Introduction to Multiagent Control and Open Problems Related to a Higher-Order Cascaded Integrator Consensus Algorithm

Bryan Peck
ECCS Department
Ohio Northern University
Ada, Ohio 45810
Email: b-peck.1@onu.edu

Heath LeBlanc
ECCS Department
Ohio Northern University
Ada, Ohio 45810
Email: h-leblanc@onu.edu

Abstract—The control of several devices and systems to perform a group objective is considered multiagent control. It has many applications in unmanned aerial vehicles (UAVs), spacecraft, robotics, and vehicular systems. These strategies enable tasks such as flocking of swarms, group formation, synchronization, rendezvous, and consensus of systems. Distributed control is a strong way to scale these strategies to large systems in an adaptable and reconfigurable manner. Instead of having a "centralized brain" of the system, information locally available to each agent is used through cooperative control algorithms so that each agent works independently to achieve the group objective.

This paper will include a brief introduction to multiagent control and a brief evaluation of a control algorithm proposed by Ren et al. utilizing the state information of agents that may be modeled as a chain of integrators[1]. This agent model and its control algorithm has several potential applications including the planar vertical takeoff and landing (PVTOL) of aircraft and Segway vehicles. Specifically, the control algorithm has the potential to control the takeoffs and landings of swarms of fixed-wing aircrafts and the motion of swarms of Segways.

We present an open problem in the resilience of the consensus algorithms proposed for higher-order integrator agents and demonstrate that they are susceptible to attacks from adversaries. One class of attacks on multiagent systems consists of hijacked agents that attempt to subvert the group objective by acting abnormally or communicating false and even harmful information. Proper resilience to hijacked agents requires that the other agents of the system are still able to achieve the desired task. The addition of resilience would improve the safety and security of the control protocol in real-world applications.

I. Introduction

Multiagent control has many applications in unmanned aerial vehicles (UAVs), spacecraft, robotic, and vehicular systems[1]. It allows for the formation of teams which can accomplish tasks more effectively than any single agent. This teamwork also leads to lowered costs in some situations, such as in spacecraft. For example, it is less expensive to launch several small payloads than one large spacecraft requiring a multiple-payload platform[2]. Another benefit of multiagent control is due to the systems being programmed to respond immediately to environmental factors. This is imperative for UAVs flying outdoors as they experience different wind speeds based on the temperature, season, time, location, nearby structures, and other factors. These factors cannot be predicted beforehand and some vary for each agent in the system. However, the agents in the system can receive enough information from the environment and other agents in order to perform the group objective despite the wind and other external influences. Another advantage of these systems lies in their ability to conduct tasks with minimal supervision once they have been constructed. The many uses of multiagent control have led to continued research to solve complex problems and allow for further implementation in various fields[1-3].

Multiagent control is utilized for a variety of group objectives including flocking, group formation, rendezvous, synchronization, and consensus. Flocking is when a swarm of agents acts similar to a flock of birds, as seen in Figure 1. The agents in a flock act as a unit, stay close, and avoid collisions[3]. Group formation control is when agents form and maintain a pattern or shape, as shown by the airplanes in Figure 2.


Figure 1: Flock of birds[4]


Figure 2: Airplane formation[5]

Rendezvous is the task involving agents meeting at a desired place at the same time[3]. Synchronization and consensus are more general duties that involve matching the states of the agents of a system[1,3]. For example, the airplane formation requires each plane to travel at the same velocity. Therefore, they will work to synchronize their velocity states. Many group objectives and the various algorithms underlying these strategies are discussed further in other papers[2-3,6-10].

A. System architecture

An example of a networked multiagent system is shown in Figure 3. The agents of this system have three main tasks: information gathering, computation, and actuation.

