Leader–Follower-Based Self-Triggered Consensus Control of Industrial Induction Motor Drives

Zohaib Ijaz, Md. Noor-A-Rahim, and Dirk Pesch, *Senior Member, IEEE*

***Abstract*—Alternating current motors are critical to industry, as they drive many machines in the manufacturing and processing industries. To accomplish heavy tasks, often, a number of small motors must operate cooperatively, which means that the operation of the motors must be coordinated using, for example, consensus control. To accomplish this, the motors must communicate with each other. This communication can be periodic or event driven. As periodic communication may waste communication resources when no control update is needed, we propose a need-based self-triggered communication (STC) mechanism to achieve improved communication efficiency. We propose an STC technique for the leader–follower-based consensus control of induction motors. To study this method, we developed both centralized and distributed STC models. In the centralized approach, each motor is connected to a central unit that calculates the next communication time. When distributed STC is used, each motor calculates the next communication time solely based on information from directly connected neighboring motors, thus eliminating the possibility of a single point of failure. Extensive simulations were conducted to validate the proposed approaches. Our results show that the proposed self-triggered consensus control technique gets the same level of performance as a standard periodic control approach while utilizing fewer communication resources.**

***Index Terms*—Centralized and distributed control systems, induction motor, leader–follower-based consensus control, self-triggered communication.**

I. Introduction

THE manufacturing industry is the backbone of majority of advanced economies. Many manufacturing systems are equipped with a variety of motors that perform various tasks, such as conveyor belt motors, material handling motors, and so on. While these motors are available in a variety of types and configurations, the simplest type of motor to implement and maintain is the induction motor [1], [2], [3], [4]. They are robust, dependable, and widely available in a variety of power configurations. Rather than employing large motors, various production environments make use of numerous small motors to accomplish a specific task, which not only simplifies installation but also maintenance, as smaller motors are much easier to maintain. This results in lower production costs and increased equipment reliability. In this setup, the number of motors must be regulated synchronously in an industrial setting.

Speed control of an induction motor in an industrial environment is not straightforward; rather, a controller is necessary to regulate the motor's speed by adjusting the motor's inductance. Initially, induction motors are supposed to operate linearly, with a fixed and constant model of the motor. Because the motor does not always operate at a constant load, the machine's parameters vary in response to changes in the amplitude and frequency of the flux due to load. The inductance is low at low flux, but once the flux starts to increase, the induction decreases, as the motor tends to operate in the saturation region [5].

In many situations, these motors perform a particular task together, e.g., driving a conveyor belt. This means that they must be controlled in a consensus manner in order to maintain the same speed when executing assigned tasks [6], [7], [8]. In consensus control, different agents may begin with a different value (for example, speed) but eventually accomplish the same goal. In our case, induction motors can begin operating at any driving speed, but those doing the same task must achieve the same speed [7].

In an industrial setting, these motors may be deployed in different locations while collectively performing the same task. As a result, it may be challenging for all the motors to reach the reference speed at which they must converge. To address this, we adopt a multiagent approach that designates some motors as leaders (those that have access to the required reference speed) and the remainder of the motors as followers [9], [10]. In this leader–follower situation, not all the followers must be connected to the leader; rather, only one motor may be attached to the leader. The leader motor establishes the system's reference speed and adjusts its speed accordingly. The speed setting is then conveyed to the follower motors, which seek to maintain consensus, e.g., by operating at the same speed as the leader, based on speed control commands received from the leader [8].

For this, the motors must communicate with one another in order to reach a consensus. Previously, in such scenarios, it was assumed that infinite communication bandwidth was available and communication was time periodic. While time-periodic communication is simple to implement, it can waste bandwidth resources if the speed does not change over an extended period of time. By employing aperiodic communication mechanisms, such waste can be avoided. Among these, the most prominent aperiodic communication approach is event-triggered control (ETC), which is only triggered when an event occurs [11], [12], [13]. However, the continuous monitoring of the system is

Manuscript received 2 June 2021; revised 11 January 2022 and 7 June 2022; accepted 8 August 2022. Date of publication 30 August 2022; date of current version 9 December 2022. This work was supported by Science Foundation Ireland under Grant 16/RC/3918 at the CONFIRM Centre for Smart Manufacturing. *(Corresponding author: Md. Noor-A-Rahim.)*

The authors are with the School of Computer Science and Information Technology, University College Cork, T12 YN60 Cork, Ireland (e-mail: zohaib.ijaz@cs.ucc.ie; m.rahim@cs.ucc.ie; dirk.pesch@ucc.ie).

Digital Object Identifier 10.1109/JSYST.2022.3198796

required for ETC, which increases the cost due to the additional sensor operation required for continuous monitoring. This can be avoided by introducing a proactive triggering mechanism, which is known as self-triggered control (STC) [14]. In STC, the system interacts only at predetermined triggering events and not between events [15], [16]. STC can be implemented in a central or distributed fashion. In a centralized case, we assume that all of the motors are connected to a single central unit. This central unit can be any of the motors in the system that are connected to the other motors. At triggering time, each motor broadcasts its status (e.g., driving speed) to its neighbors and to the central unit. The other motors compute their control inputs based on this information, and the central unit calculates the next triggering time [17]. There is no central unit in distributed STC, and the system will not collapse due to the failure of any motor. All the motors broadcast their states at triggering time, and all the neighboring motors connected to a specific motor compute their triggering time instance and control input based on that information. To avoid the loss of state information, we employ synchronized distributed control, in which all the motors trigger at the same time or at the minimum time for all the motors to trigger. When creating synchronous distributed STC, we may run into Zeno behavior. In this case, where the next communication time calculated is equal to the current one, the motor must communicate indefinitely within a finite time period, which is not practically feasible [18]. Zeno behavior can be avoided by ensuring minimal intercommunication time.

In this article, we examine leader–follower consensus-based induction motor speed control. We investigate different parameters that can be used to control the speed of the motor. As our aim is to conserve communication resources, we propose an STC-based communication mechanism. We suggest two STC algorithms: one centralized and the other distributed. When centralized STC is used, the central unit determines when the entire system should trigger next. There is no central unit in the distributed STC, and the motors determine when to communicate next based on the information they get from their connected neighboring motors. To eliminate data loss during the triggering process for distributed STC, we employ synchronous triggering. Our technique circumvents Zeno behavior by providing a minimum communication time. We use this method for induction motors, but the proposed STC method can be used on any system, regardless of how the control signals are set up. We validate our proposed schemes through extensive simulations.

The rest of this article is organized as follows. In Section II, we list and discuss related work. The mathematical model of an induction motor is discussed in Section III, and a brief introduction to the speed control of induction motors is presented in Section IV. Section V discusses the requirements for developing consensus, including algebraic graph theory in Section V-A, consensus for induction motors in Section V-B, and the controller design for consensus in Section V-C. In Section VI, we develop the centralized STC using Lyapunov stability analysis and propose a distributed STC in Section VII. In Section VIII, we present results of the proposed approach using an example of a car manufacturing facility as a use case. Finally, Section IX concludes this article.

II. Related Work

Following the introduction in [14], considerable effort has been expended on improving ETC or STC. Dimarogonas et al. [11] develop ETC for the consensus of a multiagent system. They investigated both the centralized and distributed control approaches. The results demonstrate that Zeno behavior is not prevented, and that the system also lacks a leader. STC for distributed microgrids (MGs) is introduced in [17], where STC is used to regulate the MG's active and reactive power. However, the system also shows Zeno behavior and has no leader. Xie and Lin [19] investigate event-triggered-based leader–follower consensus control in multiagent systems, where the authors develop consensus but with a constraint on the control input. Self-triggered leader–follower consensus is explored in [15], where the authors utilize an observer to have the system's output develop STC rather than using state feedback. In [20], another attempt at an event-triggered leader–follower consensus is made. The proposed technique, however, is restricted to switched nonlinear systems. Zhang et al. [21] use distributed ETC to achieve leader–follower consensus. However, they do so using an asynchronous triggering mechanism, which means that information loss cannot be mitigated. Zhu and Jiang [22] investigate an event-triggered leader–follower consensus for more general models. Symmetry is not required for the Laplacian matrix. The controller for each agent is updated at the time of the agent's event, which makes it asynchronous. In addition, it is confined to ETC, which means that continuous monitoring is not avoided. Cheng and Li [23] present static and dynamic asynchronous edge-based event-triggered communication, where the bound on control input is assumed in static event-triggered communication but relaxed in dynamic setup. In addition, continuous monitoring is required in this case.

