCP): Let $S \subseteq \mathbb{Z}^n$ and $x(0) \in S$. Let $\mathcal{U} \subseteq \mathbb{Z}^n$ be finite. Does there exist a partition $\{U(1), \dots, U(m)\}$ for which the RPCP admits a solution?

We note that in practice the available choices of partitions in the FPCP may be subject to constraints, e.g., physical limitations in design of actuators. We use the unconstrained version to provide an elegant illustration of a general approach to solving the above problems. Before moving towards solutions of the RPCP and the FPCP, let us introduce a running example.

Example 1 (Damaged vehicle): Consider an autonomous vehicle moving on $\mathbb{Z}^2$ according to dynamics (1). At every instance in time, the vehicle can perform one of five actions: go one position to the north, south, east or west, or remain in the same position. In other words, $\mathcal{U} = \{e_1, -e_1, e_2, -e_2, (0,0)\}$. The vehicle's initial position is given by $x(0) = (0,0)$, and the safe set S is given by $S = \{x \in \mathbb{Z}^2 \mid \|x\|_\infty \leq 1\}$. The setup is graphically illustrated in Fig. 1.

Let us first consider the RPCP with $m = 2$ and $U(1) = \{e_1, e_2\}$, $U(2) = \{-e_1, -e_2, (0,0)\}$. In such a case, the RPCP does not admit a solution. For instance, if the adversary continually chooses $d = 1$, the vehicle will have to keep moving north or east. Hence, after no more than 3 steps, it will be forced to leave S. This situation is shown on the left side of Fig. 1.

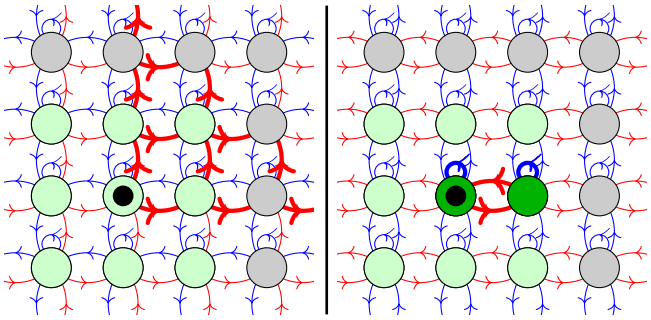

Fig. 1. The picture on the left illustrates a counterexample to solvability of the RPCP for $U(1) = \{e_1, e_2\}$, $U(2) = \{-e_1, -e_2, (0,0)\}$ in Example 1. The safe set S is denoted in light green. The vehicle's initial position $x(0) = (0,0)$ is denoted by a black circle. Possible vehicle movements from each $x \in \mathbb{Z}^n$ are denoted by an arrow. Red arrows denote available movements when the adversary chooses $d = 1$, and blue arrows denote available movements when $d = 2$. Possible movements in the case when the adversary chooses $d(0) = d(1) = d(2) = 1$ are drawn thickly. The picture on the right illustrates of solvability of the FPCP in Example 1. Same notation as in the left is used. The partition $\{U(1), U(2)\}$ is chosen in such a way that, regardless of the choice of $d(t)$, the vehicle can always remain in the dark green subset of the safe set.

On the other hand, the FPCP admits a solution for $m = 2$. Let $U(1) = \{e_1, -e_1\}$ and $U(2) = \{e_2, -e_2, (0,0)\}$. Then, when the adversary chooses $d = 1$ for the first time, the vehicle can choose to move east, then west the next time, then east again, etc. If the adversary chooses $d = 2$, the vehicle can remain in place. Hence, the vehicle will always remain within S. Such a strategy is depicted on the right side of Fig. 1. □

We now continue towards providing a solution for the RPCP and the FPCP.

III. Game Formulation

The RPCP can be easily formulated as the question of existence of a winning strategy in the following two-player game.

Game 1: Let $S \subseteq \mathbb{Z}^n$ and $x(0) \in S$. Let $\mathcal{U} \subseteq \mathbb{Z}^n$ be finite, and $\{U(1), \dots, U(m)\}$ be a partition of $\mathcal{U}$. Let $G = (V, E)$ be a graph with $V = \mathbb{Z}^n$ and $E = \{(x, y) \mid y - x \in \mathcal{U}\}$, and $l : E \to [m]$ a labeling given by

$$l(x, y) = d \qquad \text{if } y - x \in U(d). \tag{2}$$

The game proceeds as follows. Before time $t = 0$, a token is placed at $x(0)$. At every time step t, Player 1 first chooses an element $d \in [m]$. Then, Player 2 chooses an element $x(t+1) \in V$ such that $(x(t), x(t+1)) \in E$ and $l(x(t), x(t+1)) = d$, if such an element exists, and moves the token to $x(t+1)$. The game now proceeds to the next time step. Player 2 wins the game if it can always move the token, and the token remains within S for all $t \in \mathbb{N}_0$. Otherwise, Player 1 wins.

Proposition 1: The RPCP admits a solution if and only if there exists a winning strategy for Player 2 in Game 1.

Proof: By taking $u(t) = x(t+1) - x(t)$, it is clear that the movement of the token in Game 1 corresponds to (1). The requirement that $(x(t), x(t+1)) \in E$ and $l(x(t), x(t+1)) = d$ corresponds to the requirement that $u(t) \in U(d(t))$. Thus, Player 2 has a winning strategy in Game 1 if and only if there exists $u(t) \in U(d(t))$, possibly dependent on all previous inputs $d(0), \dots, d(t)$, such that $x(t) \in S$. The latter statement is exactly the statement of the RPCP. ■

Game 1 is a turn-based safety/reachability game with complete information as described [16]. Thus, for finite S, the RPCP can be solved in linear time with respect to the size of S [16]. In the remainder of this paper, we focus on the FPCP. In a game-theoretical setting, the FPCP can be posed as follows.

