



## Brief paper

Resilient distributed vector consensus using centerpoint<sup>☆</sup>Waseem Abbas<sup>a,\*</sup>, Mudassir Shabbir<sup>b,c</sup>, Jiani Li<sup>b</sup>, Xenofon Koutsoukos<sup>b</sup><sup>a</sup> Department of Systems Engineering, University of Texas at Dallas, TX, USA<sup>b</sup> Department of Computer Science, Vanderbilt University, Nashville, TN, USA<sup>c</sup> Department of Computer Science, Information Technology University, Lahore, Pakistan

## ARTICLE INFO

## Article history:

Received 17 August 2020

Received in revised form 20 August 2021

Accepted 11 October 2021

Available online 24 November 2021

## Keywords:

Resilient consensus

Computational geometry

Centerpoint

Fault tolerant networks

## ABSTRACT

In this paper, we study the resilient vector consensus problem in networks with adversarial agents and improve resilience guarantees of existing algorithms. A common approach to achieving resilient vector consensus is that every non-adversarial (or normal) agent in the network updates its state by moving towards a point in the convex hull of its *normal* neighbors' states. Since an agent cannot distinguish between its normal and adversarial neighbors, computing such a point, often called *safe point*, is a challenging task. To compute a safe point, we propose to use the notion of *centerpoint*, which is an extension of the median in higher dimensions, instead of the Tverberg partition of points, which is often used for this purpose. We discuss that the notion of centerpoint provides a complete characterization of safe points in  $\mathbb{R}^d$ . In particular, we show that a safe point is essentially an interior centerpoint if the number of adversaries in the neighborhood of a normal agent  $i$  is less than  $\frac{N_i}{d+1}$ , where  $d$  is the dimension of the state vector and  $N_i$  is the total number of agents in the neighborhood of  $i$ . Consequently, we obtain necessary and sufficient conditions on the number of adversarial agents to guarantee resilient vector consensus. Further, by considering the complexity of computing centerpoints, we discuss improvements in the resilience guarantees of vector consensus algorithms and compare with the other existing approaches. Finally, we numerically evaluate our approach.

© 2021 Elsevier Ltd. All rights reserved.

## 1. Introduction

Resilient consensus in a network of agents, some of which might be adversarial or faulty, has several applications in multi-robot networks, distributed computing, estimation, learning and optimization (for instance, see Abbas, Laszka, & Koutsoukos, 2018; Park & Hutchinson, 2017; Su & Vaidya, 2016; Sundaram & Gharesifard, 2015; Tseng & Vaidya, 2013). The main goal of resilient consensus is to ensure that all normal agents in a network agree on a common state despite some adversarial agents, whose identities are unknown to normal agents. If agents' states are vectors or points in  $\mathbb{R}^d$ ,  $d \geq 2$ , then the resilient consensus objective is to ensure that normal agents converge at some point in the convex hull of their initial states. One application is the

resilient multirobot rendezvous problem, where the goal is to gather mobile robots dispersed in a region at a common point in a distributed manner (Park & Hutchinson, 2017). The robots must move within their original convex region to avoid drifting to an unsafe region. A simple approach could be to run  $d$  instances of scalar resilient consensus, one for each dimension. However, as a result of this approach, normal agents might converge at a point outside the convex hull of their initial states, as discussed in Vaidya and Garg (2013). As a result, we cannot rely on resilient scalar consensus algorithms to achieve resilient vector consensus.

In this paper, we study the resilient vector consensus problem and propose a *resilient aggregation rule* for normal (non-adversarial) agents to update their states by combining their neighbors' states, which might include adversarial agents. In the proposed resilient aggregation rule, each normal agent computes a *centerpoint*, which is an extension of the median in higher-dimensional Euclidean space, of its neighbors' states. We then obtain necessary and sufficient conditions on the number of adversarial agents each normal node can have in its neighborhood while guaranteeing resilient vector consensus. We show that the centerpoint-based aggregation improves the resilience of the known vector consensus algorithms, for instance, the *Approximate Distributed Robust Convergence (ADRC)* algorithm recently

<sup>☆</sup> Some preliminary results are presented in Shabbir, Li, Abbas, and Koutsoukos (2020). The material in this paper was partially presented at the 2020 American Control Conference, July 1–3, 2020, Denver, CO, USA. This paper was recommended for publication in revised form by Associate Editor Claudio Altafini under the direction of Editor Christos G. Cassandras.

\* Corresponding author.

E-mail addresses: [waseem.abbas@utdallas.edu](mailto:waseem.abbas@utdallas.edu) (W. Abbas), [mudassir.shabbir@vanderbilt.edu](mailto:mudassir.shabbir@vanderbilt.edu) (M. Shabbir), [jiani.li@vanderbilt.edu](mailto:jiani.li@vanderbilt.edu) (J. Li), [xenofon.koutsoukos@vanderbilt.edu](mailto:xenofon.koutsoukos@vanderbilt.edu) (X. Koutsoukos).

proposed in Park and Hutchinson (2017). We note that computing a centerpoint of points in  $\mathbb{R}^d$  is different from taking the coordinate-wise median of points.

Various resilient vector consensus algorithms have been proposed in the literature, for instance, see Mendes, Herlihy, Vaidya, and Garg (2015), Park and Hutchinson (2017), Vaidya (2014), Vaidya and Garg (2013), Yan, Li, Mo, and Wen (2020) and Yan, Mo, Li, and Wen (2019). The main idea in these algorithms is that each normal agent updates its state iteratively by computing a state (point in  $\mathbb{R}^d$ ) that is guaranteed to lie in the *interior of the convex hull of its normal neighbors' states*. Since a normal agent cannot distinguish between its normal and adversarial neighbors, computing such a point, which is often referred to as the *safe point*, is the primary challenge in these consensus algorithms. To compute a safe point, previous works (Mendes et al., 2015; Park & Hutchinson, 2017) have utilized the idea of *Tverberg partition* of points in  $\mathbb{R}^d$  (discussed in Section 3). We argue that instead of computing Tverberg partition, it is better to use the notion of *centerpoint* in  $\mathbb{R}^d$  to compute safe points. We show that safe point is essentially an interior centerpoint of the neighbors' states. This perspective provides a complete characterization of safe points and enables improved resilience bounds for vector consensus algorithms. A centerpoint has been known to the discrete geometry community for a long time, and its properties and generalizations are an active topic of research. We summarize our contributions below:

