

Balancing Agility and Communication: Denser Networks Require Faster Agents

YU-MEI HUANG[✉], ARTHUR C. B. DE OLIVEIRA[✉], DINESH MURUGAN[✉],
AND MILAD SIAMI[✉] (Senior Member, IEEE)

Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA

CORRESPONDING AUTHOR: MILAD SIAMI (e-mail: m.siami@northeastern.edu).

This research was supported in part by ONR under Grant N00014-21-1-2431, in part by the NSF under Grants 2121121 and 2208182, and in part by Army Research Laboratory under Grant W911NF-22-2-0001.

This article has supplementary downloadable material available at <https://ieeexplore.ieee.org>, provided by the authors.

This article has supplementary downloadable material available at <https://doi.org/10.1109/OJCSYS.2023.3324274>, provided by the authors.

ABSTRACT This article delves into the challenges of ensuring stability (in some sense) and robustness in large-scale second-order consensus networks (SOCNs) and autonomous vehicle platoons in the discrete-time domain. We propose a graph-theoretic methodology for designing a state feedback law for these systems in a discrete-time framework. By analyzing the behavior of the solutions of the networks based on the algebraic properties of the Laplacian matrices of the underlying graphs and on the value of the update cycle (also known as the time step) of each vehicle, we provide a necessary and sufficient condition for the stability of a linear second-order consensus network in the discrete-time domain. We then perform an $\mathcal{H}_2$ -based robustness analysis to demonstrate the relationship between the $\mathcal{H}_2$ -norm of the system, network size, connectivity, and update cycles, providing insights into how these factors impact the convergence and robustness of the system. A key contribution of this work is the development of a formal framework for understanding the link between an $\mathcal{H}_2$ -based performance measure and the restrictions on the update cycle of the vehicles. Specifically, we show that denser networks (i.e., networks with more communication links) might require faster agents (i.e., smaller update cycles) to outperform or achieve the same level of robustness as sparse networks (i.e., networks with fewer communication links) - see Fig. 1. These findings have important implications for the design and implementation of large-scale consensus networks and autonomous vehicle platoons, highlighting the need for a balance between network density and update cycle speed for optimal performance. We finish the article with results from simulations and experiments that illustrate the effectiveness of the proposed framework in predicting the behavior of vehicle platoons, even for more complex agents with nonlinear dynamics, using Quanser's Qlabs and Qcars.

INDEX TERMS Graph theory, robustness, sparsity, stability consensus networks.

I. INTRODUCTION

Multi-agent systems are groups of autonomous agents that work together to achieve a shared goal through collaboration, feedback, and iteration [1], [2], [3], [4]. These systems have gained significant attention in recent years due to their use in a variety of real-world applications, such as smart power grids [5], vehicle platooning [6], aerial drone displays [7], epidemic networks [8], high-speed satellite internet [9], and the Internet of Things (IoT) [10], which involve complex dynamical networks. Additionally, distributed systems have

been extensively studied in the control community for their numerous applications, from robotics [11], [12], [13] to biological and ecological networks [14], [15], [16].

One major challenge in multi-agent systems is the communication protocol used for information exchange; each agent can share its state while following this protocol. All agents can reach an agreement by designing an appropriate interconnection topology in which agents are only able to receive and interact with their neighbors. Algebraic graph theory is a well-known approach for studying the behavior of multi-agent

dynamical systems and achieving consensus. Necessary and sufficient algebraic conditions for achieving consensus and robustness analysis for first-order and second-order systems in continuous time have been provided in previous work such as [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] for different assumptions and frameworks. In addition, the consensus problem has been studied in the discrete-time in [29], [30], [31], [32], [33], [34], [35]. For example, in [34], the authors consider the problem of guaranteeing consensus for first-order discrete-time linear time-varying networks. In [35] the authors consider the second-order consensus problem for discrete-time networks with uncertain discretization times and allow for different update cycles for each agent in the network.

Previous studies have mainly focused on identical underlying graphs for the first and second integrator, where they share the same information flow between neighbors. In this article, we allow for a different structure of interrelation topology in a second-order dynamical system. In this case, the underlying graph for the agents' positions is different from the underlying graph for the agents' velocities. Based on this interconnection protocol, we study not only second-order consensus networks but also vehicle platoon models, which have received increasing attention in recent years due to their potential contributions to road transportation.

This area of research poses several challenges, such as the impact of delayed agents (i.e., long update cycles) and external disturbances on the stability and performance of the entire network. In [36], [37], the authors examine the robust control of the system when subjected to stochastic disturbances, which can grow and propagate through the information exchange between vehicles. In [38], the stability of a vehicle platoon network with a ring coupling graph and path graph in the presence of time delays is studied. The spacing policy in [39], [40] specifies the desired distance between vehicles to ensure that all vehicles asymptotically track a group of heterogeneous mobiles.

The performance of vehicle platoons, in which each agent must maintain a certain agreement such as common velocity, steering angle, or inter-vehicular spacing, can be degraded by exogenous stochastic disturbances, which justifies the interest in such class of disturbances. In [41], the performance is measured in terms of the $\mathcal{H}_2$ -norm, which captures the notion of coherence, and the connection between performance measures and the scale of a multi-dimensional vehicular formation dynamical network is explored. Another performance measure that quantifies the expected value of the steady-state dispersion has been investigated in [42].

In this article we are interested in understanding the effects of discretization on the stability and robustness of network-based systems. We provide necessary and sufficient conditions for guaranteeing the stability of discrete-time second-order consensus networks (SOCNs) and vehicle platoon models based on the algebraic properties of the Laplacian matrices of the underlying graphs and the update cycles (also known as time steps) of each vehicle. We then access the robustness

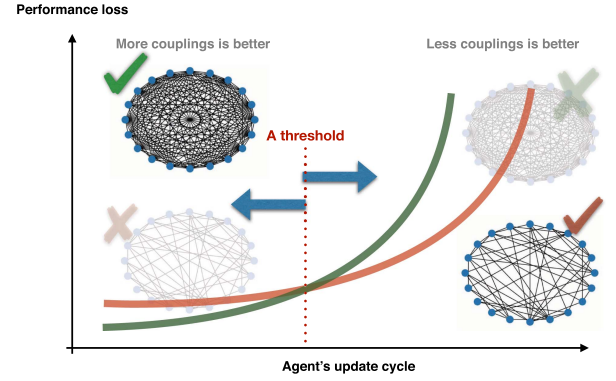


FIGURE 1. This graph illustrates one of the key observations of this article in which denser networks might require faster agents (i.e. smaller update cycles) to outperform less dense networks in terms of robustness. The green curve is the performance loss of a dense network vs. the agent's update cycle and the red curve is the performance loss of a sparse network vs. the agent's update cycle.

of each network through an $\mathcal{H}_2$ -based performance metric and demonstrate the connection between such performance metric, network size, connectivity, and update cycles. Fundamental trade-offs between the $\mathcal{H}_2$ -norm and the restrictions on vehicle update cycles are discussed in Section V and breaking the design intuition derived from continuous-time consensus network literature, as illustrated in Fig. 1. Unlike other studies in this field, our analysis accommodates a more general set of second-order networks. Specifically, we permit distinct position and velocity graph topologies within our theoretical framework, provided they adhere to the assumption of sharing the same set of eigenvectors.

This article extends the preliminary results presented in [43], but includes the full proofs for the first time. The manuscript also presents a more rigorous derivation of our original result, new results, and numerical and experimental examples. Our experiments were derived from a platoon of scaled models of real cars, which introduces nonlinearities and communication noise into the platoon dynamics. Despite the case under consideration introducing certain complexities beyond our initial assumptions, our results still demonstrate the robustness and applicability of our theoretical formulation.

II. PRELIMINARIES AND NOTATIONS

Throughout this article, we use the following notations: $\mathbf{1}_n$ is the n -dimensional vector with all elements equal to one, I_n denotes the $n \times n$ identity matrix, $\mathbf{0}_{m \times n}$ shows the $m \times n$ zero matrix, J_n presents the $n \times n$ matrix of all ones, and $A^\dagger$ represents the pseudo-inverse of matrix A .

All graphs are assumed to be finite, simple, undirected, and connected. We denote an undirected graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, w\}$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \{(i, j) \mid i, j \in \mathcal{V}, i \neq j\}$ is the set of edges, and $w : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_+$ is the weight function with $w(e) = 0$ for all $e = (i, j) \notin \mathcal{E}$. An unweighted graph $\mathcal{G}$ is a graph with a weight function $w(e) = 1$ for $e \in \mathcal{E}$, and zero otherwise. The adjacency matrix $A = [a_{ij}]$ of graph $\mathcal{G}$ is defined by setting $a_{ij} = w(e)$.

The Laplacian matrix $\mathcal{L}$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ is defined as $\mathcal{L} = D - A$ where A is the adjacency matrix defined as above and $D = \text{diag}(d_i)$ with $d_i = \sum_{j, (i,j) \in \mathcal{E}} w((i, j))$ being the degree of node i . If $\mathcal{G}$ is connected and undirected then its Laplacian has only one zero eigenvalue with multiplicity one and associated eigenvector as the vector of all ones $\mathbf{1}_n$.

III. MODELING AND STABILITY ANALYSIS

To facilitate the analysis, we refer to the first integrator state as position states (x), and to the second integrator state as velocity states (v). The reduced-order model of these dynamical networks can be expressed using two state variables of each agent: the i -th vehicle's position x_i and the i -th vehicle's velocity v_i for $i \in \mathcal{V}$. We start by designing the feedback law based on the communication topologies of the consensus networks.