In our study, we have developed self-triggered consensus algorithms for induction motors using leader–follower communication. Our proposed model makes no assumptions about the input, it does not require an observer, and it also avoids Zeno behavior, making it simpler and more robust than other approaches.

III. Mathematical Model of an Induction Motor

Induction motors can be described using Park's dq-axis or by analogy to synchronous motors utilizing dynamic equivalent circuits. The motor's dynamic model is as follows:

Stator side

$$v_{ds} = R_s i_{ds} + \frac{d}{dt}\psi_{ds} - \omega_s \psi_{qs} \tag{1}$$

$$v_{qs} = R_s i_{qs} + \frac{d}{dt}\psi_{qs} + \omega_s \psi_{ds} \tag{2}$$

$$\psi_{ds} = L_s i_{ds} + L_\mu i_{dr} \tag{3}$$

$$\psi_{qs} = L_s i_{qs} + L_\mu i_{qr} \tag{4}$$

where ψ_{ds} and ψ_{qs} denote stator-side d- and q-axis flux, respectively, and v_{ds} and v_{qs} denote stator-side voltages. R_s and L_s represent the stator resistance and self-inductance, respectively, whereas L_μ represents the magnetizing inductance. i_{ds} and i_{qs} represent stator current and i_{dr} and i_{qr} represent rotor current.

Rotor side

$$v_{dr} = R_r i_{dr} + \frac{d}{dt}\psi_{dr} - (\omega_s - \omega_r)\psi_{qr} \tag{5}$$

$$v_{ds} = R_r i_{qr} + \frac{d}{dt}\psi_{qr} + (\omega_s - \omega_r)\psi_{dr} \tag{6}$$

$$\psi_{dr} = L_r i_{dr} + L_\mu i_{ds} \tag{7}$$

$$\psi_{qr} = L_r i_{qr} + L_\mu i_{qs} \tag{8}$$

where ψ_{dr} and ψ_{qr} denote the rotor-side d- and q-axis flux, respectively, and v_{dr} and v_{qr} denote the rotor-side voltages. Rotor resistance and self-inductance are denoted by R_r and L_r, respectively. ω_r and ω_s denote the rotor and stator speeds, respectively.

Electromagnetic torque is represented as

$$T_e = \frac{3}{2}\frac{P}{2} L_\mu [i_{qs} i_{dr} - i_{ds} i_{qr}] \tag{9}$$

where P denotes the pole count and T_e denotes the electromagnetic torque. The change in rotor speed when mechanical dampening is ignored is as follows:

$$\frac{d\omega_r}{dt} = \frac{P}{2J}(T_e - T_L) \tag{10}$$

where J is the inertia of the rotor and T_L is the load torque.

IV. Speed Control of an Induction Motor

On both the rotor and stator sides of an induction motor, speed control can be achieved. Owing to the simplicity of implementation, rotor-side speed regulation is similar to that of a synchronous motor. As a result, we will control the speed of an induction motor in this article using slip-regulated indirect field-oriented control (IFOC) on the rotor side, where the rotor flux is estimated using the slip relation and the stator speed is determined based on the rotor flux position. Through the proportional–integral (PI) controller, the speed loop error generates the slip instruction ω^*_{slip}. PI control is used to manage the current on the rotor side and is dependent on the motor resistance and inductance, as well as the controller bandwidth, as follows:

$$F(s) = k_p + \frac{k_i}{s} \tag{11}$$

where

$$k_p = \alpha(R_r + R_s) \tag{12}$$

and

$$k_i = \alpha(L_r + L_s) \tag{13}$$

where α denotes the controller bandwidth.

As shown in Fig. 1, the slip is added to the feedback speed signal to generate the electromagnetic reference frequency ω^*_e. The slip equation is used to calculate the rotor flux, as slip is proportional to the developed torque at constant flux. Once the position of the rotor flux is determined, the stator flux is calculated. The PI controller takes error current as an input, which is defined as the difference between the reference and intended currents, and outputs a voltage vector that is subsequently provided to the motor to achieve the required speed.

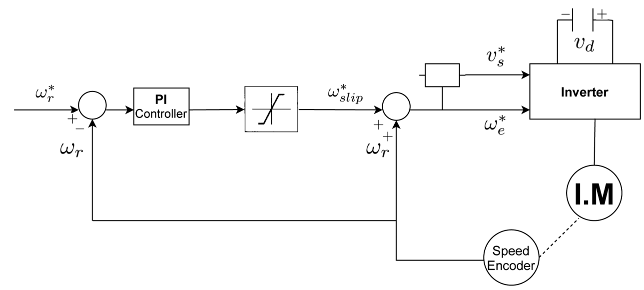

Fig. 1. Speed control of the induction motor [24].

V. Preliminaries

In the following, we introduce some preliminaries for our leader–follower-based consensus control approach. This includes a brief summary of algebraic graph theory, which is used to model the network, the system's architecture, and how we update the agents' controllers.

A. Graph Theory

We assume that $G = \{C, E, A\}$ is a weighted graph that represents the communication connectivity between motors, with $C = \{c_1, ..., c_n\}$ representing the vertex set for N motors and E representing the edge set. The edge set is composed of elements (c_i, c_j), $i, j \in \{1, 2, ..., N\}, i \neq j$. If $(c_i, c_j) \in E$, then the motor c_j can communicate with the motor c_i. The adjacency matrix is defined as $A = [a_{ij}] \in R^{N\times N}$ with $a_{ij} = 1$ when i is connected to j; otherwise, $a_{ij} = 0$. The in-degree matrix is now defined as a diagonal matrix $D = \text{diag}\{d_1\ d_2\ \ldots\ d_N\}$ with $d_i = \sum_{j\in N_i} a_{ij}$. The graph Laplacian matrix is $L = D - A$, where all row entries add up to zero. Our system model contains a large number of follower motors and a few leaders. In addition, we define a reference value that is only accessible to leaders. As a result, we divide the adjacency matrix A into two components. The first part depicts the relationship between followers and leaders, with bidirectional communication. The second part establishes a link between the leaders and the reference value (node). Communication is unidirectional in this section because the leaders only require information from the reference value in order to converge their control variable (e.g., speed) to that value, and the reference node does not require speed information from the leader motors.

