2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Resilient Consensus of Double-Integrator Multi-Agent Systems

Seyed Mehran Dibaji and Hideaki Ishii

Abstract— **We propose an algorithm for consensus of second-order sampled-data multi-agent systems in the presence of misbehaving agents. Each normal agent updates its values (position and velocity) following a predetermined control law based on local information obtained through directed interaction while some malicious agents make updates arbitrarily. The normal agents do not know the global topology of the network, but have prior knowledge on the maximum number of malicious ones. Under the assumption that the network topology has sufficient connectivity in terms of robustness, we develop a resilient algorithm where each agent ignores the neighbors which have large and small position values to avoid being influenced by the malicious agents. The effectiveness of the approach is demonstrated through a numerical example.**

I. Introduction

Recently, in the area of networked control systems, consideration of cyber security has become important since such systems are nowadays more often connected to general purpose networks such as the Internet and wireless communication. Malicious attacks can lead such systems to hazardous operating regions and might physically cause irreparable harms and losses. Safe distributed algorithms in the presence of faulty behaviors and adversarial agents have been widely studied in control [3,4] as well as computer science [7,13].

In this paper, we consider networks of agents which cooperate by interacting with each other to accomplish a global objective. In such systems, malicious intruders may take control of some agents and influence other agents to keep them from completing their planned tasks without being noticed. Here, we consider consensus, one of the essential problems in multi-agent systems. The objective in consensus is agreement on some state values within the collection of agents [8,10].

Resilient algorithms for multi-agent systems have appeared in the literature and can be classified into two approaches. One is to achieve consensus among the non-faulty agents by detecting and isolating malicious agents in the network. In the works of [9,12], techniques of observers for systems with unknown inputs are developed for a consensus problem. The paper [11] also deals with observer-based methods for fault detection when the agents have second-order dynamics. The other approach aims at consensus without finding which agents might be faulty by simply ignoring suspicious agents whether or not they are truly faulty. The work [1] is limited to the complete graph case. On the other hand, the series of papers [6,14] introduces a novel notion of graph robustness to characterize the necessary structure in the underlying network. It is noted that these works study only the case where the agent dynamics is represented as a single integrator.

Here, we focus on resilient consensus of sampled-data double-integrator multi-agent systems in the presence of malicious agents. Consensus problems for second-order agent dynamics have been studied, for example, in [2, 10, 15]. Following the second approach mentioned above, we propose a new algorithm to tackle the problem. The difficulty of this problem is twofold: (i) The presence of malicious agents which might try to deviate the network not to reach consensus and (ii) more complicated dynamics due to the double-integrator agents, which requires agreement in both position and velocity values. In our strategy, the non-faulty normal agents are equipped with an algorithm to collect the neighbors' (relative) positions, but to ignore a certain number of them. Specifically, they leave out those that take large and small values in their updates. In this way, these agents can avoid being affected by the suspicious ones in the course of arriving at consensus. We show that the notion of graph robustness from [6, 14] indeed plays an important role to guarantee sufficient connectivity among the agents.

The outline of this paper is as follows. In Section II, we present preliminary material and then formulate the resilient consensus problem for agents with second-order dynamics. In Section III, the proposed algorithm called DP-MSR is presented along with derivations necessary for further analysis. Section IV is devoted to introducing the concept of robust graphs. Section V is the main part of the paper, where a sufficient condition and a necessary condition are derived on the network topology to reach resilient consensus. In Section VI, we illustrate the effectiveness of the DP-MSR algorithm through a numerical example. Finally, Section VII concludes the paper.

II. Problem Formulation

In this section, we first provide some notations related to graphs and then introduce the problem setting considered in the paper.

A. Graph Related Notions

Given a network of n agents $(n > 1)$, we use a directed graph (or a digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the interaction network among agents, where $\mathcal{V} = \{1, \dots, n\}$ denotes the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ stands for the edge set. The edge $(j, i) \in \mathcal{E}$ indicates that node i can receive information from

The authors are with the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama, 226-8502, Japan. E-mails: dibaji@sc.dis.titech.ac.jp, ishii@dis.titech.ac.jp.

This work was supported in part by the JST-CREST Program.

978-1-4799-3274-0/$31.00 ©2014 AACC

node j. For node i, the set of neighbors is given by $\mathcal{N}_i = \{j : (j,i) \in \mathcal{E}\}$, and its degree is denoted by $d_i = |\mathcal{N}_i|$.

Let $A = [a_{ij}]$ and $L = [l_{ij}]$ be the adjacency and Laplacian matrices of $\mathcal{G}$, respectively. The adjacency matrix is given by $a_{ij} \in (0,1)$ if $(j,i) \in \mathcal{E}$ and otherwise, $a_{ij} = 0$ with $\sum_{j=1,j\neq i}^{n} a_{ij} < 1$. Then, the entries of the Laplacian matrix are defined as $l_{ii} = \sum_{j=1,j\neq i}^{n} a_{ij}$ and $l_{ij} = -a_{ij}, i \neq j$. It is easily seen that the summation of the elements of each row in the Laplacian matrix is zero. Let $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E}_0)$ be the undirected graph obtained from $\mathcal{G}$ by removing the direction of its edges.