A. SECOND-ORDER CONSENSUS NETWORK (SOCN)

We can write the dynamics of a collection of n independent (disconnected) second-order agents in the discrete-time as:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} I & \gamma I \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \gamma I \end{bmatrix} \bar{u}(k), \quad (1)$$

where γ is the agent's update cycle or the time step, and $\bar{u}_i(k) = u_i(k) + d_i(k)$ is the i -th element of vector $\bar{u}(k) = [\bar{u}_1(k), \bar{u}_2(k), \dots, \bar{u}_n(k)]^\top$, and quantifies external influences over the i -th agent, composed of the cooperative feedback of the network $u_i(k)$ and possible disturbance signals $d_i(k)$.

Of primary interest for this article is a class of disturbances characterized by Gaussian noises, which are prevalent due to their practical implications. Let our system be influenced by a continuous zero mean Gaussian noise. The discretization effects of such noise become crucial in our model formulation and can be achieved using the increments of the Brownian motion, $\mathfrak{B}(t)$. For Gaussian noise with zero mean and unitary covariance, the first-order discretization at the k -th timestep is denoted as

$$d(k) = \frac{\mathfrak{B}(t + \gamma) - \mathfrak{B}(t)}{\gamma},$$

where γ is the time interval between discrete steps, and the difference $\mathfrak{B}(t + \gamma) - \mathfrak{B}(t)$ represents the change in the Brownian motion over that interval, effectively capturing the Gaussian noise experienced by the system during the period γ . From stochastic systems literature [44] it is established that

$$\mathfrak{B}(t + \gamma) - \mathfrak{B}(t) \sim \mathcal{N}(0, \gamma)$$

indicating it follows a Gaussian distribution with zero mean and covariance γ . Given the unitary covariance of the original disturbance, post-discretization, the disturbance signal adheres to a Gaussian distribution with zero mean and covariance $1/\gamma$. With this information we can rewrite $d(k)$ as

$$d(k) = \frac{1}{\sqrt{\gamma}} \xi(k),$$

where $\xi \sim \mathcal{N}(0, \mathbf{1}_n)$.

Next we look at the proposed cooperative feedback law, which can be understood as virtually imposed springs and dampers between the agents. This interaction induces an underlying graph for both position and velocity, which are given by $\mathcal{G}_x = (\mathcal{V}_x, \mathcal{E}_x, F)$ and $\mathcal{G}_v = (\mathcal{V}_v, \mathcal{E}_v, G)$, respectively. The proposed feedback law with underlying graphs $\mathcal{G}_x$ and $\mathcal{G}_v$ is, then, as follows:

$$u(k) = -\mathcal{L}_x x(k) - \mathcal{L}_v v(k), \quad (2)$$

where $\mathcal{L}_x$ and $\mathcal{L}_v$ are the Laplacian matrices of $\mathcal{G}_x$ and $\mathcal{G}_v$ respectively.

The state space of this second-order discrete-time consensus system, where all agents collaborate based on the shared information from their neighbors, can then be rewritten as:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} I & \gamma I \\ -\gamma \mathcal{L}_x & I - \gamma \mathcal{L}_v \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \sqrt{\gamma} I \end{bmatrix} \xi(k), \quad (3)$$

which is the model structure considered for our robustness analysis of SOCNs. Notice that if we refer to the undisturbed SOCN dynamics along the text, we are assuming $\xi(k) \equiv 0$, otherwise $\xi(k) \sim \mathcal{N}(0, \mathbf{1}_n)$.

B. VEHICLE PLATOONS

Next, we extend the idea of cooperative feedback in SOCNs to formulate a vehicle platoon model. We make the following basic assumption for all vehicle platoon systems in this article:

Assumption 1: All vehicles in the platoon have access to their own states; that is, x_i and v_i are always available for computing $\dot{x}_i$ and $\dot{v}_i$, independently from the interconnection topology under consideration.

Consider having all vehicles in the platoon follow a desired trajectory while driving at a desired constant speed of v^d and maintaining certain spacing Δ between each other. The desired trajectory x^d for i -th vehicle is:

$$x_i^d(k) := k\gamma v^d + i\Delta. \quad (4)$$

The position/velocity deviations from the desired trajectory of agent i are defined as

$$\tilde{x}_i := x_i - x_i^d, \text{ and } \tilde{v}_i := v_i - v_i^d.$$

For each vehicle/node i , we can compose the feedback control u_i that satisfies the goal of following the desired trajectory x_i^d at a desired moving speed of v^d while keeping a desired distance from its neighbors into a vector $u = [u_1; u_2; \dots; u_n]$ which is given by:

$$u(k) = -(I + \mathcal{L}_x)\tilde{x}(k) - (I + \mathcal{L}_v)\tilde{v}(k). \quad (5)$$

The vehicle platoons system in state space can be written as:

$$\begin{bmatrix} \tilde{x}(k+1) \\ \tilde{v}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} I & \gamma I \\ -\gamma(I + \mathcal{L}_x) & I - \gamma(I + \mathcal{L}_v) \end{bmatrix}}_{A_{vp}} \begin{bmatrix} \tilde{x}(k) \\ \tilde{v}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \gamma I \end{bmatrix} d(k), \quad (6)$$

where $\xi(\cdot)$ is the external input/disturbance vector. Similarly to the SOCN, if we refer to (6) as the undisturbed platoon dynamics we are assuming $\xi(k) \equiv 0$. Also, notice that the same analysis done for SOCNs under continuous Gaussian disturbances holds here.

C. STABILITY ANALYSIS OF SOCNs

It is shown in [17] that for undisturbed continuous-time SOCNs with the same connected underlying graphs for position and velocity (i.e., $\mathcal{G}_x = \mathcal{G}_v$), all agents in the system will asymptotically reach an agreement. When considering these dynamical networks in the discrete-time framework, the update cycle γ starts to play an important role in the stability of the system. The undisturbed SOCNs (3) and vehicle platoons (6) can be unstable or fragile with inappropriate choices for the update cycle (i.e., γ), therefore, in this article, we first investigate the conditions on γ that guarantee marginally stable systems. As a starting point, we state the same consensus definition as the one presented in [17], but for discrete-time SOCN:

Definition 1 ([17]): An undisturbed discrete-time SOCN with dynamics as presented in (3) with $\xi(k) \equiv 0$ is said to have reached consensus if and only if, for any initial condition, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_i(k) - x_j(k)\| &= 0, \text{ and} \\ \lim_{k \rightarrow \infty} \|v_i(k) - v_j(k)\| &= 0, \end{aligned} \quad (7)$$

for all pairs of agents $i, j = 1, 2, \dots, n$.

We then present the following condition for a discrete-time SOCN to reach consensus:

Lemma 1: The undisturbed discrete-time SOCN with dynamics as presented in (3) reaches consensus for any initial condition if and only if its eigenvalue with the largest magnitude has algebraic multiplicity two and lies on the unitary circle in $\mathbb{C}$.

The proof of this Lemma follows very closely to the proof of Lemma 2 of [17]. First one can notice that if the position and velocity graphs are connected then the resulting A_{SOCN} will always have at least one zero eigenvalue with an algebraic multiplicity of two. For sufficiency we apply the Jordan block decomposition of A_{SOCN} to the limit $\lim_{k \rightarrow \infty} [x(k); v(k)] = A_{SOCN}^k [x(0); v(0)]$ and observe that a consensus is reached. The proof of necessity also follows the same line of argument as in [17]. Specifically, A_{SOCN}^k converges to a matrix of rank higher than two if the eigenvalue at the unit circle has an algebraic multiplicity greater than two, or if there are additional distinct eigenvalues on the unit disc. This implies that our solution converges to a subset of the state-space of dimension larger than two, contradicting the assumption that a second-order consensus is guaranteed to be reached.

Differently from the results for continuous time, however, the magnitude of the eigenvalues of A_{SOCN} depend not only on the graph Laplacians $\mathcal{L}_x$ and $\mathcal{L}_v$, but also on the update cycle

γ of the system. In the following lemma, we present necessary and sufficient conditions for a second-order consensus network (SOCN) to reach consensus. This result provides a quantitative method for evaluating the stability of SOCN in a discrete-time framework.

Lemma 2: For a given undisturbed discrete-time second-order consensus network (3), let the Laplacian matrices of the position and velocity graphs share the same set of eigenvectors, and let both graphs be connected. Then, the discrete-time SOCN is guaranteed to reach a consensus for any initial condition if and only if for all $i = 2, 3, \dots, n$

$$\begin{cases} 0 < \gamma < \frac{\lambda_i^{(v)}}{\lambda_i^{(x)}}, \text{ if } \left(\lambda_i^{(v)}\right)^2 - 4\lambda_i^{(x)} < 0 \\ 0 < \gamma < \frac{4}{\lambda_i^{(v)} + \sqrt{\left(\lambda_i^{(v)}\right)^2 - 4\lambda_i^{(x)}}}, \text{ otherwise} \end{cases} \quad (8)$$

where $\lambda_i^{(x)}$ and $\lambda_i^{(v)}$ are the nonzero eigenvalues of $\mathcal{L}_x$ and $\mathcal{L}_v$, that share the same eigenvectors.