Game 2: Let S, $x(0)$, $\mathcal{U}$, and $G = (V, E)$ be as in Game 1. Let $m \in \mathbb{N}$. At time $t = -1$, Player 2 chooses $U(d) \subseteq \mathcal{U}$ for all $d \in [m]$ in such a way that $\{U(d) \mid d \in [m]\}$ is a partition of $\mathcal{U}$. Then, each edge $(x, y) \in E$ is labeled as in (2). After this step, the game proceeds the same as Game 1.

Analogously to Proposition 1, it can be easily shown that the FPCP admits a solution if and only if Player 2 has a winning strategy in Game 2.

The problem of the existence of a winning strategy in Game 2 can nominally be solved by reducing it to the problem of existence of a winning strategy in Game 1. Namely, every choice of a partition $\{U(d) \mid d \in [m]\}$ at time $t = -1$ generates a different instance of Game 1, so Player 2 has a winning strategy in Game 2 if and only if there exists a partition $\{U(d) \mid d \in [m]\}$ for which Player 2 has a winning strategy in Game 1. However, an algorithm that determines a winning strategy for Game 2 by considering all partitions $\{U(d) \mid d \in [m]\}$ is infeasible for large $\mathcal{U}$, as the number of those partitions is not less than $m^{|\mathcal{U}|-m}$ [24].

In the following section, we propose a graph-theoretical approach to the problem of determining the existence of winning strategies for Player 2 in the above games, resulting in easily computable conditions for the existence of a partition and a controller in the FPCP.

IV. Graph Labeling Problem

The previous section interprets system motion as a game on a labeled graph. By building upon this approach, we can convert the problem of finding a partition of the set of control inputs that admits a safe control policy — the FPCP — to an equivalent problem of labeling of graph edges.

Theorem 1: Let S, $x(0)$, $\mathcal{U}$, and G be as in Game 1. The FPCP admits a solution if and only if there exist an induced subgraph $G_{\hat{S}} = (\hat{S}, E_{\hat{S}}) \subseteq G$ with $\hat{S} \subseteq S$ and a labeling $l : E_{\hat{S}} \to [m]$ such that the following properties hold:

(C1) $x(0) \in \hat{S}$,

(C2) for all $x \in \hat{S}$,

$$l\left(\{(x, x') \in E_{\hat{S}} \mid x' \in \hat{S}\}\right) = [m],$$

and

(C3) if $(x,y),(x',y') \in E_{\hat{S}}$ satisfy $y-x = y'-x'$, then $l(x,y) = l(x',y')$.

Proof: As previously noted, the FPCP admits a solution if and only if there exists a winning strategy for Player 2 in Game 2. Assume first that such a winning strategy exists, with the corresponding partition $\{U(d) \mid d \in [m]\}$ and a labeling $l : E \to [m]$ that satisfies (2). Let us now define $G_{\hat{S}} = (\hat{S}, E_{\hat{S}})$ as the induced subgraph of G with its vertex set $\hat{S}$ consisting of *all* the values that the system state $x(t)$ can possibly assume under the chosen winning strategy, for *all* potential input sequences $d : \mathbb{N}_0 \to [m]$. We claim that $G_{\hat{S}}$, with the labeling l restricted to $E_{\hat{S}}$, satisfies (C1)–(C3).

First, since $\hat{S}$ is constructed from the winning strategy of Player 2, $x(0) \in \hat{S} \subseteq S$. Thus, (C1) holds. Property (C2) holds because, by definition of $\hat{S}$, for each $x \in \hat{S}$ there exists a $t \geq 0$ and a sequence $d(0), \dots, d(t-1)$ such that $x(t) = x$, and for each $d' \in [m]$, setting $d(t) = d'$ requires that $l(x(t), x(t+1)) = d'$. Property (C3) holds by (2).

In the other direction, assume that there exist an induced subgraph $G_{\hat{S}}$, $\hat{S} \subseteq S$, and a labeling function $l : E_{\hat{S}} \to [m]$ that satisfies (C1)–(C3). We will prove that the FPCP admits a solution.

Define

$$\tilde{U}(d) = \big\{ y - x \mid (x,y) \in E_{\hat{S}},\, l(x,y) = d \big\} \tag{3}$$

for all $d \in \{1, \dots, m\}$, and

$$\begin{aligned} U(d) &= \tilde{U}(d) \text{ for all } d \leq m-1, \\ U(m) &= \tilde{U}(m) \bigcup \left(\mathcal{U} \setminus \bigcup_{d=1}^{m-1} \hat{U}(d) \right). \end{aligned} \tag{4}$$

Clearly, $\{U(1), \dots, U(m)\}$ is a partition of $\mathcal{U}$. We define $\tilde{l} : E \to [m]$ by (2), with $U(d)$ defined as in (3)–(4). For any $(x,y) \in E_{\hat{S}}$, $\tilde{l}(x,y) = d$ if and only if $y - x \in U(d)$ by (2), which by (3)–(4) implies $l(x,y) = d$. Thus, $\tilde{l}$ and l are the same on $E_{\hat{S}}$, so with a standard abuse of notation, we will refer to $\tilde{l}$ as l in the remainder of the proof.