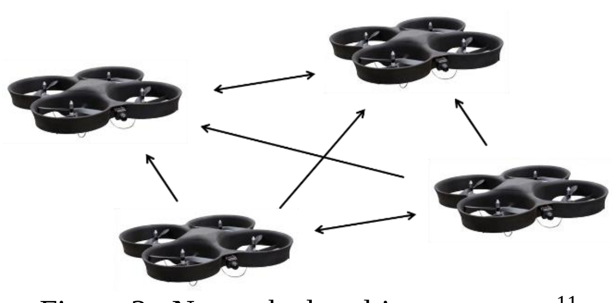
Figure 3: Networked multiagent system[11]

The system consists of four devices, quadrotors in this case, which are connected by a communication network. The information flow is shown by a network of one-way and two-way arrows. The information streams are due to sensing and communication. Agents can sense the environment and other devices using on-board sensors analyzing motion, light, heat, sound, or other parameters. In addition, agents can communicate with each other using radio waves or some other medium. Through sensing and communication, a particular agent gains local information from a subset of the agents in the system; these agents are called its neighbors.

Each agent utilizes coordination algorithms to compute its local information. These algorithms are designed to output the appropriate task that the device should enact based on its particular knowledge of the system.

Agents then must implement their computation-based decisions. This implementation involves transferring the algorithm's output into a physical realization of the decision. For our example, the quadrotor would appropriately turn its motors to move the blades and device as desired.

B. Distributed control

Distributed control is when every device uses its local information and coordination algorithms to make an individual decision on how to proceed in order to support the group objective. Decentralized control is similar but must have strongly connected routing networks where every agent has a communication routing path connecting it to every other agent[12]. This routing path may go through other agents along the way but still provides an effective communication path for transmitting information between distant agents. This allows for decentralized control agents to utilize more than just local information. Figure 4 shows how distributed and decentralized control differ in their information sharing.

Figure 4: Distributed control and decentralized control, respectively

Distributed control also differs from centralized control. Centralized control involves a single leader that instructs each agent on how to proceed, with no information sharing between other agents. These differences in communication and decision strategies are clear in Figure 5.

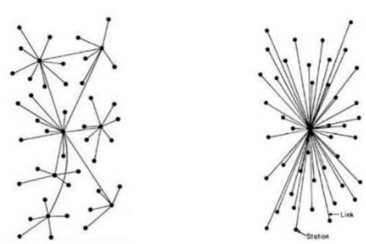

Figure 5: Distributed and centralized communication networks, respectively

There are several benefits to distributed control. It is very scalable and can be quickly translated to work on larger systems. Unlike centralized control, it is not limited by a leader's range or strength capabilities when constructing larger networks. In addition, distributed control approaches are designed to work for time-varying communication networks. This ability to work on time-varying networks is important because communication often varies over time in multiagent systems due to the changing positions of agents in space and with respect to one another.

For distributed control, the coordination algorithms must be constructed to achieve group objectives using only local information. As such, each agent has a subset of the total information of the system and likely knows different information than other agents in the same system but must work towards the same group objective. Another challenge lies in the time-changing network of the system. This time-varying communication causes agents to have different numbers of neighbors and varying information streams at different times. This dynamic system requires more rigor from the coordination algorithms. Yet another challenge in distributed control is that of resilience. Security threats in multiagent systems often occur in the form of a hijacked agent attempting to undermine the group objective. It may do this by sending malicious information or acting aberrantly. An image of a hijacked agent in a system is shown in Figure 6.

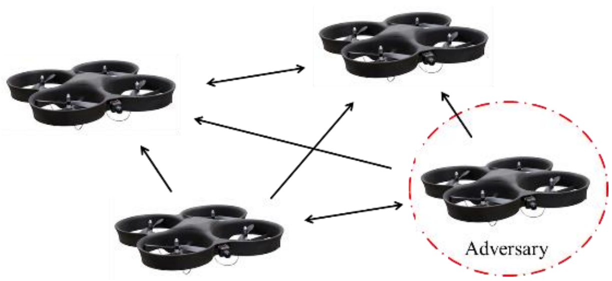

Figure 6: Malicious agent attacking a system[11]

In Figure 6, all three non-hijacked agents are receiving information directly from the adversarial agent. If the adversary moves or acts abnormally, the top two devices will sense this information and use it in their decisions. If it communicates fictitious or malignant data, the other agents will receive malicious information. Despite the potential strength of this attack, there are strategies to maintain resilience in distributed control[13]. If the necessary conditions are met, a multiagent system can still achieve the group objective with its non-hijacked agents.