Proof: In order to find the eigenvalues of the state matrix A_{SOCN} , let $\det(A_{SOCN} - \mu I) = 0$, then we have:

$$\begin{aligned} \det \left(\begin{bmatrix} (1-\mu)I & \gamma I \\ -\gamma \mathcal{L}_x & (1-\mu)I - \gamma \mathcal{L}_v \end{bmatrix} \right) \\ = \det \left((1-\mu)^2 I - \gamma(1-\mu)\mathcal{L}_v + \gamma^2 \mathcal{L}_x \right) \\ = \prod_{i=1}^n \left((1-\mu)^2 - \gamma(1-\mu)\lambda_i^{(v)} + \gamma^2 \lambda_i^{(x)} \right) = 0, \end{aligned} \quad (9)$$

where $\lambda_i^{(x)}$ is the i -th eigenvalue of $\mathcal{L}_x$ and $\lambda_i^{(v)}$ is the i -th eigenvalue of $\mathcal{L}_v$, respectively and following the same order for the basis of eigenvectors.

By solving the quadratic equation:

$$(1-\mu)^2 - \gamma(1-\mu)\lambda_i^{(v)} + \gamma^2 \lambda_i^{(x)} = 0, \quad (10)$$

one can find the eigenvalues of matrix A :

$$\mu_{i\pm} = 1 - \frac{\gamma \lambda_i^{(v)}}{2} \pm \frac{\gamma \sqrt{\left(\lambda_i^{(v)}\right)^2 - 4\lambda_i^{(x)}}}{2}. \quad (11)$$

From Lemma 1, $\mu_{i\pm}$ should be inside a unit disk, i.e., $\|\mu_{i\pm}\| < 1$ and the update cycle $\gamma \in \mathbb{R}_+$. Also, since we assumed the graphs are connected, the Laplacian matrices $\mathcal{L}_{x,v}$ have a simple zero eigenvalue $\lambda_1^{(x,v)} = 0$ associated with the eigenvector of all ones $v_1^{(x,v)} = 1$ and all the other eigenvalues have positive real parts. If $(\lambda_i^{(v)})^2 - 4\lambda_i^{(x)} < 0$, for $i = 2, 3, \dots, n$, then we know $\mu_{i\pm} \in \mathbb{C}$. Let $j = \sqrt{-1}$ and $\Re(\mu_{i\pm}) = 1 - \frac{\gamma \lambda_i^{(v)}}{2}$, $\Im(\mu_{i\pm}) = \pm \frac{\gamma \sqrt{4\lambda_i^{(x)} - (\lambda_i^{(v)})^2}}{2}$, s.t. $\|\mu_{i\pm}\| < 1$.

$$\|\mu_{i\pm}\|^2 = \left\| 1 - \frac{\gamma \lambda_i^{(v)}}{2} \pm \frac{\gamma \sqrt{4\lambda_i^{(x)} - (\lambda_i^{(v)})^2}}{2} j \right\|^2$$

$$\begin{aligned}
 &= \left(1 - \frac{\gamma \lambda_i^{(v)}}{2}\right)^2 + \left(\frac{\gamma \sqrt{4\lambda_i^{(x)} - (\lambda_i^{(v)})^2}}{2}\right)^2 \\
 &= 1 - \gamma \lambda_i^{(v)} + \gamma^2 \lambda_i^{(x)} < 1.
 \end{aligned} \tag{12}$$

Then, we get $-\gamma \lambda_i^{(v)} + \gamma^2 \lambda_i^{(x)} < 0$; thus, it follows that

$$0 < \gamma < \frac{\lambda_i^{(v)}}{\lambda_i^{(x)}}. \tag{13}$$

If $(\lambda_i^{(v)})^2 - 4\lambda_i^{(x)} \geq 0$, we have $\|\mu_{i\pm}\| = |1 - \frac{\gamma \lambda_i^{(v)}}{2} \pm \frac{\gamma \sqrt{(\lambda_i^{(v)})^2 - 4\lambda_i^{(x)}}}{2}| < 1$, where $\mu_{i\pm} \in \mathbb{R}$. After some calculation, one can find that:

$$0 < \gamma < \frac{4}{\lambda_i^{(v)} + \sqrt{(\lambda_i^{(v)})^2 - 4\lambda_i^{(x)}}}, \text{ for } i = 2, 3, \dots, n, \tag{14}$$

completing the proof. ■

Remark 1: Notice that the interpretation of this condition in general is hard and not immediately intuitive. However, if we restrict our analysis to the set of graphs considered for our simulations and experiments, that is graphs such that $\mathcal{L}_x = \zeta \mathcal{L}_v$ for some $\zeta \in \mathbb{R}$, then the meaning becomes clearer, and the results from the Lemma simplifies to

$$\begin{cases} 0 < \gamma < \zeta, & \text{if } \lambda_{\max} \leq \frac{4}{\zeta^2} \\ 0 < \gamma < \frac{4}{\zeta \lambda_{\max} + \sqrt{\zeta^2 \lambda_{\max}^2 - 4\lambda_{\max}}}, & \text{otherwise.} \end{cases}$$

This condition on the value of the maximum eigenvalue of our Laplacian can be understood as a local density since the eigenvalues are upper-bounded by two times the largest node degree of the graph. A conclusion that can be taken is that if, for a set of possible graphs, the maximum node degree is smaller or equal to $2/\zeta^2$ then all graphs are stable for the same band of discretization time.

D. STABILITY ANALYSIS OF PLATOONS OF VEHICLES

Differently from SOCNs, platoons of vehicles are expected to follow a pre-specified formation. As such, we would like the solution of the error dynamics given by (6) to converge to zero as time passes, rather than reaching a consensus. For this, the traditional notion of stability for discrete-time linear time-invariant systems is sufficient, without needing to analyze consensus. Based on this, consider the following Lemma:

Lemma 3: For a given undisturbed discrete-time vehicle platoon system (6), let the Laplacian matrices of the position and velocity graphs share the same set of eigenvectors, and let both graphs be connected. Then, the vehicle platoon is guaranteed to asymptotically track reference (4) for any initial

condition if and only if for all $i = 1, 2, \dots, n$:

$$\begin{cases} 0 < \gamma < \frac{\lambda_i^{(v)} + 1}{\lambda_i^{(x)} + 1}, & \text{if } (\lambda_i^{(v)} + 1)^2 - 4(\lambda_i^{(x)} + 1) < 0 \\ 0 < \gamma < \frac{4}{\lambda_i^{(v)} + 1 + \sqrt{(\lambda_i^{(v)} + 1)^2 - 4(\lambda_i^{(x)} + 1)}}, & \text{otherwise} \end{cases} \tag{15}$$

where $\lambda_i^{(x)}$ and $\lambda_i^{(v)}$ are the eigenvalues of $\mathcal{L}_x$ and $\mathcal{L}_v$, respectively with matching eigenvectors.

Proof: The proof of this Lemma follows very closely to the one of Lemma 2, and as such is omitted due to space limitations. ■

With Lemmas 2 and 3 we now have an understanding of how the update cycle affects the stability of both SOCNs and vehicle platoons. We next look at how we can use this understanding to characterize the robustness of our networks as a function of the update cycle and how that affects the optimal topologies.

E. ON OUR MODELING ASSUMPTIONS

Before delving further, we will briefly discuss the assumptions of our model and their implications.

The first assumption we address concerns the graph being undirected. Our findings on stability and robustness do not readily apply to directed networks, since the proofs of Lemmas 2 and 3, and Theorems 1 and 2 rely on the orthogonality of the basis of eigenvectors. In order to extend this result, one would need to approach the estimation of the eigenvalues and of the $\mathcal{H}_2$ -norm differently. Beyond this, the merits of using undirected graphs as a framework become evident in our simulations and experiments. By confining our analysis to a subset of undirected graphs that meet our assumptions, we can interpret our design choice as determining the optimal spring and damper constants to dictate agent interactions. Such an interpretation would be less intuitive in a directed setting, where interactions would need to be viewed as unidirectional springs.

Another fundamental assumption across all the Lemmas and Theorems in this article is that both the position and velocity graphs share the same set of eigenvectors. While this assumption remains the most generic condition under which our results are valid, the authors have not identified any necessary and sufficient graph-theoretical conditions that ensure that the Laplacians of two graphs share the same eigenvector set. This absence makes it challenging to offer an intuitive interpretation of this assumption concerning its application. Consequently, we have concentrated on pinpointing sufficient conditions that validate this assumption.

In our simulations and experiments, we opted to examine the set of position and velocity graphs that are proportional to one another, implying that $\mathcal{L}_x = \zeta \mathcal{L}_v$ for some real value ζ . This condition ensures that both graphs share the same eigenvector set. However, these are not the sole graph types meeting this criterion. For example, any pair consisting of a graph and its complementary graph, given by $\mathcal{L}_v = nI - J - \mathcal{L}_x$, results in a pair of Laplacians that share the same eigenvectors. Factoring in this assumption in our analysis alters our

interpretation, yet it remains consistent with our theoretical findings.

IV. ROBUSTNESS ANALYSIS

Robustness analysis is a key concern in improving the behavior or dynamics of a system with uncertainties, particularly in the context of vehicle platoons where it is related to safety, traffic capacity, and passenger comfort. Many studies have focused on addressing robustness issues in noisy consensus networks, with some using the $\mathcal{H}_\infty$ measure to evaluate the frequency response magnitude of vehicle platoons [24], [45], [46]. Other works have employed the $\mathcal{H}_2$ -norm as a robustness index to capture the steady-state dispersion of the states of the entire system [41], [42], [47], [48].

In this section, we use the $\mathcal{H}_2$ -norm of the system, which quantifies the values of disturbances from input to output as a scalar performance measure, to provide a quantitative approach to evaluating the expected value of the steady-state variance of the output of the entire system in response to zero-mean stochastic disturbances as input. We show that this performance measure can be quantified using the eigenvalues of the Laplacian matrices of the corresponding underlying graphs associated with the time step in the discrete-time domain.