A path between node v_1 and v_p is a sequence of $(v_1, v_2, \ldots, v_p)$ in which v_i and v_{i+1} are neighbors. In the undirected graph $\mathcal{G}_0$, if there is a path between each pair of nodes the graph is said to be connected. The connectivity (or the vertex connectivity) $K(\mathcal{G}_0)$ of the graph $\mathcal{G}_0$ is the minimum number of vertices whose removal makes $\mathcal{G}_0$ disconnected. The graph is said to be κ-connected, if $K(\mathcal{G}_0) \geqslant \kappa$. The digraph $\mathcal{G}$ is κ-connected if its undirected version $\mathcal{G}_0$ is κ-connected. A directed graph is said to have a directed spanning tree if there is a node from which there is a path to every other node in the graph.

B. Second-Order Consensus Protocol

Consider the network represented by the directed graph $\mathcal{G}$. Each agent i has a double-integrator dynamics given by

$$\begin{aligned} \dot{r_i}(t) &= v_i(t), \\ \dot{v}_i(t) &= u_i(t), \qquad i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $r_i(t) \in \mathbb{R}$ and $v_i(t) \in \mathbb{R}$ are the position and the velocity of the ith agent at time t, respectively, and $u_i(t)$ is the control input applied to the ith agent [10]. We study the discretized version of the system. After discretization with sampling period T [5], the system (1) can be represented by

$$\begin{aligned} r_i[k+1] &= r_i[k] + Tv_i[k] + \frac{T^2}{2}u_i[k], \\ v_i[k+1] &= v_i[k] + Tu_i[k], \qquad i = 1, \ldots, n, \end{aligned} \tag{2}$$

where $r_i[k]$, $v_i[k]$ and $u_i[k]$ are, respectively, the position, the velocity and the control of the ith agent at $t = kT$. Throughout the paper, we assume that at the initial time, the agents are at rest, i.e., $v_i[0] = 0$ for all i.

At each time step k, the agents update their positions and velocities based upon the time-varying topology of the graph $\mathcal{G}[k]$ at the time, which is a subgraph of $\mathcal{G}$, and is specified later. In particular, for each agent, the control law is based on the relative positions with its neighbors and the velocity of itself:

$$u_i[k] = -\sum_{j=1}^{n} a_{ij}[k][(r_i[k]-\delta_i)-(r_j[k]-\delta_j)]-\alpha v_i[k], \tag{3}$$

where $a_{ij}[k]$ is the (i,j) entry of the adjacency matrix $A[k] \in \mathbb{R}^{n\times n}$ corresponding to $\mathcal{G}[k]$, α is a positive scalar, and $\delta_i \in \mathbb{R}$ is a constant representing the desired position of agent i in a formation. Let $\Delta_{ij} = \delta_i - \delta_j$. The objective of the networked agents is consensus in the sense that the agents come to formation and then to stop asymptotically:

$$r_i[k] - r_j[k] \to \Delta_{ij}, \quad v_i[k] \to 0 \text{ as } k \to \infty.$$

In [2], it is shown that under the control law (3), by properly choosing α, T, and κ, consensus can be obtained. κ has to be chosen such that for any nonnegative integer k_0, the union of $\mathcal{G}[k]$ across $k \in [k_0, k_0+\kappa]$ has a directed spanning tree.

In the real world, however, when some agents are attacked, fail, or are affected by external disturbances, they may not follow the predetermined updating rules (2) and (3). In the next subsections, we introduce necessary definitions and then formulate the resilient consensus problem in the presence of malicious agents.

Finally, we represent the network system in a vector form. Let $\hat{r}_i[k] = r_i[k] - \delta_i$, $\hat{r}[k] = [\hat{r}_1[k] \ \cdots \ \hat{r}_n[k]]^T$, and $v[k] = [v_1[k] \ \cdots \ v_n[k]]^T$. Then, the system in (2) is written as

$$\begin{aligned} \hat{r}[k+1] &= \hat{r}[k] + Tv[k] + \frac{T^2}{2}u[k], \\ v[k+1] &= v[k] + Tu[k], \end{aligned} \tag{4}$$

and the control law (3) is written as

$$u[k] = -L_k\hat{r}[k] - \alpha v[k], \tag{5}$$

where L_k is the Laplacian matrix for the graph $\mathcal{G}[k]$.

C. Resilient Consensus

We classify the agents as normal and malicious based on the following definition [6].