Distributed control has many applications. Ren et al. mention the rendezvous of mobile autonomous vehicles, stabilization of a formation, maneuvering of a formation, and flocking using distributed protocols[10]. Saldaña et al. discuss a distributed consensus algorithm's application to perimeter surveillance in which robots patrol a boundary[14]. Yu et al. examine high-order consensus algorithms and found conditions that allow for consensus in leader-follower control in multiagent dynamical systems[15]. Wen et al. use distributed control to prove necessary and sufficient conditions causing agents to follow leaders in their system utilizing only observation of the leaders, not communication[16].

The rest of the paper is organized as follows. Section II introduces the mathematical notation and the cascaded integrator model. Section III addresses the contributions of this paper in presenting an open problem. Section IV concludes.

II. Introduction to cascaded integrators

A. Network model

The communication networks of multiagent systems are often modeled using digraphs (directed graphs) or undirected graphs. Digraphs consist of a set $\mathcal{N}$ of nodes to represent the agents and a set $\mathcal{E}$ of edges to represent the communication flow between nodes. This set of edges consists of ordered pairs (v_1, v_2) where $v_2 \in \mathcal{N}$ is receiving information from $v_1 \in \mathcal{N}$. Undirected graphs only consist of two-way information streams and can be described by using unordered pairs. The neighbor set $\mathcal{N}_i$ of a node v_i consists of its neighbors; i.e., $\mathcal{N}_i = \{v_j \in \mathcal{N}\colon (v_j, v_i) \in \mathcal{E}\}$. The number of neighbors, $|\mathcal{N}_i|$, is called the in-degree of v_i. A path is a sequence of nodes in a digraph such that there is an edge between each pair of nodes in the sequence. A digraph is strongly connected if a path exists between every distinct pair of nodes. An undirected graph is connected if there is a path between every distinct pair of nodes. A complete graph has every node receiving information from every other node.

B. Cascaded integrator model

The cascaded integrator model involves a sequence of derivatives of a variable of an agent. These information states are given in (1) for the ℓth-order of a system using Newton's notation for differentiation.

$$\begin{aligned} \dot{\xi}_i^{(0)} &= \xi_i^{(1)} \\ &\vdots \\ \dot{\xi}_i^{(\ell-2)} &= \xi_i^{(\ell-1)} \\ \dot{\xi}_i^{(\ell-1)} &= u_i \quad i \in \{1, \dots, n\} \end{aligned} \tag{1}$$

This system of differential equations is the set of system equations governing each of the n agents of the system, with associated control action u_i where i=1, 2, …, n. The consensus algorithm of such a system desires to have $\xi_i^{(k)} = \xi_j^{(k)}$ for k=0, 1, …, ℓ-1 for every distinct pair of agents i and j. In other words, each agent's state derivatives are desired to match all other agents' associated state derivatives.

C. Ren et al. consensus algorithm

Ren et al. propose an algorithm utilizing this model in which all ℓ derivatives are desired to match and are utilized in the formation of the control law u_i as shown in (2)[1].

$$u_i = -\sum_{j=1}^{n} g_{ij} k_{ij} \left[\sum_{k=0}^{\ell-1} \gamma_k \left(\xi_i^{(k)} - \xi_j^{(k)} \right) \right] \quad i \in \{1, \dots, n\} \tag{2}$$

The system is designed to find the difference in the derivative states k=0, 1, …, ℓ-1 for every distinct pair of agents i and j. The factor g_{ij} indicates whether there is an edge between agent j

and i. Factor $k_{ij} > 0$ is a design parameter, and the γ_k are control gains that must be carefully selected based on the communication structure[1]. Depending on these differences and the values of the three factors, the control law acts in the direction that would limit these differences in order to reach consensus.

D. Potential applications

The planar vertical takeoff and landing (PVTOL) system has the potential for achieving consensus using the Ren et al. consensus algorithm. This system has a strong coupling of its translational and rotational components which makes this possible[17]. A swarm of fixed-wing aircraft could have synchronized takeoffs and landings by controlling their pitch angles, pitch angle rates, horizontal positions, and vertical positions using (2).