- For a given set of  $N_i$  points in  $\mathbb{R}^d$ , of which any  $F_i$  are adversarial, we show the equivalence between the notion of a safe point and an interior centerpoint for  $N_i > F_i(d+1)$ . Here,  $F_i$  and  $N_i$  denote the number of adversarial agents and the total number of agents in the neighborhood of a normal agent  $i$ , respectively. Using this relationship, we discuss how the resilience of the vector consensus algorithm can be improved using centerpoint to compute a safe point.
- We generalize a sufficient condition for the existence of safe point to arbitrary  $d > 0$ , where  $d$  is the dimension of the state vector. In particular, we show that if  $F_i < \lfloor \frac{N_i}{d+1} \rfloor$ , then there always exists an *interior centerpoint*, and hence, a safe point. A similar condition for the existence of safe point was established in Park and Hutchinson (2017) for  $d \leq 8$  using the idea of Tverberg partition.
- Using the notion of centerpoint, we show that  $N_i > (d+1)F_i$  is not only sufficient but also a necessary condition for computing a safe point.
- We show that the centerpoint-based vector consensus algorithm is resilient to  $F_i < \lfloor \frac{N_i}{d+1} \rfloor$  adversaries in the neighborhood of a normal agent  $i$  in  $d = 2, 3$ . In higher dimensions ( $d > 3$ ), the centerpoint-based algorithm is resilient to  $F_i$  adversaries, where  $F_i < F_{max} = \Omega(\frac{N_i}{d^2})$ . This approach offers improved resilience compared to the Tverberg partition-based algorithm, where  $F_{max} = \Omega(\frac{N_i}{2d})$ .
- We numerically evaluate and compare our results with other algorithms by simulating resilient vector consensus in multirobot networks.

In the case of resilient *scalar* consensus, the *trimming* method has been an effective approach to aggregate neighbors' states. The basic idea is that a normal agent collects its neighbors' states (scalar), sorts them, and then disregards *extreme* values (e.g., LeBlanc, Zhang, Koutsoukos, & Sundaram, 2013; Sundaram & Ghahesifard, 2018). The trimming approach, however, is not directly applicable to the resilient vector consensus as the appropriate notion of vector states that are 'farthest' from a normal agent's own state is difficult to define. At the same time, implementing resilient scalar consensus using the trimming approach

in each dimension separately does not always solve the resilient vector consensus problem. Thus, in the resilient vector consensus case, the notion of *safe point*, which is a point obtained by a convex combination of node's normal neighbors' states, is quite useful. Using the idea of safe point, various algorithms and resilience bounds have been proposed in the literature for the resilient vector consensus and distributed optimization problems (Mendes et al., 2015; Tseng & Vaidya, 2013; Vaidya, 2014; Vaidya & Garg, 2013; Yan et al., 2020, 2019). Here, we focus on computational aspects and geometric characterization of safe points.

The rest of the paper is organized as follows: Section 2 introduces notations and preliminaries. Section 3 provides an overview of the resilient vector consensus algorithms, in particular, the ADRC algorithm and its resilience bounds. Section 4 discusses the notion of centerpoint for resilient vector consensus and presents main results in the paper. Section 5 presents a numerical evaluation of our results, and Section 6 concludes the paper.

## 2. Notations and preliminaries

We consider a network of agents modeled by a *directed graph*  $G = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  represents agents and  $\mathcal{E}$  represents interactions between agents. Each agent  $i \in \mathcal{V}$  has a  $d$ -dimensional state vector whose value is updated over time. The state of agent  $i$  at time  $t$  is represented by a point  $x_i(t) \in \mathbb{R}^d$ . An edge  $(j, i)$  means that  $i$  can observe the state of  $j$ . The *neighborhood* of  $i$  is the set of nodes  $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\} \cup \{i\}$ . For a given set of points  $X \subset \mathbb{R}^d$ , we denote its *convex hull* by  $\text{conv}(X)$ . A set of points in  $\mathbb{R}^d$  is said to be in *general positions* if no hyperplane of dimension  $d-1$  or less contains more than  $d$  points. A point  $x \in \mathbb{R}^d$  is an *interior point* of a set  $X \subset \mathbb{R}^d$  if there exists an open ball centered at  $x$  which is completely contained in  $X$ . We use terms *agents* and *nodes* interchangeably, and similarly use terms *points* and *states* interchangeably.

*Normal and Adversarial Agents* – There are two types of agents in the network, normal and adversarial. *Normal* agents, denoted by  $\mathcal{V}_n \subseteq \mathcal{V}$ , are the ones that interact with their neighbors synchronously and always update their states according to the consensus algorithm. *Adversarial* agents, denoted by  $\mathcal{V}_f \subset \mathcal{V}$ , are the ones that can change their states arbitrarily and do not follow the pre-defined state update rule. Moreover, an adversarial node can transmit different values to its different neighbors, which is referred to as the *Byzantine* model. The number of adversarial nodes in the neighborhood of a normal node  $i$  is denoted by  $F_i$ . For a normal node  $i$ , all nodes in its neighborhood are indistinguishable, that is,  $i$  cannot identify which of its neighbors are adversarial.

*Resilient Vector Consensus* – The goal of the resilient vector consensus is to ensure the following conditions:

- *Safety* – Let  $X(0) = \{x_1(0), x_2(0), \dots, x_n(0)\} \subset \mathbb{R}^d$  be the set of initial states of normal nodes, then at each time step  $t$ , and for any normal node  $i$ , the state value of  $i$ , denoted by  $x_i(t)$  should be in the  $\text{conv}(X(0))$ .
- *Agreement* –  $\forall \epsilon > 0, \exists t_\epsilon$ , such that for any normal node pair  $i, j, \|x_i(t) - x_j(t)\| < \epsilon, \forall t > t_\epsilon$ .