Let us now define $\hat{u} : \hat{S} \times [m] \to \mathcal{U}$ as any function with a following property:

$$\hat{u}(x,d) \in \{ y - x \mid y \in \hat{S},\, (x,y) \in E_{\hat{S}},\, l(x,y) = d \}. \tag{5}$$

We note that the existence of a function $\hat{u}$ that satisfies (5) follows from (C2), although uniqueness is not guaranteed.

We claim that any system run given by $x(t+1) = x(t) + \hat{u}(x(t), d(t))$ results in the system state remaining within $\hat{S} \subseteq S$, and that $\hat{u}(x(t), d(t)) \in U(d(t))$ for all $t \in \mathbb{N}_0$. For the claim that $x(t) \in \hat{S}$ for all t, we proceed by induction. By (C1), $x(0) \in \hat{S}$. Assume now that $x(t) \in \hat{S}$. Then, $x(t+1) = x(t) + \hat{u}(x(t), d(t)) \in \hat{S}$ by (5).

For the claim that $\hat{u}(x(t), d(t)) \in U(d(t))$ for all t, we note that $l(x(t), x(t) + \hat{u}(x(t), d(t))) = d(t)$ by (5), so $\hat{u}(x(t), d(t)) \in U(d(t))$ by (3)–(4).

Thus, $\hat{u}$ is a solution to the RPCP for the partition $\{U(1), \dots, U(m)\}$. Hence, the FPCP admits a solution. ■

Remark 1: In the latter direction in the proof of Theorem 1, technically we constructed a memoryless policy $\hat{u} : \hat{S} \times [m] \to \mathcal{U}$ instead of a memory-conscious policy $\hat{u} : \cup_{i=1}^{\infty} [m]^i \to \mathcal{U}$ as required in the RPCP. Thus, Theorem 1 also shows that Game 1 and Game 2 admit a winning strategy for Player 2 if and only if they admit a *memoryless* winning strategy, which was also discussed in [16].

With Theorem 1 in mind, the FPCP can be transformed into the following problem.

Invariant subgraph labeling problem (ISLP): Let S, $x(0)$, $\mathcal{U}$, m, and G be as in Game 1. Let $m \in \mathbb{N}$. Determine whether there exist an induced subgraph $G_{\hat{S}} = (\hat{S}, E_{\hat{S}}) \subseteq G$ with $\hat{S} \subseteq S$ and a labeling $l : E_{\hat{S}} \to [m]$ which satisfy (C1)–(C3).

Let us briefly note that if one was to omit requiring (C3) from the ISLP, such a problem reduces to finding an induced subgraph $G_{\hat{S}} \subseteq G$ with $x(0) \in \hat{S} \subseteq S$ and $\mathrm{mindeg}(G_{\hat{S}}) \geq m$. This problem is a variant of the *minimum subgraph of minimum degree* problem; see, e.g., [25] and the references therein. We now proceed to determine sufficient and necessary conditions for the ISLP to admit a solution.

V. Conditions for a Good Labeling

As discussed above, property (C2) in Theorem 1 trivially imposes a simple necessary condition for the ISLP to admit a solution.

Proposition 2: If there exist an induced subgraph $G_{\hat{S}}$ and a labeling l satisfying the conditions of ISLP, then

$$\mathrm{mindeg}(G_{\hat{S}}) \geq m.$$

The condition given in Proposition 2 is not sufficient for existence of a labeling satisfying the conditions of the ISLP. The following example gives an induced subgraph $G_{\hat{S}} \subseteq G$ with $\mathrm{mindeg}(G_{\hat{S}}) \geq m$ such that no labeling $l : E_{\hat{S}} \to [m]$ satisfies (C2)–(C3).

Example 2: Consider $n = 2$, $m = 3$, $x(0) = 0$, $S = \{x \in \mathbb{Z}^2 \mid \|x\|_1 \leq 1\}$, and $U = \{u \in \mathbb{Z}^2 \mid \|u\|_\infty = 1\}$. Let $\hat{S} = S$. Clearly, $x(0) \in \hat{S}$, and, as illustrated in Fig. 2, $\mathrm{mindeg}(G_{\hat{S}}) \geq m$. Nonetheless, S does not admit a labeling satisfying both (C2) and (C3). Assume otherwise. Let $l : E_{\hat{S}} \to [m]$ be such a labeling. By (C3), l is translation-invariant. Thus, we denote by $\hat{l}(1)$ the label of all edges that point north (i.e., $(x,y) \in E_{\hat{S}}$ such that $y - x = (0,1)$), $\hat{l}(2)$ the label of NE edges ($(x,y) \in E_{\hat{S}}$ such that $y - x = (1,1)$), $\hat{l}(3)$ for E edges, etc. By applying (C2) to

(i) vertices $(0,-1)$, $(-1,0)$, $(0,1)$, and $(1,0)$, respectively, we can conclude that, for each $k \in \{0,1,2,3\}$, $\hat{l}(2k)$, $\hat{l}(2k+1)$, and $\hat{l}(2k+2)$ need to be all different (for ease of notation, we identify $\hat{l}(0)$ with $\hat{l}(8)$),

(ii) vertex $(0,0)$, we note that $\hat{l}(1)$, $\hat{l}(3)$, $\hat{l}(5)$, and $\hat{l}(7)$ need to have three different values.