B. Induction Motor Consensus

Assume that we have followers $\{f_1, f_2, f_3.....f_n\}$ and a leader l trying to reach consensus. Each node is driven by its own controller and has its own control input. Let $\omega_i(t)$ denote the speed of the follower i and $\omega_0(t)$ denote the leader's speed. Let $u_i(t)$ be the follower's control input. The leader's control input is $u_0(t)$. To achieve consensus, we create a feedback control law u_i for each follower that makes use of

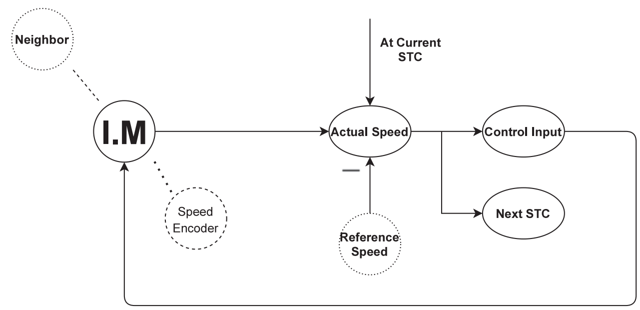

Fig. 2. Proposed control algorithm.

the information obtained over the communication network at sampling time t_k, $k = 0, 1, 2, \ldots..$, and we suppose that for all the initial conditions, $\lim_{t\to\infty}(\omega_i(t) - \omega_0(t)) = 0, i \in [1, N]$ and $\lim_{t\to\infty}(\omega_0(t) - \omega_{\text{ref}}(t)) = 0$, where ω_{ref} is the reference speed. In addition, we assume that between any two consecutive control updates, the input of each motor is held constant in a zero-order-hold fashion equal to the last control update [8]

$$\Delta\omega_i(t_k^i) = \frac{1}{|N_i|}\sum_{m\in N_i}\left(\omega_m(t_k^m) - \omega_i(t_k^i)\right)\Delta t_k^i \tag{14}$$

where $\Delta\omega_i$ is the difference between each motor's speed and the average speed of its neighbors at time t_k^i. N_i is the number of connected motors for the consensus, and Δt_k^i denotes the sampling time difference between two communication occurrences of motor i. By utilizing distinct time instances for sampling speed, the distributed STC can be implemented asynchronously. To provide centralized control, all the motors must have identical sampling instances, i.e., $t_k^m = t_k^i \forall m$. Assuming that L_i is the ith row vector of a Laplacian matrix, then (14) can be expressed as

$$\frac{\Delta\omega_i}{\Delta t_k^i} = -\frac{1}{|N_i|}L_i\boldsymbol{\omega}. \tag{15}$$

To simplify the notation, we may use $\tilde{L}_i = \frac{1}{|N_i|}L_i$, which is called the normalized Laplacian. Therefore, the speed control update (15) will become

$$\dot{\omega} = -\tilde{L}_i\omega. \tag{16}$$

We will also consider a system with $\dot{\omega} = u$, with the control law (16) and assume that the communication graph G is connected. Then, all the motors will eventually converge to the reference speed, i.e., $\lim_{t\to\infty}\omega_i(t) = \omega_{\text{ref}}$ for all $i \in N$, where ω_{ref} is the reference speed.

C. Control Input Updates

As discussed in Section V-B, the proposed model's control algorithm is depicted in Fig. 2, in which the motor receives the speed of the neighboring motor and, at its triggering time, calculates its next communication time and the control input it must provide until the next STC time. The control input for the follower i is u_i, while the leader's control input is u_0. We need $\lim_{t\to\infty}\omega_0(t) = \omega_{\text{ref}}$ to reach consensus with the leader following a reference value, ω_{ref}, which is the desired speed.

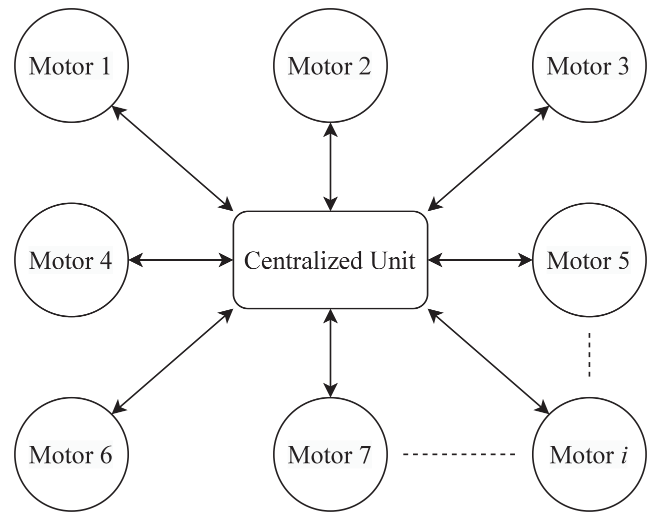

Fig. 3. Centralized topology for consensus.

By reaching consensus with the leader, the followers will finally reach a steady state at the leader's speed [25].

The control input update equation for the followers is

$$u_i(t) = -\sum_{k\in N^f}(\omega_i - \omega_k) - \sum_{k\in N^l}(\omega_i - \omega_k) \tag{17}$$

where N^f represents the follower set and N^l represents the leader set.

Similarly, the control dynamics for the leaders set are

$$u_{0l}(t) = -\sum_{k\in N^f}(\omega_i - \omega_k) - \sum_{k\in N^l}(\omega_i - \omega_k) + (\omega_{\text{ref}} - \omega_i)\,. \tag{18}$$

VI. Centralized STC

To achieve consensus on the desired speed, as described in Section V-C, all the motors exchange status or speed information with their connected neighbors. We assume that connections are established properly and that no packets are lost. In addition, we assume that at each triggering instant, all the motors broadcast their state and all the motors simultaneously receive information from neighboring motors. Motors calculate their control input based on the received information, which is supplied to the motor and is held constant in a zero-order-hold fashion until the next triggering instant. In addition, we assume that there is a central unit, which could be any of the motors, that is connected to all the other motors, as illustrated in Fig. 3. At triggering time, the central unit will have the state information of all the motors and will calculate the next triggering event based on that information [11], [17].

To implement the centralized configuration, we need to identify the time difference $\Delta t = t_{k+1} - t_k$ between two time instances when motors communicate so that the system remains stable. To verify the stability of centralized control, we consider a Lyapunov function $V(t) = \frac{1}{2}\omega^T(t)\tilde{L}\omega$, and for a stable system, we will take the derivative of $V(t)$, which needs to be negative [26]

$$\dot{V}(t) = \omega^T(t)\tilde{L}\dot{\omega}(t). \quad (19)$$

Now, using the preposition $\dot{\omega}(t) = u_\omega(t)$ mentioned earlier, and combining with (16), we can rewrite (19) as

$$\dot{V}(t) = -\omega^T(t)\tilde{L}\tilde{L}\omega(t_k). \quad (20)$$

The error in speed variable with respect to the last control action is defined as $e(t) = \omega(t_k) - \omega(t)$ for $t \in [t_k, t_{k+1})$ and $e(t) = [e_1\ e_2 \ldots\ e_N]^T$. Substituting $\omega(t_k)$ from the error equation into (20), we get

$$\dot{V}(t) = -||\tilde{L}\omega||^2 - \omega^T(t)\tilde{L}\tilde{L}e(t) \quad (21)$$

where $||\cdot||$ is the Euclidean norm. For the system to be Lyapunov stable, $\dot{V}(t)$ should be negative semidefinite by making $e(t)$ to satisfy

$$||e(t)|| \leq \eta\frac{||\tilde{L}\omega||}{||\tilde{L}||} \quad (22)$$

where η is the scaling coefficient and $\eta \in (0, 1)$, which makes $\dot{V}(t)$ in (21) negative semidefinite for $\eta < 1$ [26].

In STC, the next time instant is calculated at the current time instant. With $t \in [t_k, t_{k+1})$, we can define $\omega(t_{k+1}) = -\tilde{L}\omega(t_k)\Delta t_k + \omega(t_k)$, where $\Delta t_k = t - t_k,\ t \in [t_k, t_{k+1})$. By using this definition, we can calculate $e(t)$, which is equal to $\tilde{L}\omega(t_k)\Delta t_k$, allowing us to rewrite (22) as

$$||\tilde{L}\omega(t_k)||\Delta t_k \leq \eta\frac{||-\tilde{L}^2\omega(t_k)\Delta t_k + \tilde{L}\omega(t_k)||}{||\tilde{L}||}. \quad (23)$$

Taking the square on both sides, we get

$$||\tilde{L}\omega(t_k)||^2||\tilde{L}^2(\Delta t_k)^2 \leq \eta^2||-\tilde{L}^2\omega(t_k)\Delta t_k + \tilde{L}\omega(t_k)||^2. \quad (24)$$