## 3. Background and resilient vector consensus

In this section, we provide an overview of resilient vector consensus algorithms in which the main idea is that each normal agent updates its state by moving towards a point in the convex hull of its normal neighbors. In particular, we discuss the *approximate distributed robust convergence (ADRC)* algorithm, proposed in Park and Hutchinson (2017). Then, we discuss improvement in the resilience guarantees of the algorithm by reconsidering

its computational aspects. We note here that our results are general and mainly provide an improved way for normal agents to aggregate their neighbors' states in the presence of adversarial agents. As a result, these results can be applied to a broader class of algorithms that require each normal agent in the network to aggregate its neighbors' states, for instance, resilient distributed diffusion (Li, Abbas, Shabbir, & Koutsoukos, 2020; Sayed, Tu, Chen, Zhao, & Towfic, 2013). The ADRC algorithm is a synchronous iterative algorithm that guarantees the consensus of normal agents if the number of adversarial agents in the neighborhood of each normal agent is bounded by a certain value that depends on  $d$ . The notion of  $F$ -safe point is crucial to understanding the algorithm.

**Definition 3.1** ( $F$ -safe Point). Given a set of  $N$  points in  $\mathbb{R}^d$ , of which at most  $F$  are adversarial, then a point  $s$  that is guaranteed to lie in the interior of the convex hull of  $(N - F)$  normal points is an  $F$ -safe point.

The ADRC algorithm relies on the computation of an  $F_i(t)$ -safe point in each iteration  $t$  by every normal agent  $i$  having  $N_i(t)$  agents in its neighborhood, of which at most  $F_i(t)$  are adversaries.<sup>1</sup> Each normal agent  $i$  updates its state as follows (Park & Hutchinson, 2017):

- In the iteration  $t$ , a normal agent  $i$  gathers the state values of its neighbors, and then computes an  $F_i(t)$ -safe point, denoted by  $s_i$ .
- Agent  $i$  then updates its state as below.

$$x_i(t+1) = \alpha_i(t)s_i(t) + (1 - \alpha_i(t))x_i(t), \quad (1)$$

where  $\alpha_i(t)$  is a dynamically chosen parameter in the range  $(0, 1)$ , according to the application (Park & Hutchinson, 2017).

**Convergence** – It is shown in Park and Hutchinson (2017) that if all normal agents update their states according to (1) after computing  $F_i(t)$ -safe points in each iteration  $t$ , and the sequence of graphs induced by normal agents is *repeatedly reachable*, then all normal agents converge at a common point. The notion of repeatedly reachable sequence of graphs is explained in Park and Hutchinson (2017, Section V-B). We note that two types of conditions need to be satisfied for resilient consensus: (a) The underlying network topology should be such that the sequence of graphs induced by normal agents is repeatedly reachable. (b) Each normal agent can compute a safe point in each iteration. Our focus in this paper is on the computation of a safe point, including conditions under which it is possible.

**Computation of Safe point** – It is the key step in the ADRC algorithm. For this, Park and Hutchinson (2017) utilize the notion of *Tverberg partition*. The main idea is to partition a set of  $N$  points in  $\mathbb{R}^d$  into  $(F + 1)$  subsets such that the convex hull of points in one subset has a non-empty intersection with the convex hull of points in any other subset. The intersection of convex hulls of these  $(F + 1)$  subsets is a *Tverberg region* and any interior point in such a region is an  $F$ -safe point. Thus, to achieve resilient consensus using ADRC algorithm, a normal agent  $i$  computes an  $F_i$ -safe point by computing an interior point in the Tverberg region of an appropriate Tverberg partition. Now, for given  $N_i$ ,  $F_i$  and  $d$ , the condition guaranteeing the existence of Tverberg partition is stated below (Park & Hutchinson, 2017; Reay, 1968; Roudneff, 2009).

**Theorem 3.2** (*Sufficient Condition*). Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$ , where  $d \in \{2, 3, \dots, 8\}$ , of which at most  $F_i$  points correspond to adversarial nodes, then it is possible to find an  $F_i$ -safe point if

$$F_i \leq \frac{N_i}{d+1} - 1. \quad (2)$$

In general, computing a Tverberg partition is an NP-hard problem and it could take  $d^{O(d^2)}N_i^{(d+1)^2+1}$  time to compute a Tverberg partition of  $N_i$  points in  $\mathbb{R}^d$  into  $\lceil N_i/(d+1) \rceil$  parts (Har-Peled & Zhou, 2020). The best known algorithm that computes it in a reasonable run time is an approximate algorithm (Mulzer & Werner, 2013), which has a time complexity of  $d^{O(1)}N_i$ . The algorithm is approximate in a sense that to have a partition of  $N_i$  points into  $r$  subsets,  $N_i \geq 2^d r$  (compared to the theoretical bound,  $N_i \geq r(d+1)$ ). Consequently, to compute an  $F_i$ -safe point in the presence of  $F_i$  adversarial neighbors, a normal agent  $i$  needs to have at least  $N_i \geq (F_i + 1)2^d$  agents in its neighborhood, which gives the following resilience bound (Park & Hutchinson, 2017).

**Theorem 3.3** (*Resilience Bound*). Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$ , of which at most  $F_i$  are adversarial, then it is possible to compute an  $F_i$ -safe point using Tverberg partition if

$$F_i \leq \left\lceil \frac{N_i}{2^d} \right\rceil - 1. \quad (3)$$

Note that (3) indicates *resilience* of the ADRC algorithm based on approximate Tverberg partitions to compute safe points. Here, we ask if it is possible to improve the resilience of the ADRC algorithm. In particular, we study the following problems in the rest of the paper.

- ADRC algorithm based on approximate Tverberg partition is resilient against  $\Omega\left(\frac{N_i}{2^d}\right)$  adversaries in the neighborhood of a normal agent  $i$ . Can we improve the resilience by computing safe points some other way?
- We have a sufficient condition for an  $F_i$ -safe point in (2). Can we provide a complete characterization of an  $F_i$ -safe point? In other words, what is a necessary and sufficient condition for a point to be an  $F_i$ -safe point?
- The sufficient condition in (2) holds for  $d \leq 8$ . Can we generalize results to dimensions  $d > 8$ ?

The primary difference between our proposed approach and previous methods is the characterization and computation of safe points. We discuss in detail that the centerpoint-based approach is better for the characterization and computation of safe points and also compare this approach with the existing methods in Section 4.5.

## 4. Resilient vector consensus and centerpoint

In this section, we show that it is possible to improve the resilience of ADRC algorithm by computing safe points differently, that is, by using the notion of *centerpoint*. Moreover, centerpoint provides a complete characterization of the safe point notion.<sup>2</sup>

### 4.1. $F$ -Safe point and interior centerpoint

The notion of centerpoint can be viewed as an extension of median in higher dimensions and is defined below.