A. $\mathcal{H}_2$ -BASED ROBUSTNESS MEASURES

We begin this section with a general discrete-time LTI system subject to persistent stochastic disturbances (white noise) with zero mean and unitary covariance, resulting in the states fluctuating around the equilibrium:

$$\begin{cases} z(k+1) = Az(k) + B\xi(k) \\ y(k) = Cz(k), \end{cases} \quad (16)$$

where $\xi(k) \sim \mathcal{N}(0, I)$, and $y(k)$ is the output of the networks. Matrix $C = [C_x, C_v]$ is the output matrix of the system, where C_x and $C_v \in \mathbb{R}^{n \times n}$ indicate which nodes are measured for position and velocity, respectively.

Definition 2: Given a zero mean, unitary covariance Gaussian disturbance $\xi(k)$, and an output matrix $C = [C_x, C_v]$, the average steady-state variance of the output of the network is given by

$$\rho_{ss}(A, C) = \lim_{k \rightarrow \infty} \mathbb{E}[y(k)^\top y(k)], \quad (17)$$

and is considered a performance metric of the system.

Remark 2: For a system with dynamics as given in (16), we can compute the performance metric given in Definition 2 as

$$\rho_{ss}(A, C) := \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{Trace}(H(e^{j\omega})^\top H(e^{j\omega})) d\omega, \quad (18)$$

where H is the transfer function of (16).

Notice that both Definition 2 and Remark 2 are common results in the literature about the $\mathcal{H}_2$ -norm of linear systems. Also, it is well known that since matrix A is not necessarily Hurwitz, our performance metric might not be well defined for any choice of output matrix C . If any of the unstable

modes of the system is present in the transfer function H , the performance measure is not well defined. One can ensure this does not happen when selecting the output of the system. Specifically in this article, C is selected between two different options $C \in \{[0, M_n], [M_n, 0]\}$, where $M_n = I - (1/n)J_n$, is called the centering matrix. All of these choices of output matrix can be verified to produce a stable transfer function for either SOCNs or vehicle platoons. Intuitively, for the context of SOCNs, by choosing one of these matrices as the output matrix, we are defining the output of our system to lie on orthogonal spaces to the consensus set (generated by the span of the eigenvectors associated with the unitary eigenvalue).

An alternative, often more convenient way, of computing the $\mathcal{H}_2$ -norm of an LTI system is through its controllability (P) and observability (Q) Gramians

Remark 3: If a stable, discrete-time LTI system is controllable, there is a unique positive definite solution P to the equation $APA^\top - P = -BB^\top$. This solution can be expressed as $P = \sum_{k=0}^{\infty} A^k B B^\top (A^\top)^k$. Similarly, if the stable system is observable, there is a unique positive definite solution Q to the equation $A^\top Q A - Q = -C^\top C$. This solution can be written as $Q = \sum_{k=0}^{\infty} (A^k)^\top C^\top C A^k$.

Notice, however, that for the existence and uniqueness of the solution of the Lyapunov Equation, we need the stability of the system, which is not the case for SOCNs. Later in this article, there will be ways in which one can deal with this problem and use the Lyapunov equation to compute the $\mathcal{H}_2$ -norm of a SOCN.

We next consider three different scenarios and review the interpretations of the $\mathcal{H}_2$ -norm presented in [41], [49]. In the first scenario, the input ξ is a stochastic noise with zero mean and unitary covariance, and the $\mathcal{H}_2$ -norm of the system is the steady-state total variance of all of the output components. For the vehicle platoon system, this disturbance can be expressed as the stochastic noise received by each vehicle, which may stem from uncertainties in the states of each agent. The $\mathcal{H}_2$ -norm quantifies the expected values of the output deviation of all vehicles from the desired trajectory in the presence of noise:

$$\|\mathcal{H}\|_2^2 = \lim_{k \rightarrow \infty} \mathbb{E}\{y(k)^\top y(k)\}. \quad (19)$$

In the second scenario, we consider a random initial condition z_0 with correlation $\mathbb{E}(z_0 z_0^\top) = BB^\top$ and no input/disturbance. The $\mathcal{H}_2$ -norm is the sum of the resulting response $y(k)$. For the vehicle platoon system, since

$$BB^\top = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma I \end{bmatrix},$$

the response to the random initial condition implies that each vehicle is only subject to a random initial velocity stochastic perturbation. Therefore, the $\mathcal{H}_2$ -norm evaluates the expected values of the total energy required to steer all vehicles in the

formation to follow the desired velocity $\bar{v}(k)$, i.e.,

$$\|\mathcal{H}\|_2^2 = \sum_{k=0}^{\infty} \mathbb{E}\{y(k)^\top y(k)\}. \quad (20)$$

In the third scenario, we consider N experiments in which the i th agent is subjected to an impulse in each experiment, i.e., $e_i \delta(k)$ where e_i is a basis vector with a 1 in the i th component and zeros everywhere else. The corresponding output is $y_{e_i}(k) \in \mathbb{R}^N$. The $\mathcal{H}_2$ -norm is the total sum of the resulting response $y(k)$ of the N experiments. For the vehicle platoon system, this scenario represents a disturbance that affects only one vehicle at a time. The $\mathcal{H}_2$ -norm captures the expected values of the total energy required to steer the formation back to the desired trajectory after each disturbance, i.e.,

$$\|\mathcal{H}\|_2^2 = \sum_{i=1}^N \sum_{k=1}^{\infty} \mathbb{E}\{y_i(k)^\top y_i(k)\}. \quad (21)$$

We then evaluate these robustness measures for SOCN (3) concerning their corresponding outputs.

Theorem 1: For a given discrete-time SOCN under Gaussian noise (3), let the Laplacian matrices of the position and velocity graphs share the same set of eigenvectors, and let both graphs be connected. Then the performance measure of the system defined as in Definition 2 can be computed as:

$$\rho_{ss}(A; [M_n, 0]) = \sum_{i=2}^n \frac{\gamma \lambda_i^{(v)} - \gamma^2 \lambda_i^{(x)} - 2}{\lambda_i^{(x)} \mathcal{M}_i^{\text{SOCN}}} \quad (22)$$

$$\rho_{ss}(A; [0, M_n]) = \sum_{i=2}^n \frac{-2}{\mathcal{M}_i^{\text{SOCN}}} \quad (23)$$

where

$$\begin{aligned} \mathcal{M}_i^{\text{SOCN}} &= \gamma^3 \left(\lambda_i^{(x)} \right)^2 + 2\gamma \left(\lambda_i^{(v)} \right)^2 + 4\gamma \lambda_i^{(x)} \\ &\quad - 3\gamma^2 \lambda_i^{(x)} \lambda_i^{(v)} - 4\lambda_i^{(v)}. \end{aligned}$$

Proof: To prove the Theorem, we begin by defining the disagreement state as

$$\begin{aligned} x_d(k) &= \left(I - \frac{1}{n} J_n \right) x(k) = M_n x(k) \\ v_d(k) &= \left(I - \frac{1}{n} J_n \right) v(k) = M_n v(k). \end{aligned}$$

After some analysis of the equations above, one quickly concludes that the disagreement variables are simply the original variable minus its average. The disagreement dynamics is given by:

$$\begin{aligned} \begin{bmatrix} x_d(k+1) \\ v_d(k+1) \end{bmatrix} &= \begin{bmatrix} M_n & 0 \\ 0 & M_n \end{bmatrix} \begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} \\ &= \begin{bmatrix} M_n & \gamma M_n \\ -\gamma \mathcal{L}_x & M_n - \gamma \mathcal{L}_v \end{bmatrix} \begin{bmatrix} x_d(k) \\ v_d(k) \end{bmatrix} \end{aligned}$$

$$+ \begin{bmatrix} 0 \\ \sqrt{\gamma} M_n \end{bmatrix} \xi(k),$$

with output given by either

$$y(k) = M_n x(k) = x_d(k) = M_n x_d(k),$$

or

$$y(k) = M_n v(k) = v_d(k) = M_n v_d(k),$$

since $\mathcal{L}_{x,v} = \mathcal{L}_{x,v} M_n$, and $M_n M_n = M_n$.

Notice that the original dynamics in (3) and the disagreement dynamics have, for the same input, the same output. As a consequence, one can conclude that both have the same $\mathcal{H}_2$ -norm after consulting Definition 2.

The disagreement dynamics is a useful tool for computing the $\mathcal{H}_2$ -norm of the original SOCN because, unlike the original system, it is stable. This can be verified by computing the eigenvalues of the system as the values of $\mu \in \mathbb{C}$ for which:

$$\det \left(\begin{bmatrix} M_n & \gamma M_n \\ -\gamma \mathcal{L}_x & M_n - \gamma \mathcal{L}_v \end{bmatrix} - \mu \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \right) = 0.$$

By exploring the block and diagonal structure and performing an eigendecomposition on the four blocks of our matrices, one can write the equation above as

$$\left(\prod_{i=2}^n (1 - \mu)^2 - \gamma (1 - \mu) \lambda_i^{(v)} + \gamma^2 \lambda_i^{(x)} \right) \mu^2 = 0,$$

which has a double root at $\mu = 0$ and remaining roots at

$$\mu_{i\pm} = 1 - \frac{\gamma \lambda_i^{(v)}}{2} \pm \frac{\gamma \sqrt{\left(\lambda_i^{(v)} \right)^2 - 4 \lambda_i^{(x)}}}{2}, \quad (24)$$

for nonzero eigenvalues $\lambda_i^{(x)}$ and $\lambda_i^{(v)}$. This recovers the consensus condition from Lemma 2 and implies that if the original SOCN is such that it reaches consensus, then the corresponding disagreement dynamics is asymptotically stable.