Now, from (ii), assume without loss of generality that $\hat{l}(1) = 1$, $\hat{l}(3) = 2$, and $\hat{l}(5) = 3$. Then, by (i) for $k = 0$ and $k = 1$, $\{\hat{l}(8), \hat{l}(2)\} = \{2,3\}$ and $\{\hat{l}(2), \hat{l}(4)\} = \{1,3\}$. Hence, $\hat{l}(2) = 3$, $\hat{l}(8) = 2$, and $\hat{l}(4) = 1$. Since $\{\hat{l}(4), \hat{l}(6)\} = \{1,2\}$ by (i) for $k = 2$, we have $\hat{l}(6) = 2$. Thus, $\hat{l}(8) = \hat{l}(6)$, which is in contradiction with (i) for $k = 3$. □

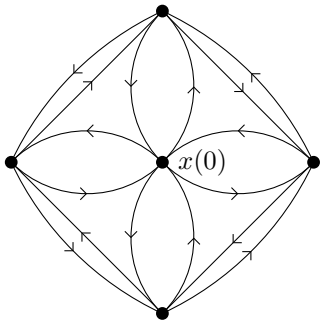

Fig. 2. An illustration of Example 2. The vertices of $\hat{S} = S$ and corresponding directed edges of $E_{\hat{S}}$ are denoted in black.

Even though Proposition 2 only gives a necessary condition for the ISLP to admit a solution, there does exist a related sufficient condition. Namely, if there exists an induced subgraph $G_{\hat{S}}$ with large enough $\mathrm{mindeg}(G_{\hat{S}})$, then there exists a labeling of $E_{\hat{S}}$ which solves the FPCP. We prove such a result using the probabilistic method (see, e.g., [18], [26], [27] for more details).

Theorem 2: Let S, $x(0)$, $\mathcal{U}$, and $G = (V, E)$ be as in Game 1. If there exists a finite induced subgraph $G_{\hat{S}} = (\hat{S}, E_{\hat{S}}) \subseteq G$ with $x(0) \in \hat{S} \subseteq S$ and

$$\mathrm{mindeg}(G_{\hat{S}}) \geq m \ln \left(m|\hat{S}| \right), \tag{6}$$

then there exists a labeling $l : E_{\hat{S}} \to [m]$ such that $G_{\hat{S}}$ and l satisfy properties (C1)–(C3).

Proof: Let us label each element $u \in \mathcal{U}$ by $\hat{l}(u) \in [m]$, where each label is chosen independently and uniformly. We define $l : E_{\hat{S}} \to [m]$ by $l(x, y) = \hat{l}(y - x)$. By definition of l, (C3) is satisfied. Property (C1) is also satisfied by the theorem assumptions.

Let B be the event that the label l does not satisfy (C2), i.e., that there exists a vertex $x \in \hat{S}$ such that

$$l\left(\{(x, x') \in E_{\hat{S}} \mid x' \in \hat{S}\} \right) \neq [m]. \tag{7}$$

Define B_x as the event that l satisfies (7) for the particular $x \in \hat{S}$. In particular, define B_x^i as the event that $i \notin l(\{(x, x') \in E_{\hat{S}} \mid x' \in \hat{S}\})$.

If we can show that $\Pr(B) < 1$, this will mean that there exists at least one labeling l such that B does *not* occur, i.e., that (C1)–(C3) are all satisfied.

By the definitions of B_x and B_x^i and the union bound [28], we obtain

$$\Pr(B) \geq \sum_{x \in \hat{S}} \Pr(B_x) \geq \sum_{\substack{x \in \hat{S} \\ i \in [m]}} \Pr(B_x^i).$$

Hence, if we show that

$$\Pr(B_x^i) < 1/(m|\hat{S}|) \tag{8}$$

holds for all $x \in \hat{S}$ and $i \in [m]$, we are done.

Consider the event B_x^i for fixed $x \in \hat{S}$ and $i \in [m]$. For each edge $(x, x') \in E_{\hat{S}}$, $x' - x \in \mathcal{U}$ is different. Thus, labels $l(x, x')$ have been chosen uniformly and independently. Consequently,

$$\begin{aligned} \Pr(B_x^i) &= \Pr\left(l(x, x') \neq i \text{ for all } (x, x') \in E_{\hat{S}} \right) \\ &= \prod_{(x,x') \in E_{\hat{S}}} \Pr\left(l(x, x') \neq i \right) = \prod_{(x,x') \in E_{\hat{S}}} (1 - 1/m). \end{aligned}$$

Hence, $\Pr(B_x^i) = (1 - 1/m)^{\deg_{G_{\hat{S}}}(x)} \leq (1 - 1/m)^{\mathrm{mindeg}(G_{\hat{S}})}$. By noting that $(1 - 1/m)^m < e^{-1}$ (see, e.g., [29]), we obtain $\Pr(B_x^i) \leq (1 - 1/m)^{\mathrm{mindeg}(G_{\hat{S}})} < e^{-\mathrm{mindeg}(G_{\hat{S}})/m}$. We now obtain (8) from (6). ∎

Theorem 2 gives a condition for solving the ISLP, i.e., the FPCP, based on finding a suitable subset $\hat{S}$ of the safe set. One way of producing such a subset is by finding a sufficiently dense subgraph of S, with a suitable definition of density. In the interest of brevity, we omit further details. We provide two illustrative examples of determining $\hat{S}$ in Section VII.

Returning to the running example, construction on the right side of Fig. 1, where $\mathrm{mindeg}(G_{\hat{S}}) = m < m \ln(m|\hat{S}|)$, shows that the condition expressed in Theorem 2 is not necessary for the solvability of the FPCP. We will return to this example in Section VII, where we provide some intuition for the "reason" that it yields a solution to the FPCP, even though it does not satisfy the sufficient condition expressed in Theorem 2.