Definition 1 (Normal/Malicious node): Node i is called normal if it applies a predetermined updating rule. Otherwise it is called malicious, i.e., if it does not apply the given updating rule and updates its state arbitrarily. The set of indices of malicious nodes is denoted by $\mathcal{M} \subset \mathcal{V}$.

Having malicious nodes in the network is a threat to achieve the objective of consensus because they might be able to separate the normal agents into more than two groups. We analyze the possible effects of those misbehaving agents and design algorithms that can avoid such effects of misbehaving agents.

We assume that an upper bound is available for the number of misbehaving agents in the network [6].

Definition 2 (f-local model): The graph $\mathcal{G}$ is f-local if for each normal agent i, the number of malicious nodes in its neighbor set $\mathcal{N}_i$ is at most f, that is, $|\mathcal{N}_i \cap \mathcal{M}| \leqslant f$, $\forall i \in \mathcal{V} \setminus \mathcal{M}$.

Note that under this model, there may be in total more than f malicious nodes in the network.

We now introduce the notion of consensus for the network of second-order agents in the presence of misbehaving agents. To this end, based on definition in [6], we introduce the notion of resilient consensus for the model.

Definition 3 (Resilient consensus): Under the f-local model, the normal nodes are said to achieve resilient consensus if both of the following conditions are satisfied for any initial position values and zero initial velocities.

1) Agreement condition: There exists $c \in \mathbb{R}$ such that

$$\lim_{k\to\infty} \hat{r}_i[k] = c, \quad \lim_{k\to\infty} v_i[k] = 0, \quad \forall i \in \mathcal{V}\backslash\mathcal{M},$$

2) Safety condition:

$$\hat{r}_i[k] \in [m, M], \quad \forall i \in \mathcal{V}\backslash\mathcal{M}, \quad \forall k \in \mathbb{Z}_+,$$

where m and M are, respectively, the minimum and the maximum of the initial values for the positions $r_i[0]$, $i \in \mathcal{V}\backslash\mathcal{M}$ of the agents.

In other words, the normal agents reach consensus if finally they stop in formation specified by δ_i and moreover, lie within the interval given by initial positions.

III. Algorithm for Resilient Consensus

In this section, we outline the proposed algorithm for achieving consensus in the presence of misbehaving agents. Normal agents following the control law (3) above from [2] can be easily manipulated by malicious agents that do not use the same law. To facilitate the control law (3) to be resilient, we establish a new algorithm to be applied to normal agents.

A. DP-MSR Algorithm

The proposed algorithm is called DP-MSR, which stands for Double-Integrator Position-Based Mean Subsequence Reduced Algorithm. The algorithm has three steps as follows:

1) At each time step k, each normal node i receives the position values of its neighbors, and sorts them from the largest to the smallest.
2) If there are less than f agents which have position values strictly larger than $r_i[k]$, then the normal node i ignores the incoming edges from those nodes. Otherwise, it ignores the incoming edges from f agents which have the largest position values. Similarly, if there are less than f agents which have position values strictly smaller than $r_i[k]$, then node i ignores all incoming edges from these nodes. Otherwise, it ignores the f incoming edges from the nodes which have the smallest position values.
3) Apply the control input (3) by substituting $a_{ij}[k] = 0$ for edges (j, i) which are ignored in step 2.

This algorithm was inspired from W-MSR in [6,14], which is applied for the single-integrator case.

In the algorithm above, each normal node may ignore up to $2f$ incoming edges: At most f edges from neighbors whose positions are large, and f edges from neighbors whose positions are small. The underlying graph $\mathcal{G}[k]$ is determined by the edges that are not ignored by the agents at each time k. The adjacency matrix A_k and the Laplacian matrix L_k are determined accordingly.

B. Discretized Second-Order Dynamics

According to the model of malicious nodes considered, the difference between normal agents and malicious agents lies in their control inputs: For normal agents, it is given by (3) while for the malicious agents, it is arbitrary. On the other hand, the position and velocity dynamics for all nodes remain the same.

In what follows, we introduce a more detailed mathematical model for the system with such malicious nodes. For ease of notation, we reorder the indices of the agents. Since the number of malicious agents is $|\mathcal{M}|$, let the normal agents take indices $1, \dots, n - |\mathcal{M}|$ and let the malicious agents be $n - |\mathcal{M}| + 1, \dots, n$. The position, velocity, and control vectors are now partitioned into the normal and malicious parts as

$$\hat{r}[k] = \begin{bmatrix} \hat{r}^N[k] \\ \hat{r}^M[k] \end{bmatrix}, \quad v[k] = \begin{bmatrix} v^N[k] \\ v^M[k] \end{bmatrix}, \quad u[k] = \begin{bmatrix} u^N[k] \\ u^M[k] \end{bmatrix}, \tag{6}$$

where the superscript N stands for normal and M for malicious.