<sup>1</sup> For simplicity, when the context is clear, we use  $N_i$  and  $F_i$  instead of  $N_i(t)$  and  $F_i(t)$ , respectively.

<sup>2</sup> For proofs of results in this section, we refer to Abbas, Shabbir, Li, and Koutsoukos (2021).

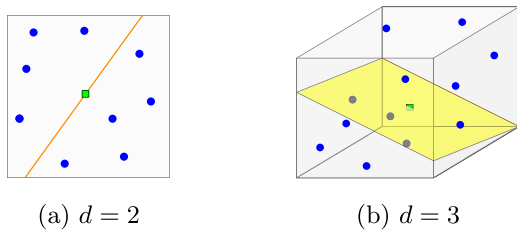


Fig. 1. Illustration of centerpoints (green squares).

**Definition 4.1 (Centerpoint).** Given a set  $S$  of  $N$  points in  $\mathbb{R}^d$  in general positions, a centerpoint  $p$  is a point, such that any closed halfspace<sup>3</sup> of  $\mathbb{R}^d$  that contains  $p$  also contains at least  $\frac{N}{d+1}$  points from  $S$ . Note that centerpoint does not need to be one of the points in  $S$ .

Intuitively, a centerpoint lies in the ‘center region’ of the point cloud, in the sense that there are enough points of  $S$  on each side of a centerpoint. More precisely, it is a point such that any hyperplane that goes through this point divides the set of points in  $S$  into two subsets, each of which contains at least  $\frac{1}{d+1}$  fraction of points. Fig. 1 illustrates centerpoint in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . In Fig. 1(a), any line through a centerpoint divides a set of 9 points into two subsets, each of which contains at least 3 points. Similarly, any plane through a centerpoint in Fig. 1(b) divides a set of 12 points into two parts, each containing at least 3 points.

A related notion of *point depth* is defined as follows:

**Definition 4.2 (Point Depth).** For a given point set, the *depth* of a point  $q$  is the maximum number  $\alpha$  such that every closed halfspace containing  $q$  contains at least  $\alpha$  points.

Thus, a centerpoint has depth at least  $\frac{N}{d+1}$ . A centerpoint does not need to be unique; in fact, there can be infinitely many centerpoints. The set of all centerpoints constitutes the *centerpoint region* or simply the *center region*, which is closed and convex. The existence of a center point for any given set  $S$  is guaranteed by the famous Centerpoint Theorem, which asserts that for any given point set in general positions in an arbitrary dimension, a centerpoint always exists (see Matoušek, 2002). We show that there is a close relationship between an  $F$ -safe point and an interior centerpoint, which can be exploited to provide a complete characterization of  $F$ -safe points. We state the following result (proof is in Abbas et al., 2021):

**Theorem 4.3.** For a given set of  $N_i$  points in  $\mathbb{R}^d$  (corresponding to  $N_i$  agents in the neighborhood of agent  $i$ ), if  $N_i > F_i(d + 1)$ , then an  $F_i$ -safe point is equivalent to an interior centerpoint.

We illustrate the equivalence between  $F_i$ -safe point and centerpoint through an example. Consider six points in  $\mathbb{R}^2$  in Fig. 2(a). The gray region is the set of all centerpoints. At the same time, the gray region is also the set of all 1-safe points. It means that no matter which one of the six points corresponds to an adversarial agent, every point in the gray region is guaranteed to lie in the convex hull of remaining five points corresponding to normal agents. For example, in Fig. 2(b), the red point is an adversary and the yellow region is the convex hull of normal agents’ points. We observe that all 1-safe points (gray region) lie inside the convex hull of normal agents’ points. The same is illustrated for other possibilities for a single adversarial point in Figs. 2(c)–(g).

<sup>3</sup> Recall that closed halfspace in  $\mathbb{R}^d$  is a set of the form  $\{x \in \mathbb{R}^d : a^T x \geq b\}$  for some  $a \in \mathbb{R}^d \setminus \{0\}$ .

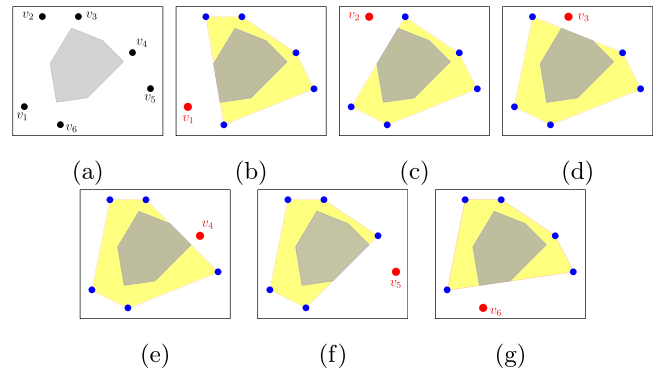


Fig. 2. The region of centerpoints (gray area) is same as the region of 1-safe points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

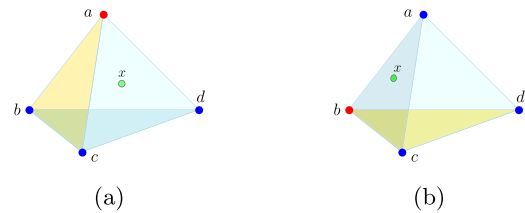


Fig. 3. An example of 4 points in  $\mathbb{R}^3$ . (a) A point  $x$  in the interior of tetrahedron cannot be 1-safe because if  $a$  is adversarial, then  $x$  is not contained in the convex hull of the normal points. (b) If a point  $x$  lies on facet, say  $a, b, c$ , and  $b$  is adversarial, then  $x$  is not contained in the convex hull of normal points  $a, c, d$ .