VI. Efficient Labeling Algorithm

The proof of Theorem 2 does not provide a mechanism for constructing a good labeling. Instead, it merely states that a uniformly chosen labeling will solve the FPCP with probability $1 - \Pr(B) \geq 1 - m|\hat{S}|(1 - 1/m)^{\mathrm{mindeg}(G_{\hat{S}})}$. Thus, an algorithm that randomly chooses labelings until it reaches one that solves the FPCP is going to have expected computational complexity no greater than

$$O\left(\frac{|E_{\hat{S}}|}{1 - m|\hat{S}|(1 - 1/m)^{\mathrm{mindeg}(G_{\hat{S}})}} \right),$$

assuming that a random draw is performed in $O(1)$ time, and including the time to verify whether a labeling satisfies (C2). Thus, if $m|\hat{S}|(1 - 1/m)^{\mathrm{mindeg}(G_{\hat{S}})} \approx 1$, a randomized algorithm might take a substantial amount of time to finish.

We now present an alternative deterministic algorithm that produces a correct labeling in $O(|E_{\hat{S}}| + |\mathcal{U}|m|\hat{S}|)$ operations.

Algorithm 1: Let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$. Define a labeling $\hat{l}$ on $\mathcal{U}$ inductively as follows. Let

$$\begin{aligned} \bar{l}_i \in \operatorname*{argmin}_{l' \in [m]} \sum_{\substack{x \in \hat{S} \\ j \in [m]}} \Pr(B_x^j \mid \hat{l}(u_1) = \bar{l}_1, \ldots, \\ \ldots, \hat{l}(u_{i-1}) = \bar{l}_{i-1}, \hat{l}(u_i) = l') \end{aligned} \tag{9}$$

and define $\hat{l}(u_i) = \bar{l}_i$ for $i = 1, 2, \ldots, |\mathcal{U}|$, where labeling $l : E_{\hat{S}} \to [m]$ is given by $l(x, y) = \hat{l}(y - x)$ for all $(x, y) \in E_{\hat{S}}$.

Theorem 3: Assume that (6) holds for a finite induced subgraph $G_{\hat{S}}$ with $x(0) \in \hat{S} \subseteq S$. Let $\hat{l}$ and l be defined as in Algorithm 1. Then, $G_{\hat{S}}$ and l satisfy properties (C1)–(C3).

Proof: By (9), for each $i \in [|\mathcal{U}|]$,

$$\sum_{\substack{x\in\hat{S}\\ j\in[m]}} \Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=\bar{l}_i) \leq$$

$$\sum_{k\in[m]} \frac{1}{m} \sum_{\substack{x\in\hat{S}\\ j\in[m]}} \Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=k) =$$

$$\sum_{\substack{x\in\hat{S}\\ j\in[m]}} \sum_{k\in[m]} \frac{1}{m} \Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=k) \leq$$

$$\sum_{\substack{x\in\hat{S}\\ j\in[m]}} \Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_{i-1})=\bar{l}_{i-1}).$$

Hence, inductively,

$$\sum_{\substack{x\in\hat{S}\\ j\in[m]}} \Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_{|\mathcal{U}|})=\bar{l}_{|\mathcal{U}|}) \leq \sum_{\substack{x\in\hat{S}\\ j\in[m]}} \Pr(B_x^j) < 1, \tag{10}$$

where the last inequality holds by the proof of Theorem 2. On the other hand, l is entirely defined by $\hat{l}(u_1),\dots,\hat{l}(u_{|\mathcal{U}|})$. Hence, $\Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_{|\mathcal{U}|})=\bar{l}_{|\mathcal{U}|})$ equals either 0 or 1 for each $x \in \hat{S}$, $j \in [m]$. By (10), we thus have $\Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_{|\mathcal{U}|})=\bar{l}_{|\mathcal{U}|})=0$ for all $x \in \hat{S}$, $j \in [m]$, i.e., $G_{\hat{S}}$ and l satisfy the conditions of the ISLP. ■

Proposition 3: Algorithm 1 can be performed in $O(|E_{\hat{S}}|+|\mathcal{U}|m|\hat{S}|)$ operations.

Proof: Clearly, the computational complexity of Algorithm 1 depends on the complexity of solving the optimization problem in (9) for each $i \in [|\mathcal{U}|]$. For each $i \in [|\mathcal{U}|]$, $x \in \hat{S}$, and $j \in [m]$, if $j=\bar{l}_k$ for some $k \in \{1,\dots,i\}$ and $x+u_k \in \hat{S}$, then $\Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=\bar{l}_i)=0$. Otherwise,

$$\Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=\bar{l}_i) = (1-1/m)^{|\{i<k\leq|\mathcal{U}| \;\mid\; x+u_k\in\hat{S}\}|}.$$

Thus, if we precompute whether $x+u_k \in \hat{S}$ for each $x \in \hat{S}$ and $k \in [|\mathcal{U}|]$, and all values $|\{i<k\leq|\mathcal{U}| \mid x+u_k\in\hat{S}\}|$, which can be performed in $O(E_{\hat{S}})$ operations, computing (9) can be performed in $m|\hat{S}|$ time for each $i \in \{1,\dots,|\mathcal{U}|\}$, by merely updating all $\Pr(B_x^j \mid \hat{l}(u_1)=\bar{l}_1,\dots,\hat{l}(u_i)=\bar{l}_i)$ at the end of step i. Hence, Algorithm 1 indeed operates in $O(E_{\hat{S}}+|\mathcal{U}|m|\hat{S}|)$ time. ■

Provided with a labeling $l : E_{\hat{S}} \to [m]$, given in Algorithm 1, the system design and control policy which solve the FPCP are given by (3)–(4) and (5). We now proceed to illustrate the obtained results on two practical scenarios.