According to the control law (3) and the malicious agent model, the control inputs $u^N[k]$ and $u^M[k]$ can be written as

$$\begin{aligned} &u^N[k] = -L_k^N \hat{r}[k] - \alpha \begin{bmatrix} I & 0 \end{bmatrix} v[k], \\ &u^M[k] : \text{arbitrary}, \end{aligned} \tag{7}$$

where L_k^N is a matrix formed by the rows of the Laplacian L_k associated with normal agents, and it has the size of $(n - |\mathcal{M}|) \times n$. Note that, as in the case of the Laplacian matrix L_k, the row sums of this matrix L_k^N are zero.

By substituting these control inputs into (4), and by using the vectors in (6), the overall system can be written as follows:

$$\begin{aligned} \begin{bmatrix} \hat{r}[k+1] \\ v[k+1] \end{bmatrix} &= \begin{bmatrix} I_n - \frac{T^2}{2}\begin{bmatrix} L_k^N \\ 0 \end{bmatrix} & TI_n - \frac{\alpha T^2}{2}\begin{bmatrix} I_{n-|\mathcal{M}|} & 0 \\ 0 & 0 \end{bmatrix} \\ -T\begin{bmatrix} L_k \\ 0 \end{bmatrix} & I_n - \alpha T \begin{bmatrix} I_{n-|\mathcal{M}|} & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \hat{r}[k] \\ v[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T^2}{2} I_{|\mathcal{M}|} \\ 0 \\ TI_{|\mathcal{M}|} \end{bmatrix} u^M[k], \end{aligned} \tag{8}$$

where the partitioning in the submatrices (such as $\begin{bmatrix} (L_k^N)^T & 0 \end{bmatrix}^T$) is in accordance with the vectors in (6).

For the sampling period T and the parameter α, we assume

$$T \leq 1, \ 1 + \frac{T^2}{2} \leq \alpha T \leq 2 - \frac{T^2}{2}. \tag{9}$$

Based on this assumption, we obtain the following lemma, which plays a key role in the analyses later.

Lemma 1: Under the control inputs (7), the position vector $\hat{r}$ of the agents can be expressed as

$$\hat{r}[k+1] = \begin{cases} \left(I_n - \frac{T^2}{2}\begin{bmatrix} L_0 \\ 0 \end{bmatrix} \right) \hat{r}[0] + \begin{bmatrix} 0 \\ \frac{T^2}{2} I_{|\mathcal{M}|} \end{bmatrix} u^M[0], \ \text{if } k = 0, \\ \begin{bmatrix} \Phi_{1k} & \Phi_{2k} \end{bmatrix} \begin{bmatrix} \hat{r}[k] \\ \hat{r}[k-1] \end{bmatrix} \\ \quad + \begin{bmatrix} 0 \\ \frac{T^2}{2} I_{|\mathcal{M}|} \end{bmatrix} u^M[k] + \begin{bmatrix} 0 \\ \frac{T^2}{2} I_{|\mathcal{M}|} \end{bmatrix} u^M[k-1], \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise,} \end{cases}$$

where

$$\Phi_{1k} = 2I_n - \alpha T \begin{bmatrix} I_{n-|\mathcal{M}|} & 0 \\ 0 & 0 \end{bmatrix} - \frac{T^2}{2} \begin{bmatrix} L_k^N \\ 0 \end{bmatrix},$$
$$\Phi_{2k} = -I_n + \alpha T \begin{bmatrix} I_{n-|\mathcal{M}|} & 0 \\ 0 & 0 \end{bmatrix} - \frac{T^2}{2} \begin{bmatrix} L_{k-1}^N \\ 0 \end{bmatrix}.$$

Moreover, the summation of each row in the matrix $\begin{bmatrix} I_{n-|\mathcal{M}|} & 0 \end{bmatrix} - \frac{T^2}{2} L_0$ and that of each row in the first $n - |\mathcal{M}|$ rows of $\begin{bmatrix} \Phi_{1k} & \Phi_{2k} \end{bmatrix}$ are one, and all entries of these matrices are nonnegative.

This lemma can be proved relatively easily by some algebraic manipulations. Since the controls $u^M[k]$, or $u^M[k-1]$ do not directly enter, the following remark is an immediate conclusion of the lemma above.

Remark 1: The position vector $\hat{r}^N[1]$ of the normal agents at time $k = 1$ is a convex combination of the initial positions $\hat{r}[0]$. Furthermore, for $k \geqslant 1$, the position vector $\hat{r}^N[k+1]$ is making update based on a convex combination of the current position $\hat{r}[k]$ and that from the previous time step $\hat{r}[k-1]$, if α, T fulfill the conditions (9).

IV. Robust Graphs

In this section, we demonstrate that looking at local malicious model from the viewpoint of the traditional concept of graph connectivity is not sufficient. Then, the notion of robust graphs will be formally introduced.