#### 4.2. Necessary condition for $F_i$ -safe point

We note that  $\frac{1}{d+1}$  is the best possible fraction (for  $F_i$ ) in Theorem 4.3, that is, there exist general node positions where allowing more adversarial nodes would mean that there is no safe point at all. For example, consider a set of three points  $a, b, c$ , in  $\mathbb{R}^2$  that lie on the vertices of a non-degenerate triangle.<sup>4</sup> If one of these points is malicious then there is no 1-safe point in the plane. Clearly, no point in the interior of the triangle can be 1-safe because it does not lie in the convex hull of normal points for any choice of point corresponding to an adversary. Similarly, any point that lies on an edge of the triangle, say  $\overline{ab}$ , cannot be 1-safe because if one of the endpoint, say  $a$ , is adversarial then the convex hull of the remaining points  $b$  and  $c$  does not contain this point. Now in  $\mathbb{R}^3$ , consider four points  $a, b, c, d$ , lying on the vertices of a non-degenerate tetrahedron. Assume one of these points, whose identity is unknown, is adversarial. A point  $x$  in the interior of tetrahedron cannot be 1-safe because if  $a$  is malicious then  $x$  is not contained in the convex hull of normal points  $b, c, d$ , as shown in Fig. 3(a). Similarly, consider a point  $x$  that lies on a facet, say  $a, b, c$ , of the tetrahedron. The point  $x$  cannot be 1-safe because if one of the points on its facet, say  $b$ , is adversarial then  $x$  is not in the convex hull of the remaining points  $a, c, d$ , as shown in Fig. 3(b). In the following, we show that this example can be generalized to arbitrary number of points and arbitrary dimension.

**Proposition 4.4.** For a set of  $N_i$  nodes in general positions, if  $F_i \geq \lceil \frac{N_i}{d+1} \rceil$ , then there exist general examples in which an  $F_i$ -safe point does not exist.

<sup>4</sup> A non-degenerate triangle has three points that do not all lie on a line.

### 4.3. Sufficient condition for $F_i$ -safe point in arbitrary dimension

In the previous section, we showed that if there are at least  $\lceil \frac{N_i}{d+1} \rceil$  adversarial nodes, then an  $F_i$ -safe point does not exist. In this section, we show that for any  $d$ , if the number of adversarial nodes  $F_i$  is less than  $\lfloor \frac{N_i}{d+1} \rfloor$ , then an  $F_i$ -safe point always exists. The main idea will be to show that for a set  $S$  of  $N_i$  points in  $\mathbb{R}^d$ , there exists an interior point of depth  $\lfloor \frac{N_i}{d+1} + 1 \rfloor$ . In other words, there exists an interior point in the intersection of all convex sets containing at least  $\lfloor \frac{N_i}{d+1} + 1 \rfloor$  points of  $S$ . For this, we first prove some intermediate results (proof is available in Abbas et al., 2021).

**Lemma 4.5.** *Let  $S$  be a set of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ . For a set of  $d + 1$  convex sets, each containing at least  $\lceil \frac{Nd}{d+1} + 1 \rceil$  points from  $S$ , there are at least  $d + 1$  points of  $S$  contained in the intersection of all halfspaces.*

The above lemma implies that any  $d + 1$  convex sets, each of which contains enough points from  $S$ , must contain at least  $d + 1$  points from the same set. Now, if the original point set  $S$  is in general position, then so must be the  $d + 1$  points contained in all convex sets. Furthermore, every  $d + 1$  points in general positions span a  $d$ -simplex that contains a ball of radius  $\epsilon$  in its interior for sufficiently small value of  $\epsilon > 0$ . Thus, we get the following:

**Corollary 4.6.** *Let  $S$  be a set of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ . For a set of  $d + 1$  convex sets each containing at least  $\lceil \frac{Nd}{d+1} + 1 \rceil$  points from  $S$ , there exists  $\epsilon > 0$  such that the intersection of all convex sets contains a ball of radius  $\epsilon$ .*

Next, we state the following version of Helly's Theorem (Klee, 1953) that will be used in the proof of Theorem 4.8.

**Theorem 4.7 (Klee, 1953).** *If  $\mathcal{K}$  is a finite family of convex sets (or an infinite family of compact convex sets) in  $\mathbb{R}^d$ ,  $C$  is a convex set such that for every  $d + 1$  members of  $\mathcal{K}$ , there exists a translate<sup>5</sup> of  $C$  contained in their intersection, then there is a suitable translate of  $C$  contained in the intersection of all sets in  $\mathcal{K}$ .*

Now we state a crucial result to prove a sufficient condition for an  $F_i$ -safe point in arbitrary  $d$ . The proofs of the following results are available in Abbas et al. (2021).

**Theorem 4.8.** *For any set  $S$  of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ , there exists an interior point in the intersection of all convex sets containing at least  $\lceil \frac{Nd}{d+1} + 1 \rceil$  points of  $S$ .*

Next, we present the main result in this subsection, that is, a sufficient condition for the existence of an  $F_i$ -safe point in an arbitrary dimension  $d$ .

**Theorem 4.9.** *For a set of  $N_i$  nodes in general positions in  $\mathbb{R}^d$ , if the number of adversarial nodes is less than  $\lfloor \frac{N_i}{d+1} \rfloor$ , then there always exists an  $F_i$ -safe point.*

### 4.4. Improvement in the resilience

If a normal agent  $i$  uses approximate Tverberg partition (Mulzer & Werner, 2013) to compute an  $F_i$ -safe point, as is done in Park and Hutchinson (2017), then the algorithm is resilient to  $F_i \leq \frac{N_i}{2^d} - 1$  adversaries in  $\mathcal{N}_i$  as compared to the best possible (theoretical) bound  $F_i \leq \frac{N_i - 1}{d + 1}$ . Here, we discuss the usefulness of centerpoint in computing an  $F_i$ -safe point to improve the ADRC algorithm's

<sup>5</sup> For a set  $C \subset \mathbb{R}^d$ , a translate of  $C$  is the set  $\{c + x : c \in C\}$  for some vector  $x \in \mathbb{R}^d$ .

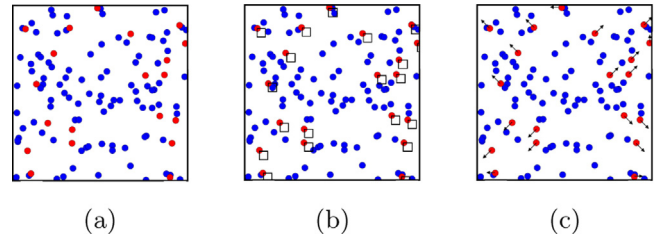


Fig. 4. Robots' initial positions with (a) stationary, (b) oscillating and (c) move-away adversaries.

resilience. The main point is that we can compute a centerpoint exactly in two and three dimensions and can compute it with better approximation (compared to Tverberg partition) in higher dimensions using known algorithms. Next, we state the resilience of the centerpoint-based computation of an  $F_i$ -safe point.