VII. Examples

A. Damaged Vehicle

Having given conditions for solvability of the RPCP and the FPCP, we return to our running example. Let us consider a vehicle operating on $V=\mathbb{Z}^n$, with the ability to either move along the coordinate axes or stay in place, i.e., $\mathcal{U}=\{0,\pm e_1,\dots,\pm e_n\}$. Naturally, only $n \leq 3$ makes direct physical sense. A similar example has been considered in the context of safety games in [17]. However, in that paper the agent and the adversary alternate in taking control of the vehicle, and the focus of the paper was on efficient computation of safe control policies for a given system design, and not on determining a good system design.

The safety objective that we consider is that the vehicle remains close to its initial position $x(0)=0$, i.e., $S=\{x \mid \|x\|_\infty \leq k\}$ for some $k \in \mathbb{N}_0$. As we showed in Example 1, there exists a safe system design for $n=2$, $m=2$, and $k=1$. In this section, we are interested in discussing the maximal loss of control that still enables a safe system design, i.e., for a given n and k, the maximal m such that the FPCP admits a solution.

It is clear that if $k=0$, the agent cannot afford any loss of authority, i.e., the only acceptable m equals 1. If $k \geq 1$, we claim that the maximal m equals $n+1$.

Let us first show that the FPCP has a solution for $m=n+1$. A partition $\{U_1,\dots,U_{n+1}\}$ that admits a solution to the RPCP is given by $U_i=\{e_i,-e_i\}$ for $i \leq n$, and $U_{n+1}=\{0\}$. Indeed, analogously to the construction on the right side of Fig. 1, a control policy which alternately chooses e_d and $-e_d$ every time the adversary chooses input $d \in [n]$, and 0 if the adversary chooses $d=n+1$, results in the agent's state always remaining in $\hat{S}=\{x \mid \|x\|_\infty \leq 1\}$.

On the other hand, if $m \geq n+2$, since there is a total of $2n$ non-zero elements in $\mathcal{U}$, for any partition $\{U_1,\dots,U_m\}$, some partition element U_j will equal $\{e_i\}$ or $\{-e_i\}$ for some i. However, by then repeatedly choosing $d(t)=j$, the adversary can be assured that $\|x(t)\|_\infty=t$, i.e., $\|x\|_\infty > k$ after finitely many steps. Thus, the maximal value of m for which the FPCP admits a solution is indeed $n+1$.

If $m=n+1$ and $\hat{S}=S=\{x \mid \|x\|_\infty \leq 1\}$, sufficient condition (6) from Theorem 2 does not hold, as $\mathrm{mindeg}(G_{\hat{S}})=m<m\ln(m|\hat{S}|)$. Nonetheless, the solution to the FPCP exists. Let us briefly discuss this gap between sufficiency and necessity of condition (6). The proof of Theorem 2 relies on some degree of genericity of a correct labeling, i.e., a positive probability that a randomly chosen labeling will be correct. On the other hand, the solution to the FPCP when $m=n+1$ is highly structured. Namely, each element of $\{U_1,\dots,U_m\}$ needs to equal $\{0\}$ or $\{e_i,-e_i\}$ for some i. Otherwise, there will exist U_j that equals $\{e_i\}$ or $\{-e_i\}$ for some i, and by repeating $d(t)=j$, the adversary will be able to force the system state to move arbitrarily far away from $x(0)$. Hence, the partition that yields a solution to the RPCP is in fact unique up to a permutation: $U(i)=\{e_i,-e_i\}$ for all $i \leq n$, and $U(n+1)=\{0\}$. Thus, as n increases, the probability of a uniformly chosen partition yielding a solution to the RPCP tends to 0.

B. Communication over a Channel

We now move from the setting of damaged autonomous systems to that of user-responsive systems. Consider the framework — originally introduced in [11] — where, at

every time t, a message chosen from some finite message set $\mathcal{M}$, $|\mathcal{M}| = m$, is sent over a communication channel. Each message is encoded as a bit-string (i.e., *codeword*) of some fixed length n. This codeword does not need to be the same every time the same message is sent; there could be multiple ways to communicate the same message. However, two different messages cannot be encoded in the same way.

The *running digital sum* (RDS) $x(t)$ is defined as the vector consisting of differences in the number of 1's and 0's that were sent in each coordinate of the bit-string until time t. Thus, $x(t)$ satisfies (1) for $x(0) = 0$, where $u(t)$ is an encoding of the message passed at time t, with zeros in the bit-string replaced by -1's, and $\mathcal{U} = \{-1, 1\}^n$ [22]. An illustration of such a system for $n = 2$ is given in Fig. 3.

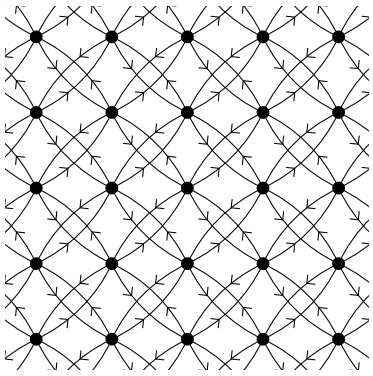

Fig. 3. An illustration of the dynamical system that describes the RDS. The vertices of G and the corresponding directed edges of E are denoted in black.

Encoding policies for which the RDS in a channel remains small regardless of the passed messages naturally reduce the effects of various categories of noise [22], [30]. Since encodings of different messages are pairwise disjoint, the problem of constructing encoding policies with bounded RDS can be naturally interpreted as the FPCP, with the safe set $S = \{x \mid \|x\|_\infty \leq k\}$. In this section, we are primarily interested in finding the smallest codeword length n such that there exists an encoding policy for which the RDS remains within S.