Proposition 1: Under the f-local model, there exists a network with connectivity $\kappa = \lfloor n/2 \rfloor + f - 1$ such that the normal agents using the DP-MSR algorithm with parameter $2f$ cannot achieve resilient consensus.

The proof of this proposition is left out owing to page limitations.

Based on the implication of this proposition, we need a new connectivity measure to deal with the problem of resilient consensus. We use the notion of robust graphs, which was first proposed in [6] to analyze resilient consensus of single-integrator multi-agent systems.

Definition 4 (r-Reachable set): In a digraph $\mathcal{G}$, a nonempty subset $\mathcal{S}$ of nodes is r-reachable if there exists a node i which has at least r neighbors outside $\mathcal{S}$. In other words, $\mathcal{S}$ is r-reachable if $\exists i \in \mathcal{S} : |\mathcal{N}_i \backslash \mathcal{S}| \geqslant r$.

Definition 5 (r-Robust graph): A digraph $\mathcal{G}$ is r-robust if for every pair $\mathcal{A}$ and $\mathcal{B}$ of nonempty disjoint subsets of nodes, at least one of the subsets is r-reachable. In other words, $\forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{V} :\ \mathcal{A} \cap \mathcal{B} = \varnothing \Rightarrow \mathcal{A}$ and/or $\mathcal{B}$ is r-reachable.

The concept of r-robust graph seems to be complicated at first glance. The following lemma helps to have a better intuition about such graphs [6].

Lemma 2: Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an r-robust network. Then it has the following properties:

(i) $r \leqslant \lceil n/2 \rceil$. Also, if $\mathcal{G}$ is a complete graph, then it is r'-robust for all $0 < r' \leqslant \lceil n/2 \rceil$.
(ii) $\mathcal{G}$ is at least r-connected.
(iii) Any subgraph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ of $\mathcal{G}$ where at most k incoming edges to each node has been removed is $(r-k)$-robust.
(iv) $\mathcal{G}$ has a directed spanning tree.
(v) The graph $\mathcal{G}' = (\mathcal{V} \cup \{v_0\}, \mathcal{E} \cup \mathcal{E}_0)$, where v_0 is a vertex added to $\mathcal{G}$ and $\mathcal{E}_0$ is the edge set related to v_0, is r-robust if $d_{v_0} \geqslant r$.

We observe that by (ii) of the lemma, robustness is a stronger concept than the traditional connectivity metric.

V. Resilient Consensus by DP-MSR

In this section, a sufficient condition and a necessary condition for resilient consensus in the second-order multi-agent system are studied.

We now state the main result, which is a sufficient condition for resilient consensus by DP-MSR.

Theorem 1: Under the f-local model, if the network is $(2f+1)$-robust, then the normal agents using the DP-MSR algorithm with parameter $2f$ achieve resilient consensus.

Proof: First we show that the safety condition is satisfied, i.e., each normal agent keeps its position value within the interval $[m, M]$, where m and M are the minimum and the maximum of the initial position values of the normal agents. According to Lemma 1 and Remark 1, at time $k = 1$, the position values of normal agents is a convex combination of the initial position values. Hence, the position values do not exceed the interval $[m, M]$. We separate two cases: All malicious agents have initial position value outside the interval $[m, M]$ or some of them are outside this interval. In both cases, in step 2 of DP-MSR, all malicious agents that are outside $[m, M]$ are ignored. As a result, the position values of the normal agents at time $k = 1$ are within the interval $[m, M]$. For time $k \geqslant 2$, we can follow a similar argument and arrive at the conclusion that the positions of the normal nodes remain within the interval $[m, M]$.

Next, we show agreement among the positions. We introduce two functions as follows:

$$\overline{r}[k] = \max\left(\hat{r}^N[k], \hat{r}^N[k-1]\right),$$
$$\underline{r}[k] = \min\left(\hat{r}^N[k], \hat{r}^N[k-1]\right),$$

where the maximum and the minimum are taken with respect to all entries of the vectors. We claim that $\overline{r}[k]$ is a nonincreasing function. Again by Lemma 1 and Remark 1, we have

$$\begin{aligned} \overline{r}[k] &= \max\left(\hat{r}^N[k], \hat{r}^N[k-1]\right) \\ &\leqslant \max\left(\hat{r}^N[k-1], \hat{r}^N[k-2]\right) = \overline{r}[k-1]. \end{aligned}$$

Similarly, we can show that the function $\underline{r}[k]$ is nondecreasing.