**Proposition 4.10.** *Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$ , of which  $F_i < \mathcal{F}_{max}$  are adversarial, then it is possible to compute an  $F_i$ -safe point using centerpoint if*

$$\mathcal{F}_{max} = \left\lfloor \frac{N_i}{d+1} \right\rfloor, \quad \text{for } d = 2, 3, \text{ and} \quad (4)$$

$$\mathcal{F}_{max} = \Omega\left(\frac{N_i}{d^2}\right) \quad \text{for } d > 3.$$

Moreover, such an  $F_i$ -safe point can be computed in  $O(N)$  and  $O(N^2)$  times in  $d = 2$  and  $3$ , respectively, and in  $O(N^{c \log d} (2d)^d)$  time in  $d > 3$  dimensions.

**Remark 1.** By comparing (3) and (4), observe that centerpoint-based computation of  $F_i$ -safe point improves the resilience of ADRC algorithm as compared to the Tverberg partition-based computation. Using centerpoint, we achieve optimal resilience in two and three dimensions, whereas in higher dimensions, the resilience improves from  $\Omega(N_i/2^d)$  (based on Tverberg partition) to  $\Omega(N_i/d^2)$  adversaries in the neighborhood of a normal agent  $i$ .

**Remark 2.** The point returned by the algorithm by Miller and Sheehy (2010) has a centerpoint depth of  $\Omega\left(\frac{N_i}{d^{r/r-1}}\right)$  for any positive integer  $r > 1$ . For instance, if we consider  $r = 2$ , then we get a point of depth  $\Omega(N_i/d^2)$ . By increasing  $r$ , the quality of approximation, and hence the bound on the number of adversarial agents improves and approaches  $\Omega(N_i/d)$ . However, it comes at the cost of increasing time complexity as the runtime of the algorithm is  $O(N_i^{c \log d} (rd)^d)$  for an integer  $r > 1$  (Miller & Sheehy, 2010). Clarkson et al. have proposed an algorithm in Clarkson, Eppstein, Miller, Sturivant, and Teng (1996) that computes a point of depth  $\Omega(N_i/d^2)$  with high probability in time  $O(\log^2 d \log \log N_i)$ .

### 4.5. Comparison of the centerpoint-based computation of an $F_i$ -safe point with other approaches

In Mendes et al. (2015, Section 6.2), a linear program (LP) is presented to compute an  $F_i$ -safe point, where  $F_i < N_i/(d + 1)$ . The proposed LP has a total of  $\binom{N_i}{N_i - F_i} (d + 1 - N_i - F_i)$  constraints in  $d + \binom{N_i}{N_i - F_i}$  variables. Hence, their algorithm's time complexity is at least exponential in  $F_i$ . Since adversarial agents can be as many as a fraction of the total agents, the LP complexity in Mendes et al. (2015) is exponential in  $N_i$ . Consequently, as  $F_i$  grows with  $N_i$ , the algorithm's time complexity is too high to be practical.

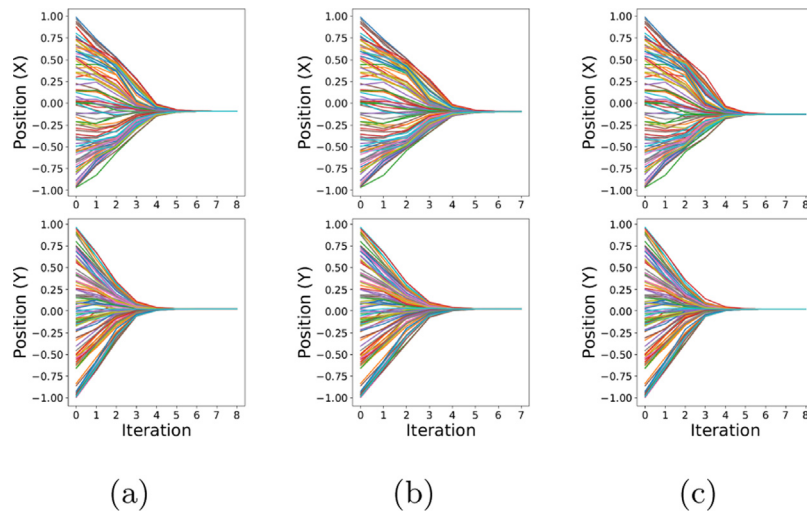


Fig. 5. Normal robots' positions as functions of iterations in (a) stationary, (b) oscillating, and (c) move-away adversaries.

We note that a centerpoint can also be computed exactly thus, achieving the same resilience in  $O(N_i^{d-1})$  time for  $d \geq 3$  using a better LP method proposed in Chan (2004). This is a significant improvement over the LP in Mendes et al. (2015) because the time complexity is exponential in  $d$  instead of the number of agents. Nevertheless, even the algorithm in Chan (2004) becomes impractical in higher dimensions. Therefore, we propose that an approximate centerpoint be computed in higher dimensions using Miller and Sheehy (2010), resulting in the resilience of  $\Omega(N_i/d^2)$ .

In Wang, Mou, and Sundaram (2019), a quadratic program is presented to solve the resilient convex combination problem to compute an  $F_i$ -safe point. The complexity of the algorithm (Wang et al., 2019, Section 3.2) is  $O(d(N_i r)^3)$ , which is linear in  $d$ . Here, the parameter  $r$  is defined (in Section 3.1 of their paper) as  $r = \binom{N_i - \sigma}{N_i - \sigma - F_i}$ . Assuming  $\sigma = 0$ ,<sup>6</sup> the complexity of computing an  $F_i$ -safe point using the quadratic program is  $O(d(N_i^{N_i+1})^3)$ , where a fraction of the agents are allowed to be adversarial. As a result, the time complexity of the quadratic program in Wang et al. (2019) is worse than the LP proposed in Mendes et al. (2015). Similarly, in Yan et al. (2020, Section IV) authors present an algorithm to compute an  $F_i$ -safe point (that they call a “middle point”) in time  $O((pr)^3)$ , where  $p = dF_i + 1$ , and  $r = \binom{(d+1)F_i+1}{F_i}$ . Note that the parameter  $r$  is exponential in the number of adversarial agents.