With this, and by easily verifying controllability with the Popov-Belevitch-Hautus (PBH) test, we can write the controllability Lyapunov equation for the disagreement dynamics as

$$\begin{aligned} \begin{bmatrix} M_n \gamma M_n \\ -\gamma \mathcal{L}_x & M_n - \gamma \mathcal{L}_v \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^\top & P_{22} \end{bmatrix} \begin{bmatrix} M_n & -\gamma \mathcal{L}_x \\ \gamma M_n & M_n - \gamma \mathcal{L}_v \end{bmatrix} \\ - \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^\top & P_{22} \end{bmatrix} &= - \begin{bmatrix} 0 & 0 \\ 0 & \gamma M_n \end{bmatrix}. \end{aligned} \quad (25)$$

We then test for a solution in which the blocks of the controllability Gramian have the same eigenvalues as the Laplacian matrices. From the uniqueness of the solution of the Lyapunov equation, we know that if a solution that shares the same eigenvectors exists, then it is the only solution to the equation. Since we assume that the blocks of the Gramian share the same eigenvectors, we can write $P_{ij} = U^\top \bar{P}_{ij} U$ where $\bar{P}_{ij}$ is a diagonal matrix with the eigenvalues of that specific submatrix.

Writing the Lyapunov equation as a system of equations results in

$$M_n (P_{11} + 2\gamma P_{12} + \gamma^2 P_{22}) M_n = P_{11} \quad (26)$$

$$M_n (-P_{11}\gamma\mathcal{L}_x + P_{12} (M_n - \gamma\mathcal{L}_v - \gamma^2\mathcal{L}_x) + P_{22} (\gamma M_n - \gamma^2\mathcal{L}_v)) = P_{12} \quad (27)$$

$$\gamma^2\mathcal{L}_x P_{11}\mathcal{L}_x - 2(\gamma M_n - \gamma^2\mathcal{L}_v)P_{12}\mathcal{L}_x + (M_n - \gamma\mathcal{L}_v)P_{22}(M_n - \gamma\mathcal{L}_v) - P_{22} = -\gamma M_n \quad (28)$$

The $\mathcal{H}_2$ -norm of the system is given by:

$$\begin{aligned} \rho_{ss}(A_{\text{SOCN}}; [M_n, 0]) &= \text{Trace}([M_n, 0]P[M_n, 0]^\top) \\ &= \text{Trace}(\bar{P}_{11}). \end{aligned}$$

$$\begin{aligned} \rho_{ss}(A_{\text{SOCN}}; [0, M_n]) &= \text{Trace}([0, M_n]P[0, M_n]^\top) \\ &= \text{Trace}(\bar{P}_{22}). \end{aligned}$$

therefore, one only needs to compute the value of the two diagonal blocks to complete the proof.

By leveraging the assumption that all matrices in (26), (27) and (28) share the same set of eigenvectors we can rewrite them in terms of only their eigenvalues as follows

$$\bar{P}_{11} + 2\gamma\bar{P}_{12} + \gamma^2\bar{P}_{22} = \bar{P}_{11} \quad (29)$$

$$p_{11} = 0 \quad (30)$$

$$\begin{aligned} -\bar{P}_{11}\gamma\bar{\Lambda}_x + \bar{P}_{12} (I - \gamma\bar{\Lambda}_v - \gamma^2\bar{\Lambda}_x) \\ + \bar{P}_{22} (\gamma I - \gamma^2\bar{\Lambda}_v) = \bar{P}_{12} \end{aligned} \quad (31)$$

$$p_{12} = 0 \quad (32)$$

$$\begin{aligned} \gamma^2\bar{\Lambda}_x\bar{P}_{11}\bar{\Lambda}_x - 2(\gamma I - \gamma^2\bar{\Lambda}_v)\bar{P}_{12}\bar{\Lambda}_x \\ + (I - \gamma\bar{\Lambda}_v)\bar{P}_{22}(I - \gamma\bar{\Lambda}_v) - \bar{P}_{22} = -\gamma I \end{aligned} \quad (33)$$

$$p_{22} = 0 \quad (34)$$

Simplifying the equations above gives

$$\bar{P}_{12} = -\frac{\gamma}{2}\bar{P}_{22} \quad (35)$$

$$\bar{P}_{11} = \Lambda_x^{-1} \left(-\frac{\gamma}{2}\bar{\Lambda}_v + \frac{\gamma^2}{2}\bar{\Lambda}_x + I \right) \bar{P}_{22} \quad (36)$$

$$\bar{P}_{22} = -2(2\gamma\bar{\Lambda}_v^2 - 3\gamma^2\bar{\Lambda}_v\bar{\Lambda}_x - 4\bar{\Lambda}_v + \gamma^3\bar{\Lambda}_x^2 + 4\gamma\bar{\Lambda}_x)^{-1} \quad (37)$$

which then recovers the Theorem statement as:

$$\rho_{ss}(A; [M_n, 0]) = \sum_{i=2}^n \frac{-2 + \gamma\lambda_i^{(v)} - \gamma^2\lambda_i^{(x)}}{\lambda_i^{(x)}\mathcal{M}_i^{\text{SOCN}}}$$

$$\rho_{ss}(A; [0, M_n]) = \sum_{i=2}^n \frac{-2}{\mathcal{M}_i^{\text{SOCN}}},$$

where

$$\mathcal{M}_i^{\text{SOCN}} = \gamma^3(\lambda_i^{(x)})^2 - 3\gamma^2\lambda_i^{(x)}\lambda_i^{(v)} + 2\gamma(\lambda_i^{(v)})^2$$

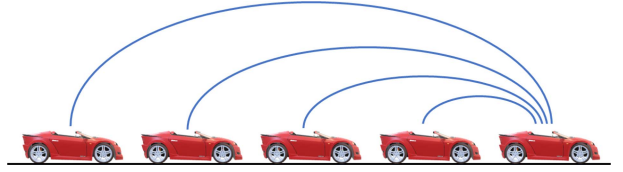


FIGURE 2. An illustration of an undirected star graph consisting of 5 agents.

$$+ 4\gamma\bar{\lambda}_i^{(x)} - 4\lambda_i^{(v)},$$

which concludes the proof. ■

Theorem 2: For a given discrete-time vehicle platoon system (6), let the Laplacian matrices of the position and velocity graphs share the same set of eigenvectors, and let both graphs be connected. Then, the performance measure of the system as defined in Definition 2 can be quantified as:

$$\rho_{ss}(A; [M_n, 0]) = \sum_{i=2}^n \frac{\gamma(\lambda_i^{(v)} + 1) - \gamma^2(\lambda_i^{(x)} + 1) - 2}{(\lambda_i^{(x)} + 1)\mathcal{M}_i^{\text{VP}}} \quad (38)$$

$$\rho_{ss}(A; [0, M_n]) = \sum_{i=2}^n \frac{-2}{\mathcal{M}_i^{\text{VP}}} \quad (39)$$

where $\mathcal{M}_i^{\text{VP}} = \gamma^3(\lambda_i^{(x)} + 1)^2 + 2\gamma(\lambda_i^{(v)} + 1)^2 + 4\gamma(\lambda_i^{(x)} + 1) - 3\gamma^2(\lambda_i^{(x)} + 1)(\lambda_i^{(v)} + 1) - 4(\lambda_i^{(v)} + 1)$.

Proof: The proof is similar to the proof of Theorem 1, therefore, it is not repeated here. ■

V. NUMERICAL SIMULATIONS

In this section, we present several numerical examples to illustrate and validate the theoretical results obtained in this article. These examples aim to provide insight into specific behaviors of the proposed control laws and the impact of different design parameters on the stability and robustness of the system. To present these observations in an orderly manner, unless explicitly specified otherwise, we did not consider disturbances for this section, focusing on the behavior of our system as specific parameters are changed. We consider different scenarios with varying network sizes, connectivity, and update cycles to evaluate the performance of the control laws under different operating conditions. The results of these examples provide evidence for the effectiveness of the proposed control approaches, while the experimental results in the next section demonstrate their practical value in real-world applications.

Example 1: Consider a five-node discrete-time vehicle platoon model (6) with both underlying graphs $\mathcal{G}_x$ and $\mathcal{G}_v$ being star graphs with edge weights $F((i, j)) = 2$ and $G((i, j)) = 1.5$, for (i, j) being an edge of the star graph (see Fig. 2).

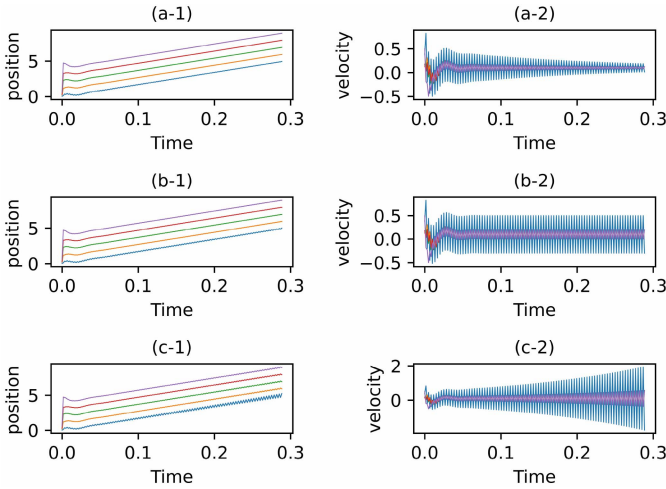


FIGURE 3. In this figure, a discrete-time vehicle platoon dynamic with $n = 5$ vehicles and a star graph as underlying position and velocity graphs is considered. The update cycle (time step) of the system is varied around the theoretical upper bound, illustrating its effects on the stability of the system. In the first two plots, (a-1) and (a-2), the update cycle is chosen slightly below the theoretical upper bound; in the middle two plots (b-1) and (b-2) it is exactly at the upper bound; and for the last two plots (c-1) and (c-2) it is slightly above the upper bound.