Since the functions $\overline{r}[k]$ and $\underline{r}[k]$ are monotone and bounded by m and M, respectively, they have limits. Denote the limits of $\overline{r}[k]$ and $\underline{r}[k]$ by A_M and A_m, respectively. We claim that the limits coincide as $A_M = A_m$, i.e., the positions of the normal agents come to consensus. The proof is by contradiction. Assume that $A_M > A_m$. Given a sequence $\{\epsilon_i\}$ of positive numbers, let

$$\mathcal{X}_M(k, \epsilon_i) = \{i \in \mathcal{V} \backslash \mathcal{M} : r_i[k] > A_M - \epsilon_i\},$$
$$\mathcal{X}_m(k, \epsilon_i) = \{i \in \mathcal{V} \backslash \mathcal{M} : r_i[k] < A_m + \epsilon_i\}.$$

We show that there exists such a sequence that makes one of these sets to be empty in a finite number of steps, which is a contradiction to the definition of the limits A_M and A_m.

First, take $\epsilon_0 > 0$ and $0 < \epsilon < \epsilon_0$ such that

$$\begin{aligned}&A_M - \epsilon_0 > A_m + \epsilon_0,\\ &\exists k_\epsilon :\ \forall k \geqslant k_\epsilon,\ \overline{r}[k] < A_M + \epsilon,\ \underline{r}[k] > A_m - \epsilon\end{aligned}$$

(by definition of convergence, such an ϵ exists). Because the graph is $(2f+1)$-robust, between the two disjoint sets $\mathcal{X}_M(k_\epsilon, \epsilon_0)$ and $\mathcal{X}_m(k_\epsilon, \epsilon_0)$, one of them is $(2f+1)$-reachable. Suppose that $\mathcal{X}_M(k_\epsilon, \epsilon_0)$ is $(2f+1)$-reachable. Then, there is a node i which has $2f+1$ neighbors outside $\mathcal{X}_M(k_\epsilon, \epsilon_0)$. Since there are at most f malicious neighbors for this normal node, there are at least $f+1$ normal neighbors outside of the set $\mathcal{X}_M(k_\epsilon, \epsilon_0)$. By DP-MSR, one of the normal neighbors has to be used for updating. By definition, an upper bound on the position of this normal neighbor is $A_M - \epsilon_0$. From Remark 1, every normal agent updates by a convex combination of position values of current and previous time steps. Let β be the minimum element of all possible cases of $\begin{bmatrix}\Phi_{1k} & \Phi_{2k}\end{bmatrix}$ (for all k). By the assumptions on α and T in (9), we have $\beta \in (0,1)$. Then, for node i, it holds that

$$\begin{aligned}r_i^N[k_\epsilon+1] &\leqslant (1-\beta)\overline{r}[k_\epsilon] + \beta(A_M - \epsilon_0)\\ &\leqslant (1-\beta)(A_M+\epsilon) + \beta(A_M - \epsilon_0)\\ &\leqslant A_M - \beta\epsilon_0 + (1-\beta)\epsilon. \qquad (10)\end{aligned}$$

By choosing a sufficiently small ϵ, $\beta\epsilon_0 - (1-\beta)\epsilon$ is positive. Let $\epsilon_1 = \beta\epsilon_0 - (1-\beta)\epsilon$. We can conclude that if $\mathcal{X}_M(k_\epsilon, \epsilon_0)$ is $(2f+1)$-reachable, then there exists a normal node i which goes outside of this set at the next update, which is given by $\mathcal{X}_M(k_\epsilon+1, \epsilon_1)$.

Further, we prove that there is no node that goes inside the set $\mathcal{X}_M(k_\epsilon+1, \epsilon_1)$ from the outside of $\mathcal{X}_M(k_\epsilon, \epsilon_0)$. Suppose that node j has position value $\hat{r}_j^N[k] \leqslant A_M - \epsilon_0$. Then, the position of node j at the next time can be bounded as

$$\hat{r}_j^N[k_\epsilon+1] \leqslant (1-\beta)\overline{r}[k_\epsilon] + \beta(A_M - \epsilon_0).$$

By following steps similar to those in (10), the maximum possible value for $\hat{r}_j^N[k+1]$ is bounded by $A_M - \beta\epsilon_0 + (1-\beta)\epsilon$; this is the same as the bound on $\hat{r}_i^N$.

Hence, the cardinality of the set $\mathcal{X}_M(k_\epsilon+1, \epsilon_1)$ is smaller than that of $\mathcal{X}_M(k_\epsilon, \epsilon_0)$, that is $|\mathcal{X}_M(k_\epsilon+1, \epsilon_1)| \leqslant |\mathcal{X}_M(k_\epsilon, \epsilon_0)|$. Likewise, we can establish that if $\mathcal{X}_m(k_\epsilon, \epsilon_0)$ is $(2f+1)$-reachable, then $|\mathcal{X}_m(k_\epsilon+1, \epsilon_1)| \leqslant |\mathcal{X}_m(k_\epsilon, \epsilon_0)|$. The two sets $\mathcal{X}_M(k_\epsilon+1, \epsilon_1)$ and $\mathcal{X}_m(k_\epsilon+1, \epsilon_1)$ are disjoint. Consequently, analogous steps can be done repeatedly for $k_\epsilon + j$, $j \geqslant 2$ and conclude that either $|\mathcal{X}_M(k_\epsilon+j, \epsilon_j)| \leqslant |\mathcal{X}_M(k_\epsilon+j-1, \epsilon_{j-1})|$ and/or $|\mathcal{X}_m(k_\epsilon+j, \epsilon_j)| \leqslant |\mathcal{X}_m(k_\epsilon+j-1, \epsilon_{j-1})|$, where $\epsilon_j = \beta\epsilon_{j-1} + (1-\beta)\epsilon$.