The upper bound t, which will become t_{k+1} when the next triggering occurs, can be obtained by solving (24) and setting $\Delta t_k = t_{k+1} - t_k$. Without loss of generality, we can convert the inequality (24) into an equality to obtain the upper bound

$$(\Delta t_k)^2 = \eta^2\frac{||-\tilde{L}^2\omega(t_k)\Delta t_k + \tilde{L}\omega(t_k)||^2}{||\tilde{L}\omega(t_k)||^2||\tilde{L}||^2} \quad (25)$$

which can be further simplified to

$$(\Delta t_k)^2 = \frac{\eta^2(||\tilde{L}^2\omega(t_k)||^2(\Delta t_k)^2 + ||\tilde{L}\omega(t_k)||^2 - 2\tilde{L}^2\omega(t_k)\tilde{L}\omega(t_k)\Delta t_k)}{||\tilde{L}\omega(t_k)||^2||\tilde{L}||^2}. \quad (26)$$

To solve for Δt_k, we use (26), and setting $\Delta t_k = t_{k+1} - t_k$, we obtain

$$t_{k+1} = t_k + \frac{-2\eta^2||\tilde{L}^2\omega(t_k)||^2||\tilde{L}\omega(t_k)|| \pm \sqrt{\delta}}{2(||\tilde{L}\omega(t_k)||^2||\tilde{L}||^2 - \eta^2||\tilde{L}^2\omega(t_k)||^2)} \quad (27)$$

where δ can be defined as

$$\delta = 2\eta^2||\tilde{L}^2\omega(t_k)||^2||\tilde{L}\omega(t_k)|| - 4||\tilde{L}^2\omega(t_k)||^2||\tilde{L}^2|| + \eta^3||\tilde{L}^2\omega(t_k))||^2||\tilde{L}\omega(t_k)||^2.$$

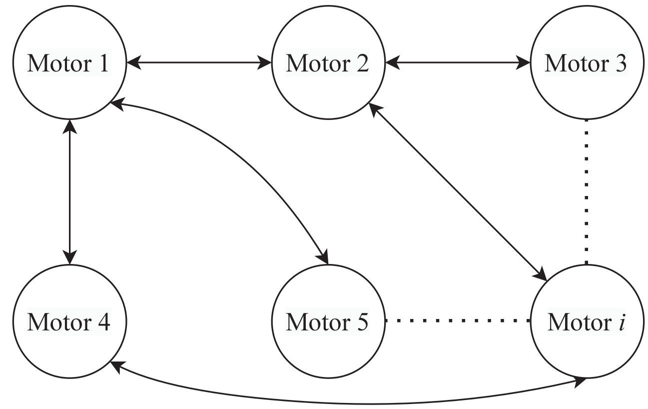

Fig. 4. Distributed topology for consensus.

In our proposed centralized STC, we assume that the minimum intersampling time is bounded away from zero, so that the STC sampler does not produce zero as the next sampling interval at a particular time. Similar to [11], we can prove that by taking the time derivative of $(||e||/||\tilde{L}\omega||)$, and we have

$$\frac{d}{dt}(||e||/||\tilde{L}\omega||) \leq (1 + (||\tilde{L}||||e||/||\tilde{L}\omega||))^2. \quad (28)$$

Now, $p = (||e||/||\tilde{L}\omega||)$ implies that

$$\dot{p} \leq (1 + ||\tilde{L}||p)^2 \quad (29)$$

so that p satisfy the bound $p(t) \leq \phi(t, \phi_0)$, where $\phi(t, \phi_0)$ is obtained by

$$\dot{\phi} = (1 + ||\tilde{L}||\phi)^2 \quad (30)$$

with $\phi(0, \phi_0) = \phi_0$. Therefore, the minimum intersampling time is bounded by τ, which satisfies $\phi(\tau, 0) = (\eta/||\tilde{L}||)$. The solution to (28) is $\phi(\tau, 0) = (\tau/1 - \tau||\tilde{L}||)$, so that $\tau = (\eta/||\tilde{L}||(1 + \eta))$.

VII. Distributed STC for Motors

In distributed STC, in contrast to centralized control, we do not require a central control unit to manage information exchange. Refer to Fig. 4 for an illustration of a possible distributed configuration. In this distributed situation, a motor communicates only with its connected neighbors. The motor will compute the next triggering time itself based on state information it obtains locally through communication with neighbors. The following time instances at which motor i updates are denoted by $t_0^i, t_1^i, \ldots.., t_k^i, \ldots,$. The motor's speed measurement error is as follows:

$$e_i(t) = \omega_i(t_k^i) - \omega_i(t) \quad (31)$$

where $t \in [t_k^i, t_{k+1}^i)$. The control input update for motor i using the distributed scenario is defined as

$$u_i(t) = \sum_{n \in N_i} \omega_n(t^n_{k_{n(t)}}) - \omega_i(t_k^i). \quad (32)$$

To keep things simple, we suppose that the next triggering instance for all the motors is the minimum time instance, i.e., in

(32), $t^n_{k_{n(t)}} = \min_{\{s\in\mathbb{N}:t^n_s\le t\}}(t - t^n_s)$, where $\mathbb{N}$ denotes the set of natural numbers. In other words, control updates $u_i(t)$ are computed not only at their own updating times $t^i_0, t^i_1, \ldots$, but also at their neighbors' updating times, $t^n_0, t^n_1, \ldots$,. Given that $e(t) = \omega(t_k) - \omega(t)$, it follows

$$\omega_n(t^n_{k_{n(t)}}) = \omega_n(t) + e_n(t). \tag{33}$$

Let $x_i(t) = \tilde{L}_i\omega(t)$, which can be written as

$$x_i(t) = \omega_i(t) - \frac{1}{|N_i|}\sum_{n\in N_i}\omega_n(t). \tag{34}$$

Combining $x_i(t)$ with (19), we obtain

$$\dot{V}(t) = -\sum_i x_i^2(t) - \sum_i\sum_{n\in N_i} x_i(t)\left(e_i(t) - e_m(t)\right)$$
$$= -\sum_i\left(x_i^2(t) + |N_i|x_i(t)e_i(t)\right) + \sum_i\sum_{n\in N_i} x_i(t)e_m(t). \tag{35}$$

With Young's inequality $|xy| \le (a/2)x^2 + (1/2a)y^2$, for $a > 0$, we get

$$\dot{V}(t) \le -\sum_i\left(x_i^2(t) - a|N_i|x_i^2(t)\right) + \sum_i\frac{|N_i|}{2a}e_i^2(t)$$
$$+ \sum_i\sum_{n\in N_i}\frac{1}{2a}e_n^2(t). \tag{36}$$

Since matrix $\tilde{L}$ is symmetric, the last term in (36) can be written as $\sum_i\sum_{n\in N_i}\frac{1}{2a}e_n^2(t) = \sum_i\frac{|N_i|}{2a}e_i^2(t)$. For the system to be stable, the derivative of the Lyapunov function should be negative, which means that (36) needs to be upper bounded by 0. Assume that $0 < a < \frac{1}{|N_i|}$ for all $i \in N$, and introducing a scaling coefficient η_i, $0 \le \eta_i \le 1$, the expression in (36) can be rearranged as

$$e_i^2(t) \le \eta_i\frac{a(1 - a|N_i|)}{|N_i|}x_i^2(t). \tag{37}$$

Since $\dot{\omega}_{i(t)} = \frac{1}{|N_i|}\sum_{n\in N_i}(\omega_n(t^n_{k_{n(t)}}) - \omega_i(t^i_k))$, it can be simplified to

$$\omega_i(t) = -\frac{1}{|N_i|}\sum_{n\in N_i}(\omega_i(t^i_k) - \omega_n(t^m_{k_{n(t)}}))(\Delta t^i_k) + \omega_i(t^i_k). \tag{38}$$