In summary, the centerpoint-based approach provides fundamental limits on the computation of an  $F_i$ -safe point. An immediate result of this approach is that in general, computation of an  $F_i$ -safe point, where the maximum value of  $F_i$  is  $\Omega(N_i/d)$ , is a challenging problem. Since checking whether a given point is a centerpoint or not is *co-NP-Complete* (Teng, 1991), we deduce that checking whether a point is an  $F_i$ -safe point or not in general is also *co-NP-Complete*.

## 5. Numerical evaluation

We perform simulations to illustrate resilient consensus in multirobot systems in two dimensions using centerpoint, and compare it with the one using approximate Tverberg partition (Park & Hutchinson, 2017).<sup>7</sup> We model interconnections between

robots using the following graphs: (1) A disk graph  $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}(t))$ , in which each node  $i \in \mathcal{V}$  (representing a robot) has a sensing radius  $r$  and  $(j, i) \in \mathcal{E}(t)$  if and only if  $\|x_j(t) - x_i(t)\| \leq r$ . (2) A fixed undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  whose edge set does not change over time.

At each iteration  $t$  of the consensus algorithm, a normal robot  $i$  computes an  $F_i$ -safe point  $s_i(t)$  of its neighbors' positions (using centerpoint or approximate Tverberg partition), and calculates its new position using (1). In our experiments, we set  $\alpha_i(t) = 0.8$ , for all  $t$ .

### 5.1. Centerpoint based resilient consensus

We simulate a group of 120 robots, out of which 100 are normal and 20 are adversarial. They are deployed in a planar region  $\mathcal{W} = [-1, 1] \times [-1, 1] \in \mathbb{R}^2$  as shown in Fig. 4(a), where nodes in blue are normal robots executing consensus and nodes in red represent adversarial robots. We consider three types of adversarial behaviors:

- *Stationary* – each red node has a fixed position that does not change throughout the simulation.
- *Oscillating* – each red node changes its position by moving from corner to corner within the square of length 0.1, as depicted in Fig. 4(b).
- *Move-away* – each red node moves towards the closest corner of the region  $\mathcal{W}$  from its initial position and then stays there, as shown in Fig. 4(c).

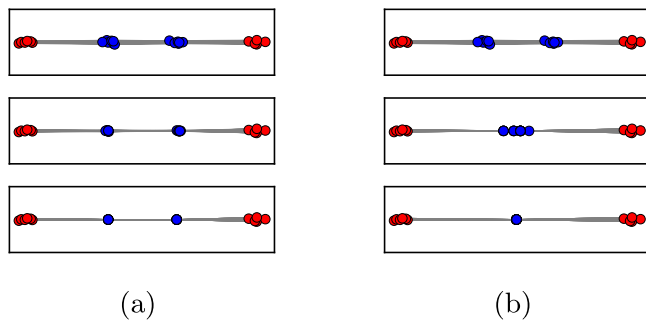
The interconnection topology is captured by a disk graph with a fixed sensing radius of  $r = 0.45$  for every normal robot. Each normal robot executes consensus algorithm based on centerpoint. Consensus is guaranteed if the number of adversarial robots in the neighborhood of each normal robot  $i$  is  $F_i \leq \left(\left\lfloor \frac{N_i}{3} \right\rfloor - 1\right)$  and the sequence of graphs induced by normal agents is repeatedly reachable. Both of these conditions are satisfied in all the simulations here. Consequently, normal robots achieve consensus in the presence of all three types of adversarial nodes, as shown in Fig. 5.

### 5.2. Comparison of approximate Tverberg partition and centerpoint-based resilient consensus

To demonstrate the improved resilience of centerpoint-based consensus compared to the Tverberg partition-based one, we

<sup>6</sup> This is so because in our setting, a normal agent is unaware of the identities of the normal and adversarial agents in its neighborhood.

<sup>7</sup> Our code is available at <https://github.com/JianiLi/MultiRobotsRendezvous>.



**Fig. 6.** The evolution of robots' positions (from top to down) in the case of (a) approximate Tverberg partition-based and the (b) centerpoint-based algorithms.

consider stationary adversarial nodes and assume that the network graphs are fixed. In our first example, we consider a group of 28 robots deployed in a planar region  $\mathcal{W} = [-1.5, 1.5] \times [-0.375, 0.375] \in \mathbb{R}^2$  as shown in Fig. 6. There are six adversarial nodes (red) on the left and six on the planar region's right side. At the same time, 16 normal robots (blue) are divided equally into two clusters in the middle part of the region, each containing eight robots. Moreover, each normal robot is adjacent to six adversarial robots located on its side. Figs. 6 shows the evolution of robots' positions using approximate Tverberg partition-based and centerpoint-based consensus algorithms. As illustrated, the approximate Tverberg partition-based algorithm fails to make the normal robots converge to one point, but the centerpoint-based algorithm succeeds.

## 6. Conclusion

We discussed the merits of using the notion of centerpoint for resilient vector consensus. For a normal agent  $i$ , we provided a geometric characterization – using centerpoint – of states that are guaranteed to lie in the convex hull of agent  $i$ 's normal neighbors' states when the number of adversarial agents is limited to a  $1/(d+1)$  fraction of the size of the neighborhood of  $i$ . It also followed that the upper bound on the number of adversarial agents is best possible in the worst case. For higher dimensions, by using the approximate centerpoint algorithm proposed in Miller and Sheehy (2010), we showed that the resilience of vector consensus algorithm to adversarial agents in the neighborhood of a normal agent  $i$  could be improved from  $\Omega(N_i/2^d)$ , which is due to approximate Tverberg partition (Park & Hutchinson, 2017), to  $\Omega(N_i/d^2)$ . The centerpoint-based aggregation of data in the presence of adversarial agents can be used in a various other applications, including resilient distributed diffusion and learning.