The damped inverted pendulum cart may also be controlled with a variant of (2). This system consists of a pendulum connected to a cart on wheels where the pendulum's bob is above the cart (i.e. the pendulum is inverted). This problem is notoriously difficult to control. However, Aguilar-Ibáñez et al. express the system with a chain of integrators that has the basic design of (1) with a few variations[18]. The inverted pendulum is similar to Segways; therefore, control of the damped inverted pendulum cart could translate to control of a Segway.

III. Resilient consensus of higher-order cascaded integrators

An open problem in the field of multiagent control is ensuring resilience in the consensus process of a network of high-order cascaded integrator agents. Many multiagent systems are in danger of attacks and can become dangerous if controlled by an attacker. Several control strategies work to allow non-hijacked agents to accomplish the task despite the presence of malicious agents[13]. Necessary and sufficient conditions for resilience are developed for many control algorithms to address these types of attacks. Adding resilience to the higher-order cascaded integrator consensus algorithm would increase the number of potential applications. Specifically, it would make the algorithm feasible for applications that would be dangerous and impractical without resilience.

A. Preliminary results

In order to demonstrate that the high-order cascaded integrator consensus algorithm of (2) is sensitive to a malicious agent hijacking the network, we first simulate the case where there is no malicious agent. This demonstrates the nominal behavior of the networked system. Then, a single malicious agent is introduced, and it is shown that the network behavior is hijacked by the malicious agent. After demonstrating the need for resilience, we introduce a strategy for resilience. We show that the strategy for resilience does not disrupt the consensus behavior when no malicious agent is present. Finally, we show that the resilient strategy can maintain the desired consensus behavior of the non-hijacked agents when one malicious agent is present.

For each of the simulations, the conditions of Ren et al.[1] are matched closely be setting the gains γ_0, γ_1, and γ_2 to 1, 2, and 3, respectively, as shown in Figures 7-10. The communication network consists of a complete graph of four agents. Plots of ${\xi_i}^{(2)}$ for i=1, …, 4 are shown in Figures 7-

10 with initial conditions set to 2, 10, 0, and -5, respectively. The initial conditions of the zeroth and first derivatives for all agents are set to zero. Figure 7 shows the nominal behavior of the network of high-order cascaded integrator agents using (2) when no malicious agent is present. Observe that each agent's second derivative converges to consensus at a fixed constant value.

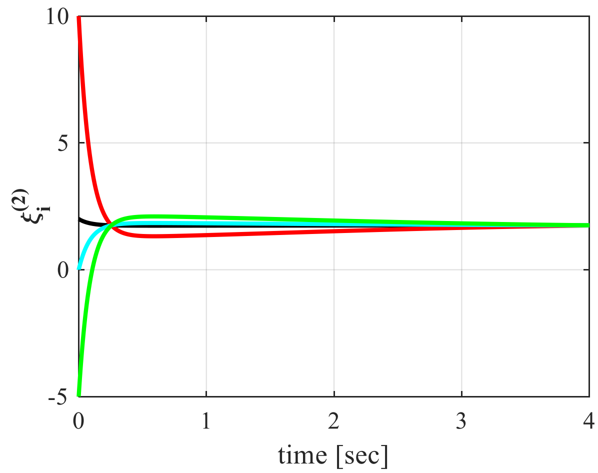

Figure 7: High-order cascaded integrators using (2) with no malicious agent

Figure 8 shows that a network of high-order cascaded integrator agents using (2) is susceptible to just one malicious agent being present. In this case, the control input of the malicious agent is set to $u_1 = 1$, causing its second derivative to increase linearly on a divergent path. The malicious agent hijacks the entire network of agents to follow its divergent trajectory.