For $k = 0$, there clearly does not exist n which yields a solution for the RPCP. For $k = 1$, the only m for which there exists an n which yields a solution for the RPCP is $m = 1$, and in that case $n = 1$ suffices. For $k \geq 2$, a bound on n can be obtained from Theorem 2 as follows.

Proposition 4: Let $m, n \in \mathbb{N}$, $\mathcal{U} = \{-1, 1\}^n$, and $x(0) = 0$. Then, if $n \geq 3\max(\log_2 m, 11)$, the FPCP admits a solution for $S = \{x \in \mathbb{Z}^n \mid \|x\|_\infty \leq 2\}$.

Proof: Let us define $\hat{S} = V_1 \cup V_2$, where $V_1 = \{-1, 1\}^n$, and $V_2 = \{(x_1, \ldots, x_n) \in \{-2, 0, 2\}^n \mid x_i = 0 \text{ for at least } n/2 \text{ } i\text{'s}\}$. We note that $x(0) \in \hat{S} \subseteq S$.

Let us examine the outgoing degree $\deg_{G_{\hat{S}}}(v)$ of every vertex $v \in \hat{S}$ in the induced subgraph $G_{\hat{S}} \subseteq G$. If $v = (v_1, v_2, \ldots, v_n) \in V_1$, then

$$\deg_{G_{\hat{S}}}(v) = \sum_{i \geq n/2} \binom{n}{i} \geq 2^{n-1}, \tag{11}$$

as the set of neighbors of v is given by all vertices $\overline{v} = (\overline{v}_1, \ldots, \overline{v}_n) \in \mathbb{Z}^n$ that satisfy (i) $\overline{v}_i \in \{0, 2v_i\}$ for all $i \in [n]$, and (ii) $\overline{v}_i = 0$ for at least $n/2$ i's. If $v \in V_2$, then

$$\deg_{G_{\hat{S}}}(v) \geq 2^{n/2}, \tag{12}$$

as the set of neighbors of v is given by all $\overline{v} \in \mathbb{Z}^n$ that satisfy $\overline{v}_i \in \{-1, 1\}$ if $v_i = 0$, and $\overline{v}_i = v_i/2$ otherwise. Thus, from (11) and (12), we obtain $\operatorname{mindeg}(G_{\hat{S}}) \geq 2^{n/2}$.

We note that $|\hat{S}| = |V_1| + |V_2| \leq 2^n + 3^n \leq 3^{n+1}$. Thus, $m \ln(m|\hat{S}|) \leq m \ln m + m(n+1)\ln 3 \leq n2^{n/3}\ln(2)/3 + (n+1)2^{n/3}\ln(3)$. It can be shown that $n2^{n/3}\ln(2)/3 + (n+1)2^{n/3}\ln(3) \leq 2^{n/2}$ for all $n \geq 33$. Thus, the conditions of Theorem 2 are satisfied. ∎

An illustration of the construction of $\tilde{S}$ used in the proof of Proposition 4 is given in Fig. 4, for $m = n = 2$. We note that Fig. 4 shows that it is possible to construct a labeling (i.e., partition $\{U_1, U_2\}$) even for $n = 2$, indicating that the bound in Proposition 4 is very liberal.

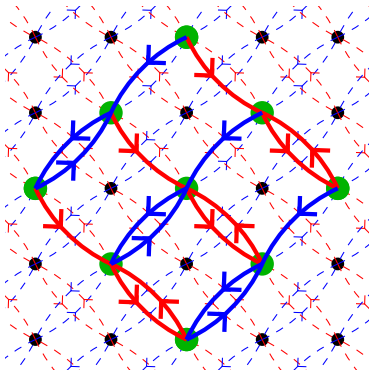

Fig. 4. An illustration of a safe labeling for the RDS with $m = 2$, $n = 2$. Set $\hat{S}$ is denoted in green. A control policy on $\hat{S}$ that ensures safety is described by thicker arrows. We note that each element in $\hat{S}$ has an outgoing thick arrow in each color pointing into $\hat{S}$. Hence, the system state controlled by such a law will always remain within $\hat{S}$, for any $x(0) \in \hat{S}$.

As it is necessary to use codewords (i.e., bit-strings) of length at least $\lceil \log_2 m \rceil$ to distinguish between m different messages, Proposition 4 states that, if we use three times as many bits as necessary, we can ensure that the RDS stays within the smallest possible bounds. We remark that from the proof of Proposition 4 it is clear that $n \geq 3\max(\log_2 m, 11)$ can be replaced by $n \geq (2+\varepsilon)\max(\log_2 m, n_\varepsilon)$ for any $\varepsilon \geq 0$, where $n_\varepsilon \to \infty$ as $\varepsilon \to 0$.

VIII. Conclusion and Future Work

This paper presents a preliminary discussion on control, design, and motion planning abilities of an autonomous system where the controller experienced a partial loss of control authority. The paper is primarily interested in developing sufficient and necessary conditions for existence of a safe control policy in such a partly controlled system. In order to obtain these conditions, we interpreted the system motion as a variant of an adversarial safety game on a graph, where one of the player's moves is to label the edges of the game graph. We showed that the safety objective in the original control system is attainable if and only if such a game has a winning

strategy, and showed that the game has a winning strategy if and only if there exists a labeling of the game graph that satisfies particular properties. We found a sufficient condition and a necessary condition for the existence of such a labeling in terms of minimal degrees of a subgraph of the original graph, and discussed how those conditions apply to the motion of an autonomous vehicle operating on an n-dimensional surface and to communication using a set of codewords of length n with a bounded running digital sum.