## References

- Abbas, W., Laszka, A., & Koutsoukos, X. (2018). Improving network connectivity and robustness using trusted nodes with application to resilient consensus. *IEEE Transactions on Control of Network Systems*, 5, 2036–2048.
- Abbas, W., Shabbir, M., Li, J., & Koutsoukos, X. (2021). Resilient distributed vector consensus in high dimensions using centerpoint. [arXiv:2003.05497](https://arxiv.org/abs/2003.05497).
- Chan, T. M. (2004). An optimal randomized algorithm for maximum tukey depth. In *Proceedings of the 15th annual ACM-SIAM symposium on discrete algorithms* (pp. 430–436). SIAM.
- Clarkson, K. L., Eppstein, D., Miller, G. L., Sturtivant, C., & Teng, S.-H. (1996). Approximating center points with iterative radon points. *International Journal of Computational Geometry & Applications*, 6(3), 357–377.
- Har-Peled, S., & Zhou, T. (2020). Improved approximation algorithms for tverberg partitions. [arXiv preprint arXiv:2007.08717](https://arxiv.org/abs/2007.08717).
- Klee, V. (1953). The critical set of a convex body. *American Journal of Mathematics*, 75(1), 178–188.

- LeBlanc, H. J., Zhang, H., Koutsoukos, X., & Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4), 766–781.
- Li, J., Abbas, W., Shabbir, M., & Koutsoukos (2020). Resilient distributed diffusion for multi-robot systems using centerpoint. In *Robotics: science and systems*.
- Matoušek, J. (2002). *Lectures on discrete geometry*. Springer.
- Mendes, H., Herlihy, M., Vaidya, N., & Garg, V. K. (2015). Multidimensional agreement in Byzantine systems. *Distributed Computing*, 28(6), 423–441.
- Miller, G. L., & Sheehy, D. R. (2010). Approximate centerpoints with proofs. *Computational Geometry*, 43(8), 647–654.
- Mulzer, W., & Werner, D. (2013). Approximating Tverberg points in linear time for any fixed dimension. *Discrete & Computational Geometry*, 50(2), 520–535.
- Park, H., & Hutchinson, S. A. (2017). Fault-tolerant rendezvous of multirobot systems. *IEEE Transactions on Robotics*, 33, 565–582.
- Reay, J. R. (1968). An extension of Radon's theorem. *Illinois Journal of Mathematics*, 12(2), 184–189.
- Roudneff, J.-P. (2009). New cases of Reay's conjecture on partitions of points into simplices with  $k$ -dimensional intersection. *European Journal of Combinatorics*, 30(8).
- Sayed, A. H., Tu, S.-Y., Chen, J., Zhao, X., & Towfic, Z. J. (2013). Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior. *IEEE Signal Processing Magazine*, 30(3), 155–171.
- Shabbir, M., Li, J., Abbas, W., & Koutsoukos, X. (2020). Resilient vector consensus in multi-agent networks using centerpoints. In *American control conference* (pp. 4387–4392).
- Su, L., & Vaidya, N. (2016). Multi-agent optimization in the presence of Byzantine adversaries: Fundamental limits. In *2016 American control conference* (pp. 7183–7188).
- Sundaram, S., & Ghahesifard, B. (2015). Consensus-based distributed optimization with malicious nodes. In *53rd Annual allerton conference on communication, control, and computing*.
- Sundaram, S., & Ghahesifard, B. (2018). Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3), 1063–1076.
- Teng, S.-H. (1991). *Points, sphere, and separators: A unified geometric approach to graph partitioning* (Ph.D. thesis), Carnegie Mellon University: School of Computer Science.
- Tseng, L., & Vaidya, N. (2013). Iterative approximate Byzantine consensus under a generalized fault model. In *International conference on distributed computing and networking* (pp. 72–86). Springer.
- Vaidya, N. H. (2014). Iterative Byzantine vector consensus in incomplete graphs. In *International conference on distributed computing and networking* (pp. 14–28).
- Vaidya, N. H., & Garg, V. K. (2013). Byzantine vector consensus in complete graphs. In *ACM Symposium on principles of distributed computing* (pp. 65–73).
- Wang, X., Mou, S., & Sundaram, S. (2019). A resilient convex combination for consensus-based distributed algorithms. *Numerical Algebra, Control and Optimization*, 9(3), 269–281.
- Yan, J., Li, X., Mo, Y., & Wen, C. (2020). Resilient multi-dimensional consensus in adversarial environment. [arXiv preprint arXiv:2001.00937](https://arxiv.org/abs/2001.00937).
- Yan, J., Mo, Y., Li, X., & Wen, C. (2019). A “safe kernel” approach for resilient multi-dimensional consensus. [arXiv preprint arXiv:1911.10836](https://arxiv.org/abs/1911.10836).



**Waseem Abbas** is an Assistant Professor in the Systems Engineering Department at the University of Texas at Dallas. Previously, he was a Research Assistant Professor in the Electrical Engineering and Computer Science Department at the Vanderbilt University, Nashville, TN. He received Ph.D. and M.Sc. degrees, both in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA, and was a Fulbright scholar from 2009 till 2013. His research interests are in the areas of resilience and security of network control systems, cyber-physical systems, and graph-theoretic methods in complex networks.



**Mudassir Shabbir** is an Associate Professor in the Department of Computer Science at the Information Technology University, Lahore, and a Research Assistant Professor with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He received his Ph.D. from Rutgers University, NJ USA in 2014. Previously, Mudassir has worked at LUMS, Pakistan, Los Alamos National Labs, NM, Bloomberg L.P. New York, NY, and at Rutgers University. He was Rutgers Honors Fellow for 2011–12. His main area of research is Algorithmic and Discrete Geometry and has

developed new methods for the characterization and computation of succinct representations of large data sets with applications in nonparametric statistical analysis.



**Jiani Li** is currently working as a Research Scientist at Facebook. She received her Ph.D. degree from the Electrical Engineering and Computer Science Department, Vanderbilt University, Nashville, TN, USA in 2021. Her research focuses on resilient multi-agent distributed systems and resilient design of cyber-physical systems with machine learning components.



**Xenofon Koutsoukos** is the Chair of the Department of Computer Science at Vanderbilt University. He is a Professor of Computer Science, Computer Engineering, and Electrical Engineering and a Senior Research Scientist with the Institute for Software Integrated Systems. His research is in the area of cyber-physical systems with emphasis on learning-enabled systems, security and resilience, control and decision-making, diagnosis and fault tolerance, formal methods, and adaptive resource management. Prof. Koutsoukos was the recipient of the NSF Career Award in 2004, the Excellence in Teaching

Award in 2009 from the Vanderbilt University School of Engineering, and the 2011 NASA Aeronautics Research Mission Directorate (ARMD) Associate Administrator (AA) Award in Technology and Innovation. He is a Fellow of the IEEE.