In (38), $\Delta t^i_k = t - t^i_k$ for $t \in [t^i_k, t^i_{k+1})$. Let us now define $\mu_i = \frac{1}{|N_i|}\sum_{n\in N_i}(\omega_i(t^i_k) - \omega_n(t^n_{k_{n(t)}}))$; substituting (38) into (34), we obtain

$$x_i(t) = -\frac{1}{|N_i|}\sum_{n\in N_i}\Big(-\mu_n(t - t^n_{k_{n(t)}}) + \omega_i(t^n_{k_{n(t)}})$$
$$+ \mu_i\Delta t^i_k - \omega_i(t^i_k)\Big)$$
$$= -\mu_i\Delta t^i_k + \frac{1}{|N_i|}\sum_{n\in N_i}(\omega_i(t^i_k) - \omega_i(t^n_{k_{n(t)}})$$
$$+ \frac{1}{|N_i|}\sum_{n\in N_i}\mu_n(t - t^i_k + t^i_k - t^n_{k_{n(t)}}) \tag{39}$$

which can be simplified to

$$x_i(t) = \left(\frac{1}{|N_i|}\sum_{n\in N_j}\mu_n - \mu_i\right)\Delta t^i_k + \mu_i$$
$$+ \frac{1}{|N_i|}\sum_{n\in N_i}\mu_n(t^i_k - t^n_{k_{n(t)}}). \tag{40}$$

Now, define $\Theta_i = \mu_i + \frac{1}{|N_i|}\sum_{n\in N_i}\mu_n(t^i_k - t^n_{k_{n(t)}})$ and $\mu' = \frac{1}{|N_i|}\sum_{n\in N_i}(\mu_n - \mu_i)$; substituting (40) into (37), we obtain

$$e_i^2(t) \le \eta_i\frac{a(1 - a|N_i|)}{|N_i|}(\mu'_i\Delta t^i_k + \Theta_i)^2. \tag{41}$$

Denote $\rho_i = \frac{a(1-a|N_i|)}{N_i}$ and using $e_i(t) = \omega_i(t) - \omega_i(t^i_k)$, (41) then becomes

$$(\omega_i(t) - \omega_i(t^i_k))^2 \le \eta_i\rho_i(\mu'_i\Delta t^i_k + \Theta_i)^2. \tag{42}$$

Using (38), we can rewrite (41) as

$$(\mu_i(\Delta t^i_k))^2 \le \eta_i\rho_i(\mu'_i\Delta t^i_k + \Theta_i)^2. \tag{43}$$

Rearranging yields

$$(\mu_i^2 - \eta_i\rho_i{\mu'}^2_i)(\Delta t^i_k)^2 - 2\eta_i\rho_i\Theta_i\mu'_i(\Delta t^i_k) \le \eta_i\rho_i\Theta_i^2. \tag{44}$$

To obtain the upper bound, we use $\Delta t^i_k = t - t^i_k$ for $t \in [t^i_k, t^i_{k+1})$, and using the equality in (44), we get

$$t_{k+1} = t_k + \frac{(\eta_i\rho_i)^{1/2}\Theta_i(\eta_i\rho_i)^{1/2}\mu'_i \pm \mu_i}{\mu_i^2 - \eta_i\rho_i\mu_i^2}. \tag{45}$$

To avoid Zeno behavior, we may use the minimum time sample τ

$$\Delta t^i_k = \max\left\{\tau, \frac{(\eta_i\rho_i)^{1/2}\Theta_i}{\mu_i - (\eta_i\rho_i)^{1/2}\mu'_i}, \frac{-(\eta_i\rho_i)^{1/2}\Theta_i}{\mu_i + (\eta_i\rho_i)^{1/2}\mu'_i}\right\}. \tag{46}$$

VIII. Case Study

We chose a production line in a car manufacturing factory as a use case to demonstrate the applicability and performance of our proposed self-triggered consensus control approach. In a normal production line, numerous motors must agree on their speed in order to operate a conveyor belt or a steel press, for example. Assume that we have a single leader motor that provides a stable reference speed. We suppose that data are conveyed from the reference to the leader. In addition, we suppose that we have three followers to develop consensus based on their speed as well as the leader's speed and attempt to match the leader's speed. The control network's topology is depicted in Fig. 5.

We assume that the reference speed is 3600 revolutions per minute (RPM) initially, but is reduced to 1800 RPM after a period of time due to load changes and then increased to 7200 RPM after another period of time.

$f_1, f_2,$ and f_3 are set to 2000, 4000, and 6000, RPM, respectively, with the leader's initial speed set at 8000 RPM. Because our proposed method imposes no restrictions on the motor's initial speeds, the motor can start from zero initial values as well.

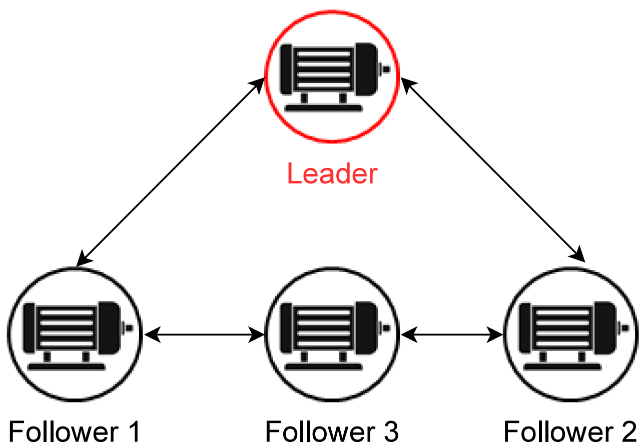

Fig. 5. Connection topology of motors.

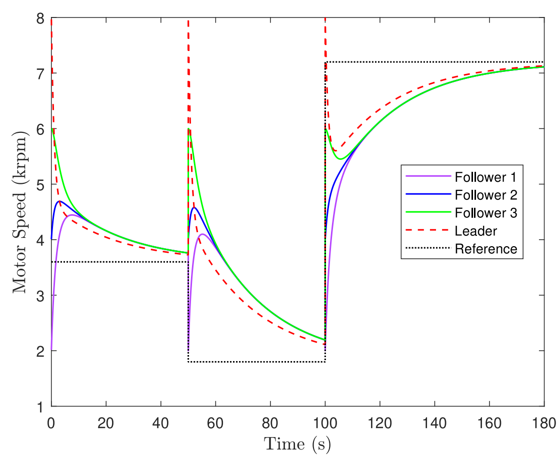

Fig. 6. Speed consensus of motors using periodic communication.

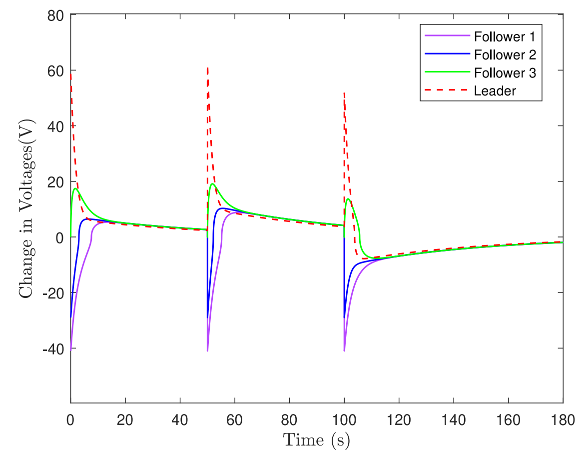

Fig. 7. Change in voltages applied as an input to the motors.

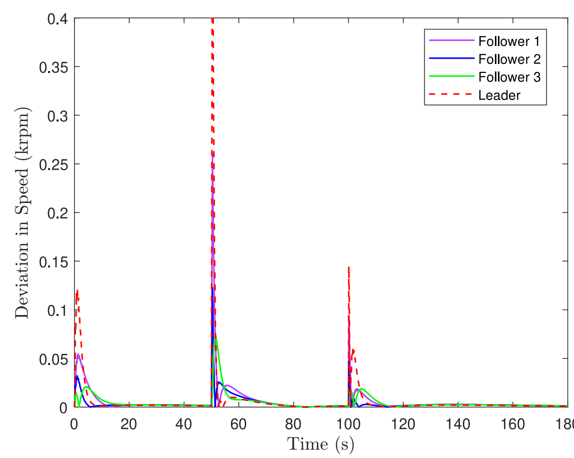

Fig. 8. Deviation in the convergence speed of centralized STC with a periodic approach.