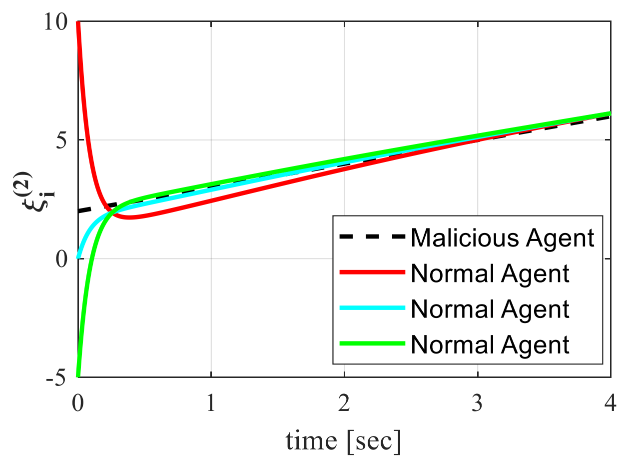

Figure 8: High-order cascaded integrators using (2) with a single malicious agent

In contrast, Figure 9 shows a simulation of the same network with no adversarial agents, but with a strategy for resilience. It employs an approach to resilience that eliminates extreme values relative to the individual's states, making use of a parameter $F \in \{1, 2, \ldots\}$, similar to the Adversarial Robust Consensus Protocol (ARC-P)[12,13]. For each derivative term, the node sorts the received corresponding derivative terms and removes up to F values strictly larger or smaller than its own. The remaining values are used to form the differences shown in (2), applying the same gains, summations, etc. as (2). In Figure 9, the parameter F is set to 1. Observe that the agents achieve consensus to a fixed constant value, similar to Figure 7. Notice, there is a tradeoff between resilience and speed of convergence, as has been shown for single integrator agents[12]. The agents require more time to reach consensus to a fixed constant value.

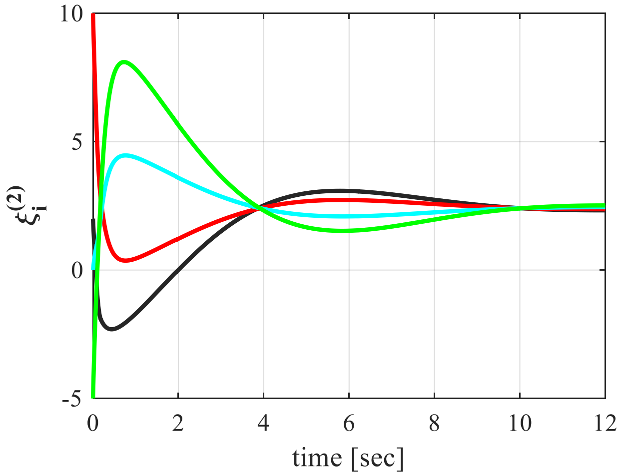

Figure 9: High-order cascaded integrators with resilience and no malicious agent

Figure 10 shows the simulation under the same initial conditions as shown in Figure 9. However, this system has a malicious agent, whose control input is set to $u_1 = -1$. Due to the resilience strategy, the second derivatives of the non-hijacked agents settle and do not diverge with the malicious agent like the result shown in Figure 8. The malicious agent does initially influence the other agents causing consensus to be reached at a lower value than Figure 9. In contrast to the case of single integrator agents[13], the final consensus value may lie outside the range of initial condition values of the normal agents.

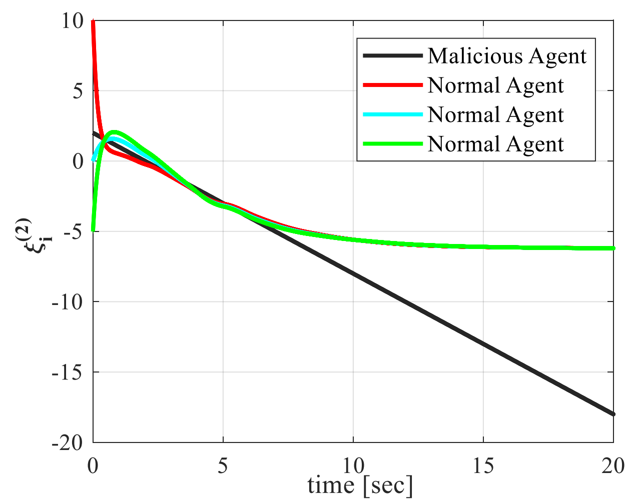

Figure 10: High-order cascaded integrators with resilience and a single malicious agent

IV. Conclusion

Multiagent control uses groups to perform tasks effectively. Distributed control is a subset of multiagent control which only uses the local information received by an agent. This distributed control has many benefits and its various algorithms have been used in many applications. Ren et al. propose a distributed consensus algorithm for high-order cascaded integrator agents to control the high-order derivatives of the agents in the system[1]. This algorithm has potential applications in controlling swarms of fixed-wing aircraft and Segways.