The primary avenue of future work is in broadening the scope of the considered framework. In addition to discussing system dynamics more general than (1) — which may be achieved by considering two-stage motions on a graph, one stage being involuntary ("drift"), and the other resulting from the performed actions — it is meaningful to consider a broader class of control specifications, rather than solely safety. In general, tasks for autonomous systems are often expressed by a temporal logic specification (e.g., "visit area A infinitely many times, never go into area B, and eventually reach area C"). Previous work on designing provably correct control policies — i.e., policies that are guaranteed to result in the system behavior satisfying a temporal logic specification — primarily deals with systems whose control abilities are not compromised; see [31] for a thorough study. While there is a substantial body of work (see, e.g., [32] and the references therein) on systems whose control capabilities may depend on the environment, procedures for determining provably correct control policies for such systems are computationally complex. Providing simple graph-based criteria for existence of a system design that admits a correct control policy would present a significant next step towards ensuring system resilience under partial loss of control authority.

REFERENCES

[1] S. Rathinam and R. Sengupta, "A safe flight algorithm for unmanned aerial vehicles," in *IEEE Aerospace Conference*, 2004, pp. 3025–3031.

[2] K. Chatzilygeroudis, V. Vassiliades, and J.-B. Mouret, "Reset-free trial-and-error learning for robot damage recovery," *Robotics and Autonomous Systems*, vol. 100, pp. 236–250, 2018.

[3] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson, and T. Smith, "Autonomous rovers for Mars exploration," in *IEEE Aerospace Conference*, 1999, pp. 237–251.

[4] M. A. Vatis, "Cyber attacks during the war on terrorism: A predictive analysis," Institute for Security, Technology, and Society, Dartmouth College, Tech. Rep., 2001.

[5] S. M. Amin and A. M. Giacomoni, "Smart grid — safe, secure, self-healing," *IEEE Power and Energy Magazine*, vol. 10, no. 1, pp. 33–40, 2012.

[6] Y. Zhu, J. Yan, Y. Tang, Y. L. Sun, and H. He, "Resilience analysis of power grids under the sequential attack," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2340–2354, 2014.

[7] M. Wolf, M. Minzlaff, and M. Moser, "Information technology security threats to modern e-enabled aircraft: A cautionary note," *Journal of Aerospace Information Systems*, vol. 11, no. 7, pp. 447–457, 2014.

[8] K. Hartmann and K. Giles, "UAV exploitation: A new domain for cyber power," in *8th International Conference on Cyber Conflict*, 2016, pp. 205–221.

[9] D. G. Feitelson and L. Rudolph, "Distributed hierarchical control for parallel processing," *Computer*, vol. 23, no. 5, pp. 65–77, 1990.

[10] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "A dynamic QoS-aware semantic web service composition algorithms," in *10th International Conference on Service-Oriented Computing*, 2012, pp. 623–630.

[11] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[12] J. Downer, "When failure is an option: Redundancy, reliability and regulation in complex technical systems," Centre for Analysis of Risk and Regulation, London School of Economics and Political Science, Tech. Rep. 53, 2009.

[13] C. J. Tomlin and J. Lygeros, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.

[14] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.

[15] J. Bernet, D. Janin, and I. Walukiewicz, "Permissive strategies: From parity games to safety games," *Theoretical Informatics and Applications*, vol. 36, pp. 261–275, 2002.

[16] L. Doyen and J.-F. Raskin, "Games with imperfect information: Theory and algorithms," in *Lectures in Game Theory for Computer Scientists*, K. R. Apt and E. Grädel, Eds. Cambridge University Press, 2011, pp. 185–212.

[17] D. Neider and U. Topcu, "An automaton learning approach to solving safety games over infinite graphs," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2016, pp. 204–221.

[18] N. Alon and J. H. Spencer, *The probabilistic method*. Wiley, 2008.

[19] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.

[20] N. Megow, K. Mehlhorn, and P. Schweitzer, "Online graph exploration: New results on old and new algorithms," *Theoretical Computer Science*, vol. 463, pp. 62–72, 2012.

[21] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[22] G. D. Cohen and S. Litsyn, "dc-constrained error-correcting codes with small running digital sum," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 949–955, 1991.

[23] M. Sghairi, A. de Bonneval, Y. Crouzet, J.-J. Aubert, and P. Brot, "Challenges in building fault-tolerant flight control system for a civil aircraft," *IAENG International Journal of Computer Science*, vol. 35, no. 4, 2008.

[24] B. C. Rennie and A. J. Dobson, "On Stirling numbers of the second kind," *Journal of Combinatorial Theory*, vol. 7, no. 2, pp. 116–121, 1969.

[25] O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh, "On the approximability of some degree-constrained subgraph problems," *Discrete Applied Mathematics*, vol. 160, pp. 1661–1679, 2012.

[26] P. Erdős, "Graph theory and probability," *Canadian Journal of Mathematics*, vol. 11, pp. 34–38, 1959.

[27] ——, "Graph theory and probability. II," *Canadian Journal of Mathematics*, vol. 13, pp. 346–352, 1961.

[28] S. S. Venkatesh, *The Theory of Probability: Explorations and Applications*. Cambridge University Press, 2012.

[29] V. H. Moll, *Numbers and Functions: From a Classical-Experimental Mathematician's Point of View*. American Mathematical Society, 2012.

[30] K. A. Schouhamer Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publishers, 2004.

[31] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[32] O. Kupferman, M. Y. Vardi, and P. Wolper, "Module checking," *Information and Computation*, vol. 164, no. 2, pp. 322–344, 2001.