A. Periodic Scenario

In the following evaluation of the consensus control approach, we assume that the motors communicate on a time-periodic basis and that the sample time is fixed to 0.01 s. The motors' speed consensus is depicted in Fig. 6.

As illustrated in Fig. 6, the motors began at different speeds and attempted to achieve a consensus. To achieve consensus, they must all attain the same value, referred to as the reference speed. With all the followers observing the leader's speed, the leader's speed converges to the reference value. After a period of time, load disturbances alter the motor speed requirements for the reference and, hence, the reference value. The leader attempts to re-establish the reference value and build consensus among the followers. This operation is repeated until a steady state is reached in which all the motors operate at the same speed as the reference value and maintain that speed. As a result of the change in torque caused by the reference shift, the flux changes, affecting the current produced and, hence, the voltages provided to the motor, as illustrated in Fig. 7.

B. Centralized STC Scenario

We assume centralized STC in the following, with one of the follower motors acting as a central unit. Here, motor 2 is designated as the central unit, which will include information on the speed of the other motors. Motor 2 will compute the next trigger time for all the motors based on the information it has from all the motors. This is achieved by all the motors broadcasting their speed to their neighbors and changing their own control input based on the speed information of their neighbors. The convergence speed of centralized STC is compared to that of a periodic control signal exchange. The speed deviation is depicted in Fig. 8.

As depicted in Fig. 8, the absolute deviation in the speed of all the followers and the leader is extremely close to zero in steady state, but there are two spikes in the deviation caused by system transients.

C. Distributed STC Scenario

We assumed in the previous section that a central control unit was responsible for computing the next triggering time based on the speed data from all the motors. This configuration does not alleviate the single point of failure issue. The system would fail if any of the motors failed to convey the speed information. To avoid this, we designed a distributed STC, in which there is no central unit and each motor calculates its own triggering time

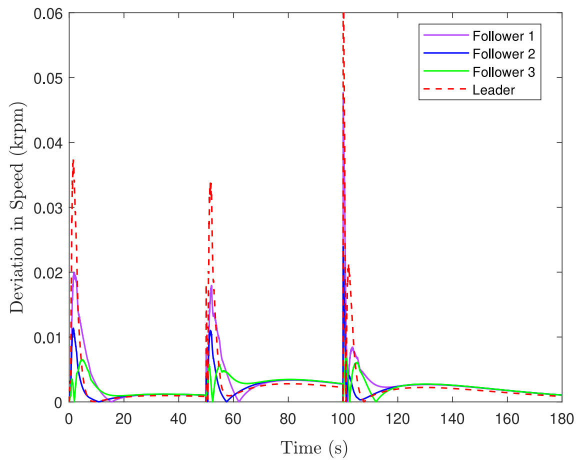

Fig. 9. Deviation in the convergence speed of distributed STC with a periodic approach.

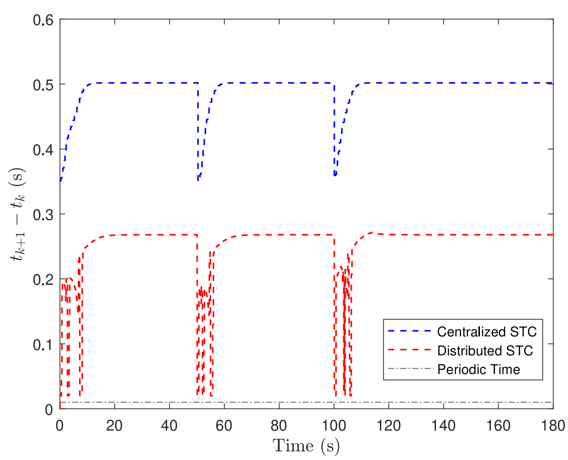

Fig. 10. Intersampling times for centralized, distributed STC, and periodic approaches.

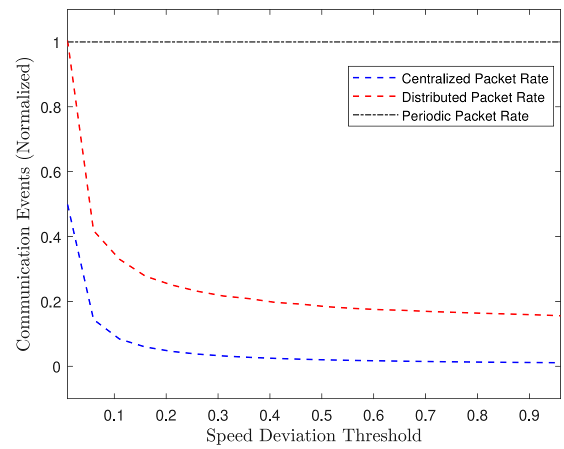

Fig. 11. Number of normalized communication events for centralized and distributed STCs compared to periodic communication.

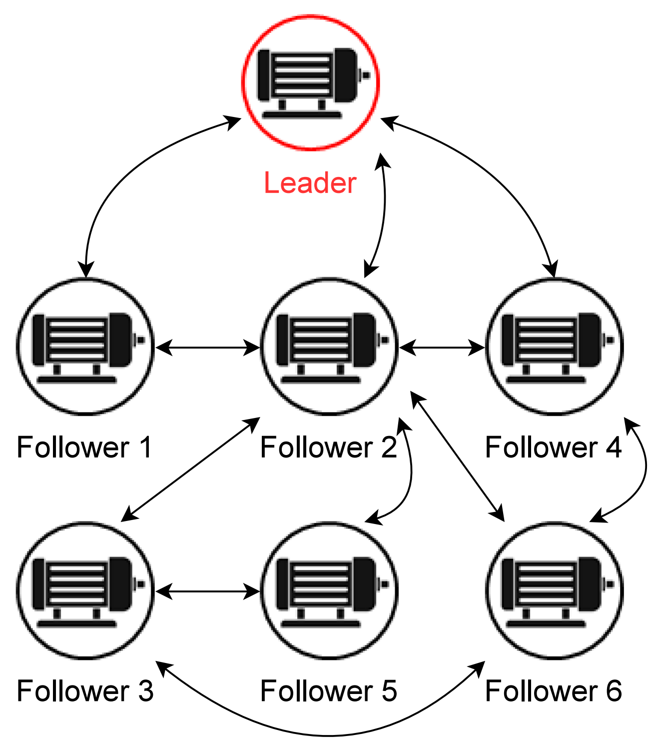

Fig. 12. Scaled connection of motors.

separately, but each motor will trigger at the minimum time of all the motors. We utilize the same parameters for distributed STC as we do for centralized STC. To avoid Zeno behavior, we utilize τ as minimum intersampling time as in (46). The speed difference between the periodic and distributed techniques is illustrated in Fig. 9 .

Both the centralized and distributed systems have a different triggering time than a periodic setup. We have a specified sampling time in the periodic configuration, and the system will trigger and communicate at that time. On the other hand, in centralized and distributed STC, the system communicates in response to system events, as seen in Fig. 10. As illustrated in Fig. 10, the sampling period increases as the system approaches its steady state and drops as the system experiences transients.

The message exchanges needed for centralized and distributed approaches as well as for the periodic approach are compared. As illustrated in Fig. 11, centralized STC requires fewer messages to communicate the motors' state than distributed STC. This is because the distributed technique allows for more communication, as each motor communicates at the synchronous sampling time. In addition, as η increases, the normalized message rate for both the centralized and distributed STCs continues to decrease, but the message communication requirements for periodic communication remain constant, resulting in a higher rate than for both the centralized and distributed STCs.

As illustrated in Figs. 10 and 11, our STC model achieves much better results than the periodic approach. We observe that STC increases communication interval time, resulting in fewer message communication events for both the STC models.