We have demonstrated through simulation that a malicious agent is able to disrupt the consensus employed by Ren et al's algorithm. A resilience strategy has been added to the networked system successfully. The employment of this resilience algorithm is also able to achieve consensus despite a malicious agent. These simulations have been limited to the case of a complete communication graph. The derivation of necessary and sufficient conditions on the communication graph to achieve resilient consensus of high-order cascaded integrator agents is still an open problem. This addition of resilience enables the protocol to be more effective and serve more applications.

Bibliography

1. W. Ren et al., "High-Order and Model Reference Consensus Algorithms in Cooperative Control of MultiVehicle Systems," J. of Dynamic Syst., Measurement, and Control, vol. 129, no. 5, pp. 678-688, Sept., 2007.
2. Scharf et al., "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control," in American Control Conference, Boston, MA, 2004, pp. 2976-2985.
3. K. Oh et al., "A survey of multi-agent formation control," Automatica, vol. 53, pp. 424-440, Mar. 2015.
4. How It Works Team. (2015). Why do birds flock together? [Online]. Available: https://www.howit worksdaily.com/why-do-birds-flock-together/
5. Raytheon. (2015, June 22). Wowing the Crowds at Paris [Online]. Available: https://www.raytheon.com/news /feature/paris_airshow.html
6. Olfati-Saber et al., "Consensus and Cooperation in Networked Multi-Agent Systems," Proceedings of the IEEE, vol. 95, no. 1, pp. 215-233, Jan. 2007.
7. Zhang et al., "Optimal Design for Synchronization of Cooperative Systems: State Feedback, Observer and Output Feedback," IEEE Trans. Autom. Control, vol. 56, no. 8, pp. 1948-1952, Aug., 2011.
8. Scharf et al, "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," in American Control Conference, Denver, CO, 2003.
9. R. W. Beard et al., "A Coordination Architecture for Spacecraft Formation Control," IEEE Trans. Control Syst. Technol., vol. 9, no. 6, pp. 777-790, Nov., 2001.
10. W. Ren et al., "Information Consensus in Multivehicle Cooperative Control," *IEEE Control Syst. Mag.* vol. 27, no. 2, pp. 71-82, April, 2007.
11. D. Hambling. (2009, Dec. 17). Aussie Hovering Drone is Straight Outta Avatar [Online]. Available: https://www.wired.com/2009/12/australian-drone-perches-stares/
12. H. J. LeBlanc, "Resilient Cooperative Control of Networked Multi-Agent Systems," Ph.D. dissertation, Dept. Elect. Eng., Vanderbilt Univ., Nashville, TN, 2012.
13. H. J. LeBlanc and X. Koutsoukos., "Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multi-agent systems," *IEEE Trans. On Control of Network Systems*, 2017.
14. D. Saldaña et al., "Resilient Consensus for Time-Varying Networks of Dynamic Agents," in American Control Conference, Seattle, WA, 2017, pp. 252-258.
15. W. Yu et al., "Distributed Higher Order Consensus Protocols in Multiagent Dynamical Systems," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 8, pp. 1924-1932, Aug., 2011.

16. G. Wen et al., "Containment of Higher-Order Multi-Leader Multi-Agent Systems: A Dynamic Output Approach," IEEE Trans. Autom. Control, vol. 61, no. 4, pp. 1135-1140, April, 2016.
17. Sanahuja et al, "Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system," Int. J. Robust Nonlinear Control, vol. 20, no. 10, pp. 1129-1139, July, 2010.
18. C. Aguilar-Ibáñez et al., "Stabilizing the damped inverted pendulum cart system by means of a cascade chain of integrators," *Int. Conf. Mechatronics, Electron., and Auto. Eng.*, 2013, pp. 146-151.