Since the number of normal agents is finite, in some finite time τ, one of the sets $\mathcal{X}_M(k_\epsilon+\tau, \epsilon_\tau)$ or $\mathcal{X}_m(k_\epsilon+\tau, \epsilon_\tau)$ becomes empty. Note that τ is smaller than the number of normal agents in the network, denoted by S. To guarantee $\epsilon_\tau > 0$, it is sufficient that ϵ is chosen from the interval $(0, \beta^S\epsilon_0/(1-\beta^S))$ because

$$\begin{aligned}0 < \beta^S\epsilon_0 - (1-\beta^S)\epsilon &\leqslant \beta^\tau\epsilon_0 - (1-\beta^\tau)\epsilon\\ &= \beta^\tau\epsilon_0 - (1-\beta)(1+\beta+\cdots+\beta^{\tau-1})\epsilon\\ &= \beta^2\epsilon_{\tau-2} - \beta(1-\beta)\epsilon - (1-\beta)\epsilon\\ &= \beta\epsilon_{\tau-1} - (1-\beta)\epsilon = \epsilon_\tau.\end{aligned}$$

Therefore, we arrive at contradiction and thus it must hold that $A_M = A_m$.

Finally, we show that all normal agents stop in the limit. If there is agreement among positions of normal agents $i \in \mathcal{V}\setminus\mathcal{M}$, using (3), we can write $u_i^N[k] \to -\alpha v_i^N[k]$ as $k \to \infty$. By substitution of this into (2), we have

$$r_i^N[k+1] \to r_i^N[k] + T\left(1 - \frac{\alpha T}{2}\right) v_i^N[k]$$

as $k \to \infty$. By the choice of α and T in (9), it follows that $v_i^N[k] \to 0$ as $k \to \infty$. ■

For the case without malicious agents, it is shown in [2] that consensus can be obtained under the control law (3) if the union of $\mathcal{G}[k]$ across $k \in [k_0, k_0+\kappa]$ has a directed spanning tree for some κ and any nonnegative integer k_0, and by choosing $\alpha T = 3/2$. After applying DP-MSR algorithm, each agent removes k incoming edges at each time. From Lemma 2 (iii) and (iv), $\mathcal{G}[k]$ is 1-robust. Hence, for all k, the graph $\mathcal{G}[k]$ has a directed spanning tree. We can thus conclude that Theorem 1 is consistent with the result of [2].

The following theorem states the necessary condition, which is stated without proof due to space limitation.

Theorem 2: Under the f-local model, if the normal agents using DP-MSR algorithm with parameter $2f$ achieve resilient consensus, then the network is $(f+1)$-robust.

The following proposition exhibits that the sufficient condition in Theorem 1 can not be relaxed.

Proposition 2: There exists a $2f$-robust network under which normal agents cannot achieve resilient consensus by the DP-MSR algorithm.

VI. Numerical Example

In this section, we show through a numerical example that the proposed resilient consensus algorithm is effective.

We consider a complete digraph with five agents. Based on Lemma 2 (i), we can show that this graph is 3-robust. Note that with five agents, the complete graph is the only case to have this robustness property. In this network, agent 1 is set to be malicious ($f = 1$). The initial values for the agents are assigned as below:

$$\hat{r}[0] = \begin{bmatrix}5 & 2 & 10 & 70 & 4\end{bmatrix}^T,\quad v[0] = \begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}^T.$$

In these examples, the sampling period is set as $T = 0.3$, and the parameter α as $\alpha = 1.5/0.3 = 5$.