D. Scaled System

The preceding results assumed a single leader and three followers, but we are now interested in the effect on the overall

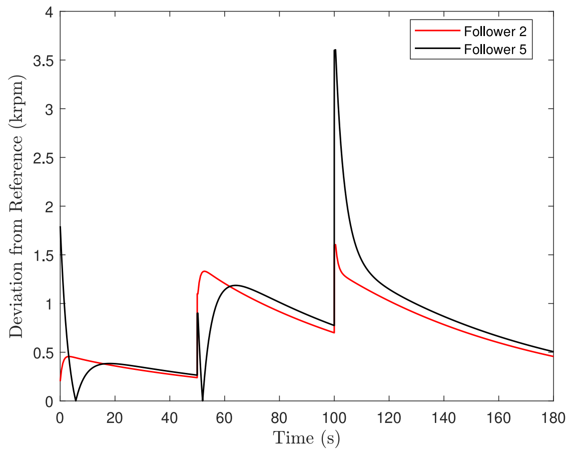

Fig. 13. Deviation in speed (of follower motor which is directly connected to the leader versus follower motor, which is not directly connected) from the reference speed.

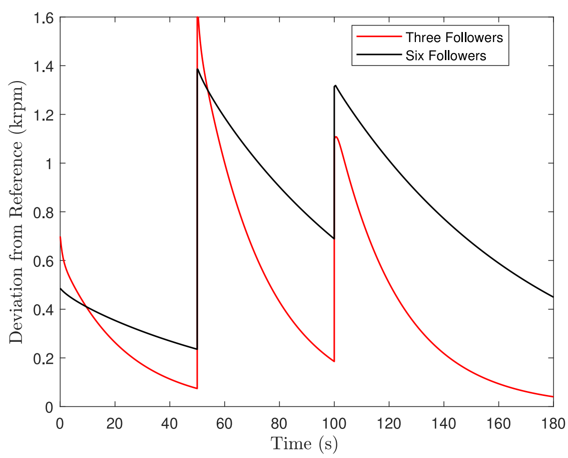

Fig. 14. Deviation in speed (average of followers and leader) from the reference speed of original (small) versus scaled system set.

performance of increasing the number of motors. Assume now that there are six follower motors and a leader, as seen in Fig. 12.

As can be seen in this example, followers 1, 2, and 4 are all directly connected to the leader motor, while the other follower motors are not. The motors that are not directly connected to the leader will need to reach an agreement with the motors that are directly connected. We notice the deviation in speed from the reference value of follower 2, which is directly connected to the leader, and follower 5, which is not directly connected to the leader, as shown in Fig. 13. To keep things simple, and because periodic, centralized, and distributed STCs all produce identical results, we have just shown the findings for centralized STC. As illustrated in Fig. 13, follower 2 has a lower deviation in speed from the reference value than follower 5, which receives the leader's speed indirectly.

In addition, we examine the effects of increasing the number of nodes on the overall performance of the system. We compare the systems depicted in Figs. 5 and 12. We will average the speeds of all the followers and leader motors and compare them

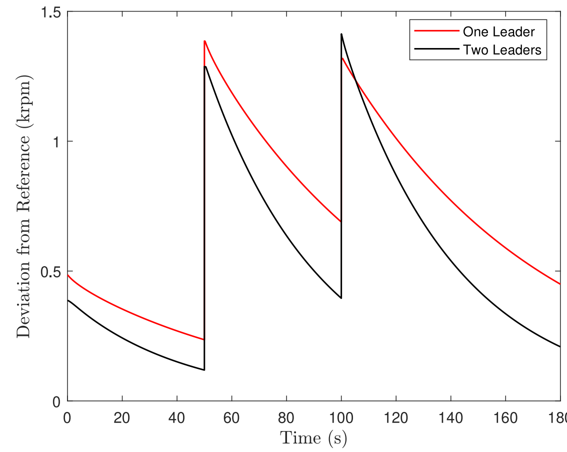

Fig. 15. Deviation in speed (average of followers and leader) from the reference speed of a system with one leader versus a system with two leaders.

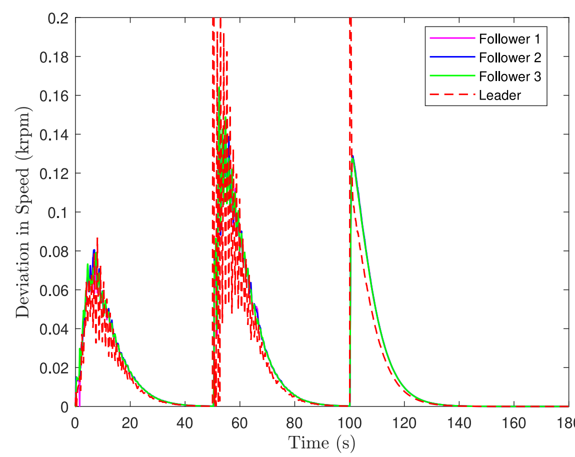

Fig. 16. Deviation in speed with perfect communication and with packet loss of 30%.

to the reference value, as well as observe the effect of scaling the system. As illustrated in Fig. 14 , when we have fewer followers, the average deviation of speed from the reference is lower than when we have a large system. This is because a few nodes that are not directly connected to the leader diverge more from the reference.

Finally, we increase the number of leaders from 1 to 2 to see what influence this has on performance. We now consider a system with six followers and two leaders. As illustrated in Fig. 15, increasing the number of leaders greatly reduces the deviation in speed from the reference value. This is because the system will converge faster if there are more leaders with access to the reference value.

E. Impact of Packet Loss

So far, we have assumed that the connection between the motors is perfect and that no information is lost during communication. In practice, if we use a wireless communication channel, communication will not be perfect, and there is a possibility of loss. To the best of our knowledge, no study has been conducted

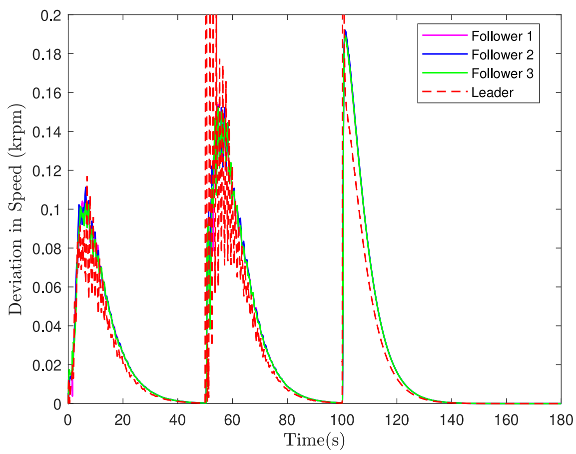

Fig. 17. Deviation in speed with perfect communication and with packet loss of 50%.

on the effect of message loss on self-triggered communication for consensus algorithms, and we are attempting to determine the effect of such loss on speed control. In Fig. 16, we average a random packet loss of 30% across five simulation runs and then calculate the absolute deviation from the speed with perfect communication. In addition, we observe the effect of 50% packet loss on the speed using the same configuration, as described previously in Fig. 17. During transients, packet loss will slow down the system, but we will still reach a consensus and the system will stay stable.

IX. Conclusion

In this article, we devised a consensus control method for a network of induction motors built on leader–follower relationships in order to accomplish a common goal. Each motor used IFOC speed control based on a set point furnished through the leader–follower mechanism. In prior work, a consensus control approach based on periodic communication was presented. However, in order to reduce communication bandwidth requirements, we proposed a need-based STC method that achieves both message exchange frequency reduction and bandwidth conservation. We developed our STC using both the centralized and distributed approaches. While centralized STC requires less communication than distributed STC, it cannot address the single point of failure issue of the central controller, which does not occur with distributed STC. We created a synchronous distributed STC to avoid Zeno behavior. The proposed method was subsequently validated through the use of a case study involving a network of industrial induction motors, such as those used in conveyor belts or steel presses. Our case study showed that both the centralized and distributed STCs are viable options, and that each has their own benefits.