Their corresponding Laplacian matrices are given by

$$\mathcal{L}_x = \begin{bmatrix} 8 & -2 & -2 & -2 & -2 \\ -2 & 2 & 0 & 0 & 0 \\ -2 & 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 2 & 0 \\ -2 & 0 & 0 & 0 & 2 \end{bmatrix},$$

$$\mathcal{L}_v = \begin{bmatrix} 6 & -1.5 & -1.5 & -1.5 & -1.5 \\ -1.5 & 1.5 & 0 & 0 & 0 \\ -1.5 & 0 & 1.5 & 0 & 0 \\ -1.5 & 0 & 0 & 1.5 & 0 \\ -1.5 & 0 & 0 & 0 & 1.5 \end{bmatrix},$$

and their eigenvalues by

$$\lambda(\mathcal{L}_x) = [0 \quad 2 \quad 2 \quad 2 \quad 10],$$

and

$$\lambda(\mathcal{L}_v) = [0 \quad 1.5 \quad 1.5 \quad 1.5 \quad 7.5].$$

The state-space matrices of the system are built as in (6).

To observe numerically the behavior of the system around the theoretical upper bound of $\gamma \leq 0.2895$ (as given by Lemma 3), the simulations (6) are done for random initial states and three values of the update cycle: $\gamma = 0.2885$, $\gamma = 0.2895$, and $\gamma = 0.2905$. Fig. 3 shows that when the update cycle is slightly smaller than the upper bound ($\gamma = 0.2885$), the system is stable, since the velocity output converges asymptotically to the desired velocity, and all vehicles keep a spacing of $\Delta = 1$ between each other. The system

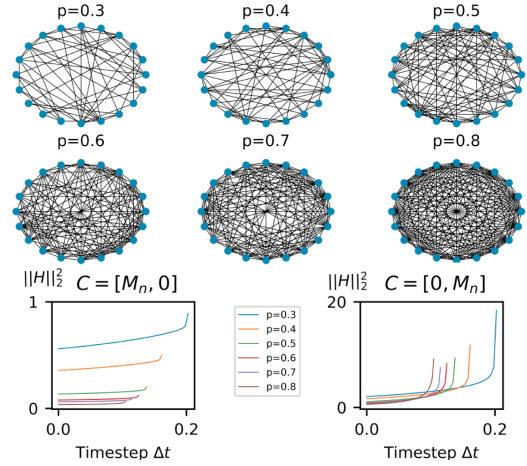


FIGURE 4. This figure illustrates, for the six Erdős-Rényi graphs depicted, the relationship between the density of connections and performance metric for different values of the update cycle. This simulation illustrates one of the most interesting observations of this work, where for networks with the same number of nodes and following the same model structure, more densely connected networks might be outperformed in terms of robustness by a less densely connected network if the update cycle is large enough.

becomes marginally stable when we are at our upper bound ($\gamma = 0.2895$), and in the simulation, one can observe both position and velocity values fluctuating around the equilibrium value. When the update cycle is slightly larger than the upper bound ($\gamma = 0.2905$), the system is unstable, and both position output and velocity output diverge.

Example 2: The next set of simulations investigates the relationship between connectivity and the performance metric for varying values of the update cycle. To perform this analysis we consider a set of 20 agents and an Erdős-Rényi model [50] for the underlying position and velocity graphs. By varying the parameters that generate the network, six graphs with different values of edge density are obtained and their behavior is compared in terms of the performance metric. From Fig. 4 one can notice that for any pair of graphs with different edge densities, the performance metric graphs cross for some value of the update cycle, indicating a trade-off between robustness and connectivity between networks following the same structure. Notice that this happens because, for the same network model, more densely connected networks have higher values of the maximum eigenvalue, which from Lemma 3 means their discretized dynamics will become unstable for smaller values of the update cycle.

Interestingly, this contradicts the traditional intuition from continuous-time consensus networks, where the more connected the networks, the more coherent or better performance measures the system has (see [42], [49], [51] and references therein). As a result, these networks are more capable of reducing the influence of stochastic disturbance by increasing the number of connections/edges. However, in the discrete-time domain, one can see from Fig. 4 that denser networks require faster agents (i.e., smaller update cycles) to achieve

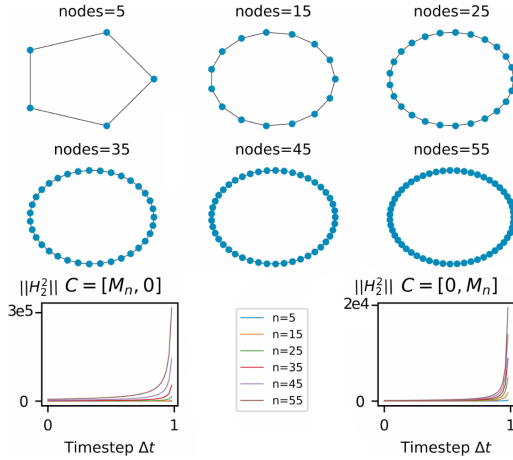


FIGURE 5. This figure illustrates, for the six ring graphs depicted, the relationship between network size and performance metric for different values of the update cycle. As pointed out in Remark 1, the crossing of the performance metric functions between two different graphs is highly dependable on the largest eigenvalue of the Laplacian of the graphs, which is upper-bounded by twice the largest node degree of the network. In this case, where ring graphs are considered, no trade-off is observed between robustness and the considered models for varying values of the update cycle because they all have the same value of node degree, independently of the size of each network.

the same level of robustness as sparse networks (see Fig. 1). Moreover, the subplots in Fig. 4 show that the values of performance measure dramatically increase where the time step γ is approaching the condition of marginally stable discussed in Section III-C.

Example 3: In this final set of simulations, we study the effect of the size of a network on the performance metric of the system. We fix our network topologies as ring graphs and vary the number of nodes from 5 to 55 by increments of 10, as depicted in Fig. 5. Notice that in this set of simulations, the order of the graphs remains unchanged, and all systems become unstable for the same value of the update cycle. This happens because the chosen graph topology is very sparse, and all graphs satisfy the same condition for instability when analyzed in terms of Lemma 2.

In this section we provided three sets of simulations that investigate the behavior of our system as different parameters vary. These findings suggest that it is important to carefully consider the size and update cycle of a second-order consensus network to optimize its performance and robustness. We next investigate how this framework performs in a more realistic application, with simulations and experiments of a platoon of vehicles.

VI. EXPERIMENTAL RESULTS

Our experimental approach involves two phases: a set of simulations executed on a third-party nonlinear platform, the Qlabs; and real-world experiments using QCars [52]. These are remotely controlled cars that operate autonomously under the supervision of a control station. The goal of these experiments is to test the application of our theoretical framework

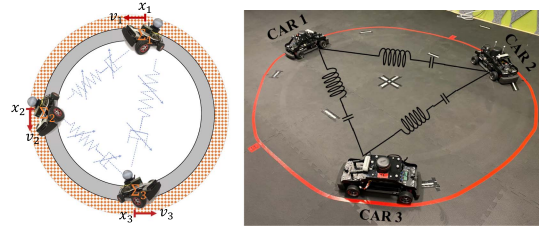


FIGURE 6. A vehicle platoon of multiple autonomous robots. The converted 1/10-scale RC cars are equipped with LiDAR, a stereo camera, an inertial measurement unit, and embedded GPUs. The interaction between the cars is modeled as a virtual spring-damper on their angular state inside the circle, as indicated.

in more general applications where the real system is not the same as the design model and noise is inherent to the hardware implementation.

As discussed before, we assume during our experiments that the position graph is a multiple of the velocity graph, *i.e.*, $\mathcal{L}_x = \zeta \mathcal{L}_v$ for some $\zeta \in \mathbb{R}$. This is a way of complying with the assumptions from the theory part about both graphs having the same eigenvectors while also maintaining some intuition on the meaning of our choices of parameters: The position and velocity graph gains represent the “spring and damper constants” imposed by our feedback law, as depicted in Fig. 6. That is, by assuming this structure we are imposing a virtual spring-damper as the interaction between our agents with the freedom of finding the best set of parameters with respect to our performance metric.

Furthermore, the code used for this section of the article is available in a GitHub repository [53].

A. VIRTUAL CAR SIMULATIONS

The virtual simulations are run on Quanser’s Qlabs. We can determine the position of the cars using the built-in GPS module. The velocity and steering commands for the QCars are issued using Python. We configure the QCars to follow a fixed circular trajectory with a radius of 20 meters.