We made simulations for three cases. In all cases, agent 1 behaves differently, following the trajectory of $\hat{r}_1[k] = 2k$, $k \geqslant 0$. The first case is with the conventional control. In the results of Fig. 1, observe that the normal agents come together, but then start to follow agent 1. This shows that

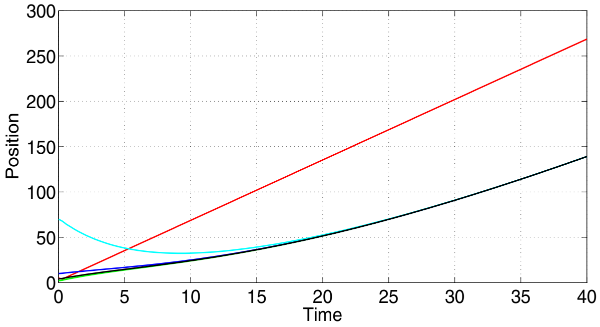

Fig. 1. Time responses of the agent's positions: Conventional method with malicious agent 1 ($\hat{r}_i[k] = 2k$)

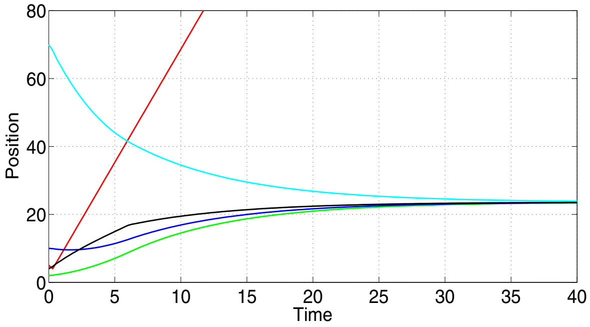

Fig. 2. Time responses of the agent's positions: Proposed method with malicious agent 1 ($\hat{r}_i[k] = 2k$)

the normal agents do not come to stop and moreover go outside of the region specified by the safety condition. Note that agent 1 is playing the role of a leader, and the normal agents are tracking it.

In the second simulation, we applied the proposed control law to the same setup. In Fig. 2, we see that the normal agents are not affected by the malicious one and achieve resilient consensus. The normal agents reach the final position around 23.5. In fact, the trajectories look similar to the case without any misbehaving agent. Thus, we confirm that as Theorem 1 suggests, 3-robust graph is sufficient for the DP-MSR algorithm.

Finally, as the third case, we modified the network structure by removing an edge from the complete graph and ran the proposed algorithm in the presence of the same malicious behavior of agent 1. The graph is in fact no longer 3-robust. As a consequence, in Fig. 3, we confirm that consensus in the agent positions is not attained and there is a certain gap remaining among them at 20.3 and 8.89 though the velocities go to zero.

VII. Conclusion

In this paper, we have considered the problem of resilient consensus for second-order multi-agent systems. We proposed an algorithm for the normal agents to reach consensus which ignores the furthest and nearest agents and applies consensus based on information received from the remaining agents. A necessary condition and a sufficient condition to reach consensus based on this algorithm have been developed. There are two limitations in the consensus protocol studied here. One is that each agent has to start from zero velocity. The other is that the necessary condition and sufficient condition presented in the paper are different. In future works, we will address these issues.

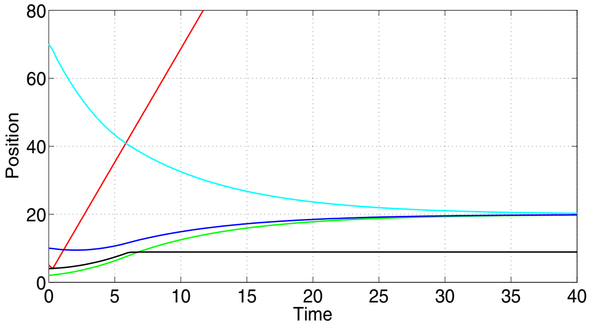

Fig. 3. Time responses of the agent's positions: Proposed method with malicious agent 1 ($\hat{r}_i[k] = 2k$) over a non-robust graph

References

[1] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil, "Optimal Byzantine-resilient convergence in uni-dimensional robot networks," *Theoretical Computer Science*, vol. 411, pp. 3154–3168, 2010.

[2] Y. Cao and W. Ren, "Sampled-data discrete-time coordination algorithms for double integrator dynamics under dynamic directed interaction," *Int. J. Control*, vol. 83, no. 3, pp. 506–515, 2010.

[3] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic, 1999.

[4] S. X. Ding, P. Zhang, Ch. Chihaia, W. Li, Y. Wang, and E. L. Ding, "Advanced design scheme for fault tolerant distributed networked control systems," In *Proc. 17th IFAC World Congress.*, pp. 13569–13574, 2008.

[5] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, Adison Wesley, 2006.

[6] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications,* vol. 31, no. 4, pp. 766–781, 2013.

[7] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann, 1996.

[8] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[9] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[10] W. Ren and R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*, Springer, 2008.

[11] I. Shames, A. M. H. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[12] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

[13] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal of Computing*, vol. 28, pp. 1347–1363, 1999.

[14] H. Zhang and S. Sundaram, "Robustness of complex networks with implications for consensus and contagion," in *Proc. IEEE 51st Conference on Decision and Control*, pp. 3426–3432, 2012.

[15] Y. Zhang, Y. Tian, "Consentability and protocol design of multi-agent systems with stochastic switching topology," *Automatica*, vol. 45, no. 5, pp. 1195–1201, 2009.