The technique outlined in this article is based on perfect communication. We have briefly studied the impact of communication over an unreliable wireless channel. From the simulation results, we find that packet loss does affect the performance of the system. The communication between the transmitter and the receiver also requires some time, making it challenging to have an asynchronous model, as the actuator might not be able to distinguish between packet loss and whether an agent has not transmitted information at all. Therefore, we use a synchronous model to ease the burden on the actuator. We plan to evaluate the use of an asynchronous model in the future. If control information packets are lost because of wireless communication, we plan to make up for the loss with techniques like regression or machine learning algorithms.

References

[1] D.-K. Hong, J.-H. Choi, D.-J. Kim, Y.-D. Chun, B.-C. Woo, and D.-H. Koo, "Development of a high speed induction motor for spindle systems," *IEEE Trans. Magn.*, vol. 49, no. 7, pp. 4088–4091, Jul. 2013.

[2] D. Liang and V. Zhou, "Recent market and technical trends in copper rotors for high-efficiency induction motors," in *Proc. Int. Power Electron. Conf.*, 2018, pp. 1943–1948.

[3] W. Li, Z. Xu, and Y. Zhang, "Induction motor control system based on FOC algorithm," in *Proc. IEEE 8th Joint Int. Inf. Technol. Artif. Intell. Conf.*, 2019, pp. 1544–1548.

[4] T. Matsuo, K. Sugahara, A. Kameari, and Y. Shindo, "Model order reduction of an induction motor using a Cauer ladder network," *IEEE Trans. Magn.*, vol. 56, no. 3, Mar. 2020, Art. no. 7514704.

[5] A. K. Singh, A. Dalal, R. Roy, and P. Kumar, "Improved dynamic model of induction motor including the effects of saturation," in *Proc. IEEE Int. Conf. Power Electron., Drives Energy Syst.*, 2014, pp. 1–5.

[6] G. Pasolini, D. Dardari, and M. Kieffer, "Exploiting the agent's memory in asymptotic and finite-time consensus over multi-agent networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 479–490, 2020.

[7] 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.

[8] H. R. Chamorro, M. Nazari, D. Babazadehi, N. R. Malik, and M. Ghandhari, "Consensus control for induction motors speed regulation," in *Proc. IEEE 16th Eur. Conf. Power Electron. Appl.*, 2014, pp. 1–6.

[9] X. Meng and C. Xu, "Multi-agent leader-follower tracking over random structurally balanced networks," *IEEE Access*, vol. 7, pp. 109474–109481, 2019.

[10] X. Zhang, W. Han, and X. Liu, "Bipartite tracking consensus of linear multi-agent systems with a dynamic leader under signed digraph," *IET Control Theory Appl.*, vol. 14, no. 15, pp. 2127–2133, 2020.

[11] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.

[12] Y. Wu, M. Xu, N. Zheng, and X. He, "Event-triggered resilient consensus for multi-agent networks under deception attacks," *IEEE Access*, vol. 8, pp. 78121–78129, 2020.

[13] W. Zou, P. Shi, Z. Xiang, and Y. Shi, "Consensus tracking control of switched stochastic nonlinear multiagent systems via event-triggered strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 1036–1045, Mar. 2020.

[14] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 3270–3285.

[15] X. You, C. Hua, and X. Guan, "Self-triggered leader-following consensus for high-order nonlinear multiagent systems via dynamic output feedback control," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2002–2010, Jun. 2019.

[16] Z. Ijaz, M. Tahir, and S. Arshad, "Self-triggered control plane for cognitive radio networks," in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.

[17] M. Tahir and S. K. Mazumder, "Self-triggered communication enabled control of distributed generation in microgrids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 441–449, Apr. 2015.

[18] J. Zhao, L. Xiang, F. Chen, and W. Lan, "Distributed output-based self-triggered control for general linear multi-agent systems," in *Proc. IEEE 27th Chin. Control Decis. Conf.*, 2015, pp. 285–290.

[19] Y. Xie and Z. Lin, "Event-triggered global leader-following consensus for multi-agent systems with bounded controls," in *Proc. IEEE 36th Chin. Control Conf.*, 2017, pp. 8413–8418.

[20] W. Zou and Z. Xiang, "Event-triggered leader–following consensus of nonlinear multi-agent systems with switched dynamics," *IET Control Theory Appl.*, vol. 13, no. 9, pp. 1222–1228, 2018.

[21] J. Zhang, H. Zhang, Y. Lu, and S. Sun, "Event\self-triggered leader-following consensus of multi-agent systems with general dynamics," *IET Control Theory Appl.*, vol. 14, no. 9, pp. 1209–1219, 2020.

[22] W. Zhu and Z.-P. Jiang, "Event-based leader-following consensus of multi-agent systems with input time delay," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1362–1367, May 2015.

[23] B. Cheng and Z. Li, "Coordinated tracking control with asynchronous edge-based event-triggered communications," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4321–4328, Oct. 2019.

[24] B. K. Bose, *Modern Power Electronics and AC Drives*, vol. 123. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[25] W. Liu, Q. Wu, S. Zhou, and G. Yin, "Leader-follower consensus control of multi-agent systems with extended Laplacian matrix," in *Proc. IEEE 27th Chin. Control Decis. Conf.*, 2015, pp. 5393–5397.

[26] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

Zohaib Ijaz received the bachelor's degree in electronic engineering from the Islamia University of Bahawalpur, Bahawalpur, Pakistan, in 2014, and the master's degree in electrical engineering with a specialization in control systems from the University of Engineering and Technology, Lahore, Pakistan, in 2017. He is currently working toward the Ph.D. degree with the School of Computer Science, University College Cork, Cork, Ireland, with Science Foundation Ireland funding from the Confirm Centre for Smart Manufacturing.

His research interests include event-based communication for cyber-physical systems and consensus control of multiagent systems, such as synchronous motors and power systems.

Md. Noor-A-Rahim received the Ph.D. degree from the Institute for Telecommunications Research, University of South Australia, Adelaide, SA, Australia, in 2015.

He is currently a Senior Researcher with the School of Computer Science and Information Technology, University College Cork, Cork, Ireland. He was a Postdoctoral Research Fellow with the Centre for Infocomm Technology, Nanyang Technological University, Singapore. His research interests include intelligent transportation systems, machine learning, Internet of Things, and signal processing.

Dr. Noor-A-Rahim was a recipient of the Michael Miller Medal from the Institute for Telecommunications Research, University of South Australia, for the most outstanding Ph.D. thesis in 2015.

Dirk Pesch (Senior Member, IEEE) received the Dipl.Ing. degree from RWTH Aachen University, Aachen, Germany, and the Ph.D. degree from the University of Strathclyde, Glasgow, Scotland.

He is currently a Professor with the School of Computer Science and Information Technology, University College Cork, Cork, Ireland. He was previously the Head of the Nimbus Research Centre, Cork Institute of Technology, Cork. He has a research and development experience of more than 25 years in both industry and academia and has (co)authored more than 200 scientific articles and book chapters. He is a principle investigator of the National Science Foundation Ireland funded collaborative centers CONNECT (Future Networks) and CONFIRM (Smart Manufacturing) and the Director of the Science Foundation Ireland Centre for Research Training in Advanced Networks for Sustainable Societies. He has also been involved in a number of European Union funded research projects on smart and energy-efficient buildings and urban neighborhoods, including as coordinator. His research interests include problems associated with architecture, design, algorithms, and performance evaluation of low-power, dense, and vehicular wireless/mobile networks and services for Internet of Things and cyber-physical system's applications in building management, smart connected communities, independent living, and smart manufacturing.

Dr. Pesch was the Technical Program Chair of 2020 IEEE International Symposium on World of Wireless Mobile and Multimedia Networks and the Executive Vice-Chair of 2020 IEEE International Conference on Communications.