The experiments are set up as follows: Initially, we apply a lane-keeping control to ensure that the cars remain within the pre-specified manifold, while still allowing for varying velocities within it. Subsequently, each agent receives and tracks a desired trajectory that evolves according to their initial position and a fixed desired velocity that applies to all cars. All control methods described thus far operate locally for each agent, independent of their neighbors. This is indicated by the “Augmented System” block in Fig. 7, which can be thought of as a collection of independently actuated augmented agents. In the next step, each augmented agent receives the desired acceleration from a formation control strategy, which focuses on maintaining a safe distance from its neighbors per a specific network structure. Additionally, it is important to highlight that, while we did not introduce any Gaussian noise into the simulation, the inherent non-linearity of the actual system

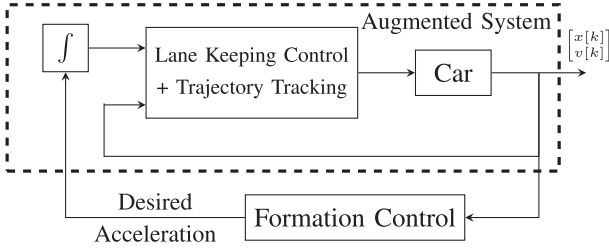


FIGURE 7. Block diagram of the control structure implemented for the experiments. Notice that all control efforts inside the augmented system are done on a node level and use no information of possible neighbors. One could think of the augmented system as a collection of independently augmented nodes.

introduces uncertainty from the feedback law's perspective, which in turn serves as a validation of its robustness

To recover a consensus network, we look at the dynamics of the error between the state of each node and their desired values, similarly to how it is described in Section III-B. We define $\tilde{x} = x - x^r$ and $\tilde{\dot{x}} = \dot{x} - \dot{x}^r$, and during our experiments we imposed a ramp reference for x^r which means that $\dot{x}^r := v^r$, and $x_i^r = x_{i0}^r + v^r t$ for all nodes $i = 1, 2, \dots, n$.

The dynamics of the output reference tracking and the formation control block can be expressed respectively as,

$$\begin{bmatrix} \tilde{x}(k+1) \\ \tilde{\dot{x}}(k+1) \end{bmatrix} = \begin{bmatrix} I & \gamma I \\ -\gamma a(\mathcal{L} + I) & I - \gamma b(\mathcal{L} + I) \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ \tilde{\dot{x}}(k) \end{bmatrix}. \quad (40)$$

where $\mathcal{L}$ is the Laplacian matrix of a n -node unweighted, undirected network graph, and a and b are the spring and damper coefficients of our virtual interactions.

Let us consider the SOCN (40) consisting of $n = 5$ agents (Qcars) in which the underlying connecting graphs $\mathcal{G}_x$ and $\mathcal{G}_v$ are unweighted complete graphs. When the update cycle is 0.1 seconds (10 Hz), the SOCN is stable and both the position and velocity converge asymptotically to its steady-state. The velocity reaches a steady-state value of 12.5 m/sec, and the positions of each node reach a consensus state of 0 (maintaining a constant spacing amongst other nodes). Now, when the update cycle is increased to 0.25 seconds (4 Hz), the underlying system becomes unstable, and the velocity and position outputs both diverge.

Next, we implement the same experiment with $n = 5$ (Qcars), but having the underlying connecting graphs, $\mathcal{G}_x$ and $\mathcal{G}_v$ as unweighted path graphs. We can observe that when the update cycle is 0.1 seconds (10 Hz), the SOCN is stable and both the position and velocity graph converge to its steady-state. Now, when we increase the update cycle to 0.25 seconds (4 Hz), we can observe that the system is still stable, and the velocity and position graphs converge to their steady-state.

The experimental results thus far, depicted in Fig. 8, indicate that there lies a trade-off between faster agents and network density. We can observe that while increasing the number of nodes, the path graph lies stable for higher update cycles, while the complete graph breaks and becomes unstable.

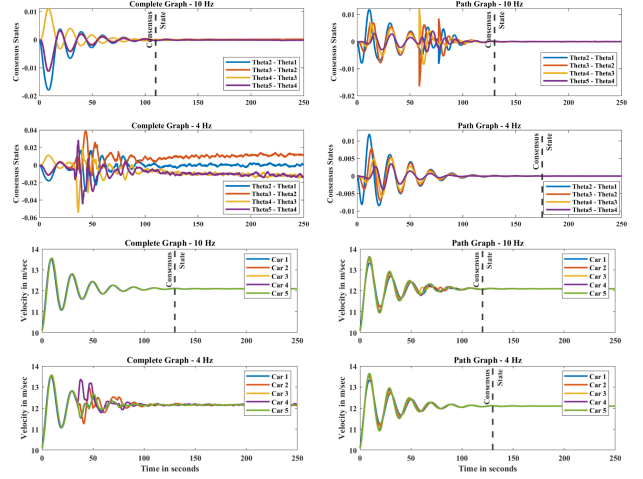


FIGURE 8. This figure depicts the consensus states reached in position, $\mathcal{G}_x$, (sub-plots in rows 1 and 2) and for velocity $\mathcal{G}_v$, (sub-plots in rows 3 and 4) for a set of 5 nodes, and when the update cycle is 0.1 seconds (sub-plots in rows 1 and 3) and 0.25 seconds (sub-plots in rows 2 and 4) correspondingly. The left sub-plots depict the consensus states reached in position and velocity when the underlying formation is in complete graph topology and the right sub-plots depict the consensus states reached when the underlying formation in the network is of path graph topology. It can be seen that while the nodes remain stable for a path graph topology for higher update cycles (0.25 seconds), the complete graph topology loses consensus.

B. SIMULATION STUDIES OF MINIATURE SELF-DRIVING CARS

In this section, we aim to examine the real-time applications of our theoretical advancements using QCars [52]. Throughout the experiments presented in this section, the QCars were operated remotely via Wi-Fi.

For our experiments, we place the QCars in a room equipped with six Opti-Track motion capture cameras. The vehicles are programmed to follow a circular trajectory with a radius of 0.85 meters. We use Quanser's QCars as autonomous ground vehicles to execute this trajectory, and the control architecture for these experiments is implemented using Matlab Simulink.

The overall architecture of the experiments is the same as the previous section as depicted in Fig. 7, except that now, we also include an internal PI controller for tracking velocity references given to each car, and the parameters were tuned so that each car follows a step reference on the speed with zero steady-state error and with approximately the same rising time, accounting for each car's unique behavior. The cars reach a reference velocity in about one second, limiting how quickly we can enforce consensus to the system before higher-order behavior related to the dynamics of the motors starts being noticeable.

While our car formation comprises a system that is structurally distinct from a linear consensus network, we assert that insights derived from considering the structure of our interconnections remain applicable to our system. Fig. 9 displays the experimental results when the underlying interconnection

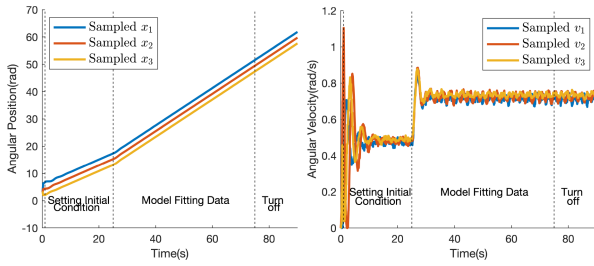


FIGURE 9. Full Experimental Data for a Ring Graph: Following a brief initialization period (approximately 1 s), all cars are brought to an initial velocity before the formation control is initiated, and a new desired velocity is set. This procedure ensures non-zero initial conditions for the system during the period considered for model fitting (from 25 seconds to 75 seconds).

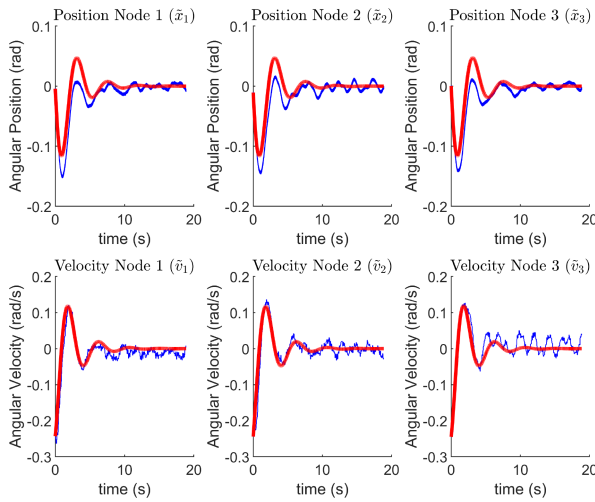


FIGURE 10. The figure above presents sampled data from our experiments with a Path Graph (blue), alongside the simulation results for the fitted consensus network (red). The quality of the fit suggests that a consensus network offers a robust linear approximation for the system's dynamics.

is selected to be a ring graph. As can be observed, this approximation allows us to reproduce the platoon dynamics for the cars as described in Section III-B. The resulting fit models, in comparison with the actual data, are illustrated in Fig. 10 for the path graph.

VII. CONCLUDING REMARKS

This article focuses on the distributed consensus and vehicle platoon control problems in discrete-time systems. We propose a graph-theoretic approach to design the feedback laws for these systems and provide a necessary and sufficient condition for their stability based on the structure of the underlying graphs and the update cycles (time steps) of the autonomous agents. Additionally, we use algebraic graph theory to study the robustness and performance of cooperative control methods in discrete-time vehicle platoons. We employ a $\mathcal{H}_2$ -based metric as a performance measure that captures the coherence of the system [41] and quantifies the expected values of output dispersion in the presence of stochastic disturbances. This measure is shown to increase monotonically

with network size and decrease with network connectivity. We also observe a fundamental tradeoff between graph connectivity and the update cycle, illustrating this through experimental results. During our experimental results section we argue that despite the dynamics of real cars being nonlinear, the results of this article can still be relevant through the choice of control structure. Key contributions of this article include:

- Necessary and sufficient conditions for stabilizing a discrete-time vehicle platoon model;
- A quantitative method for evaluating the performance measures of vehicular formation dynamical systems in a discrete-time framework;
- An analysis of the relationship between performance measures, network size, connectivity, and update cycles;
- A discussion of fundamental tradeoffs between performance measures and the restrictions on vehicle update cycles; and
- Experimental validation of our theoretical observations.

As a potential future direction, we suggest casting the feedback design problem as a convex optimization problem to improve both the stability and robustness of second-order consensus networks simultaneously, similar to the approach taken in [54].

ACKNOWLEDGMENT

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

REFERENCES

- [1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
- [2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [3] P. Wieland, J.-S. Kim, H. Scheu, and F. Allgöwer, "On consensus in multi-agent systems with linear high-order agents," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 1541–1546, 2008.
- [4] M. C. De Gennaro and A. Jadabae, "Formation control for a cooperative multi-agent system using decentralized navigation functions," in *Proc. IEEE Amer. Control Conf.*, 2006, pp. 1346–1351.
- [5] O. P. Mahela et al., "Comprehensive overview of multi-agent systems for controlling smart grids," *CSEE J. Power Energy Syst.*, vol. 8, no. 1, pp. 115–131, Jan. 2022.
- [6] K. Y. Liang, J. Mårtensson, and K. H. Johansson, "Heavy-duty vehicle platoon formation for fuel efficiency," *IEEE Trans. Intell. Trans. Syst.*, vol. 17, no. 4, pp. 1051–1061, 2015.
- [7] M. Campion, P. Ranganathan, and S. Faruque, "UAV swarm communication and control architectures: A review," *J. Unmanned Veh. Syst.*, vol. 7, no. 2, pp. 93–106, 2018.
- [8] A. Darabi and M. Siami, "Stability and robustness analysis of epidemic networks with multiple time-delays," in *Proc. Amer. Control Conf.*, 2022.
- [9] F. Jiang, Q. Zhang, Z. Yang, and P. Yuan, "A space-time graph based multipath routing in disruption-tolerant earth-observing satellite networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 5, pp. 2592–2603, Oct. 2019.
- [10] A. Zelenkauskaitė, N. Bessis, S. Sotiriadis, and E. Asimakopoulou, "Interconnectedness of complex systems of Internet of Things through social network analysis for disaster management," in *Proc. IEEE 4th Int. Conf. Intell. Netw. Collaborative Syst.*, 2012, pp. 503–508.

- [11] Y. Zhao, L. Jiao, R. Zhou, and J. Zhang, "UAV formation control with obstacle avoidance using improved artificial potential fields," in *Proc. IEEE 36th Chin. Control Conf.*, 2017, pp. 6219–6224.
- [12] Y. Yang, Y. Xiao, and T. Li, "A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 815–841, Secondquarter 2021.
- [13] Z. Xu and H. Pei, "Satellite formation control and navigation experiment platform based on UAVs," in *Proc. IEEE 35th Chin. Control Conf.*, 2016, pp. 5764–5768.
- [14] S. Berto, A. Perdomo-Sabogal, D. Gerighausen, J. Qin, and K. Nowick, "A consensus network of gene regulatory factors in the human frontal lobe," *Front. Genet.*, vol. 7, 2016, Art. no. 31.
- [15] Y. Wang, H. Ishii, F. Bonnet, and X. Défago, "Resilient consensus against epidemic malicious attacks," in *Proc. IEEE Eur. Control Conf.*, 2021, pp. 1393–1398.
- [16] E. L. Sander, J. T. Wootton, and S. Allesina, "Ecological network inference from long-term presence-absence data," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, 2017.
- [17] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, 2010.
- [18] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [19] W. Ren and R. W. Beard, "Consensus algorithms for double-integrator dynamics," in *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Berlin, Germany: Springer, 2008, pp. 77–104.
- [20] W. Yu, G. Chen, M. Cao, and J. Kurths, "Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 40, no. 3, pp. 881–891, Jun. 2010.
- [21] W. Yu, W. X. Zheng, G. Chen, W. Ren, and J. Cao, "Second-order consensus in multi-agent dynamical systems with sampled position data," *Automatica*, vol. 47, no. 7, pp. 1496–1503, 2011.
- [22] Y. Yazıcıoğlu, W. Abbas, and M. Shabbir, "Structural robustness to noise in consensus networks: Impact of degrees and distances, fundamental limits, and extremal graphs," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4777–4784, Oct. 2021.
- [23] H. Hao and P. Barooah, "Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction," *Int. J. Robust Nonlinear Control*, vol. 23, no. 18, pp. 2097–2122, 2013.
- [24] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed H-infinity controller synthesis," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1353–1364, May 2018.
- [25] D. Zelazo and M. Bürger, "On the robustness of uncertain consensus networks," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 170–178, Jun. 2017.
- [26] F. Pasqualetti, C. Favaretto, S. Zhao, and S. Zampieri, "Fragility and controllability tradeoff in complex networks," in *Proc. IEEE Annu. Amer. Control Conf.*, 2018, pp. 216–221.
- [27] W. Abbas, M. Shabbir, A. Y. Yazıcıoğlu, and A. Akber, "Tradeoff between controllability and robustness in diffusively coupled networks," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1891–1902, Dec. 2020.
- [28] F. Pasqualetti, S. Zhao, C. Favaretto, and S. Zampieri, "Fragility limits performance in complex networks," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 1774.
- [29] F. Sun, Z.-H. Guan, X.-S. Zhan, and F.-S. Yuan, "Consensus of second-order and high-order discrete-time multi-agent systems with random networks," *Nonlinear Anal.: Real World Appl.*, vol. 13, no. 5, pp. 1979–1990, 2012.
- [30] Y. Han and C. Li, "Second-order consensus of discrete-time multi-agent systems in directed networks with nonlinear dynamics via impulsive protocols," *Neurocomputing*, vol. 286, pp. 51–57, 2018.
- [31] D. Xie and S. Wang, "Consensus of second-order discrete-time multi-agent systems with fixed topology," *J. Math. Anal. Appl.*, vol. 387, no. 1, pp. 8–16, 2012.
- [32] P. Lin and Y. Jia, "Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically changing topologies," *Automatica*, vol. 45, no. 9, pp. 2154–2158, 2009.
- [33] G.-H. Xu, J.-S. Zhang, R.-Q. Liao, D.-X. Zhang, and Z.-H. Guan, "Second-order consensus of discrete-time multi-agent systems via one-step delayed data," in *Proc. IEEE 26th Chin. Control Decis. Conf.*, 2014, pp. 3687–3692.
- [34] G. Shi and K. H. Johansson, "Robust consensus for continuous-time multiagent dynamics," *SIAM J. Control Optim.*, vol. 51, no. 5, pp. 3673–3691, 2013.
- [35] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Optimal synchronization for networks of noisy double integrators," *IEEE Trans. Autom. Control*, vol. 56, no. 5, pp. 1146–1152, May 2011.
- [36] S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li, "String stability for vehicular platoon control: Definitions and analysis methods," *Annu. Rev. Control*, vol. 47, pp. 81–97, 2019.
- [37] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Proc. IEEE Intell. Transp. Syst.*, 2001, pp. 625–630.
- [38] D. Wang and R. Sipahi, "Stability of a large-scale connected vehicle network in ring configuration and with multiple delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 663–667, Jan. 2022.
- [39] P. Wijnbergen, M. Jeeninga, and B. Besselink, "Nonlinear spacing policies for vehicle platoons: A geometric approach to decentralized control," *Syst. Control Lett.*, vol. 153, 2021, Art. no. 104954.
- [40] B. Besselink and K. H. Johansson, "String stability and a delay-based spacing policy for vehicle platoons subject to disturbances," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4376–4391, Sep. 2017.
- [41] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson, "Coherence in large-scale networks: Dimension-dependent limitations of local feedback," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2235–2249, Sep. 2012.
- [42] M. Siami and N. Motee, "Fundamental limits and tradeoffs on disturbance propagation in linear dynamical networks," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4055–4062, Dec. 2016.
- [43] Y. Huang and M. Siami, "Role of agent update cycle in stability and robustness of second-order consensus networks," in *Proc. IEEE 30th Mediterranean Conf. Control Automat.*, 2022, pp. 643–648.
- [44] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*, vol. 10. Cambridge, U.K.: Cambridge Univ. Press, 2019.
- [45] S. E. Li, F. Gao, K. Li, L.-Y. Wang, K. You, and D. Cao, "Robust longitudinal control of multi-vehicle systems—A distributed h-infinity method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2779–2788, Sep. 2018.
- [46] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intell. Transport Syst.*, vol. 10, no. 7, pp. 503–513, 2016.
- [47] M. Siami and N. Motee, "Robustness and performance analysis of cyclic interconnected dynamical networks," in *Proc. Proc. Conf. Control Appl.*, 2013, pp. 137–143.
- [48] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *Proc. IEEE Amer. Control Conf.*, 2010, pp. 6312–6317.
- [49] E. Tegling, B. Bamieh, and D. F. Gayme, "The price of synchrony: Evaluating the resistive losses in synchronizing power networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 3, pp. 254–266, Sep. 2015.
- [50] P. Erdos et al., "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
- [51] M. Siami and N. Motee, "Fundamental limits on robustness measures in networks of interconnected systems," in *Proc. IEEE 52nd Conf. Decis. Control*, 2013, pp. 67–72.
- [52] Quanser Company Website, 2021. Accessed: Oct. 20, 2023. [Online]. Available: <https://www.quanser.com/products/qcar/>
- [53] "Git hub repository - Balancing agility and communication," 2023. Accessed: Aug. 24, 2023. [Online]. Available: https://github.com/SiamiLab/OJCSYS_2023
- [54] M. Siami and J. Skaf, "Structural analysis and optimal design of distributed system throttlers," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 540–547, Feb. 2018.