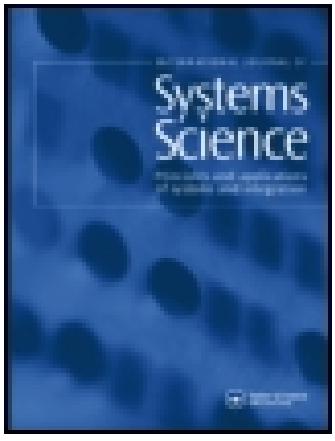

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Stability of formation control using a consensus protocol under directed communications with two time delays and delay scheduling

Rudy Cepeda-Gomez[a] & Nejat Olgac[b]
[a] Faculty of Mechatronic Engineering, Universidad Santo Tomás, Bucaramanga, Colombia
[b] Department of Mechanical Engineering, University of Connecticut, Storrs, CT, USA
Published online: 13 Mar 2014.

International Journal of Systems Science, 2014
http://dx.doi.org/10.1080/00207721.2014.886745

Stability of formation control using a consensus protocol under directed communications with two time delays and delay scheduling

Rudy Cepeda-Gomez[a,*] and Nejat Olgac[b]

[a]*Faculty of Mechatronic Engineering, Universidad Santo Tomás, Bucaramanga, Colombia;* [b]*Department of Mechanical Engineering, University of Connecticut, Storrs, CT, USA*

(*Received 11 June 2013; accepted 10 December 2013*)

We consider a linear algorithm to achieve formation control in a group of agents which are driven by second-order dynamics and affected by two rationally independent delays. One of the delays is in the position and the other in the velocity information channels. These delays are taken as constant and uniform throughout the system. The communication topology is assumed to be directed and fixed. The formation is attained by adding a supplementary control term to the stabilising consensus protocol. In preparation for the formation control logic, we first study the stability of the consensus, using the recent cluster treatment of characteristic roots (CTCR) paradigm. This effort results in a unique depiction of the non-conservative stability boundaries in the domain of the delays. However, CTCR requires the knowledge of the potential stability switching loci exhaustively within this domain. The creation of these loci is done in a new surrogate coordinate system, called the 'spectral delay space (SDS)'. The relative stability is also investigated, which has to do with the speed of reaching consensus. This step leads to a paradoxical control design concept, called the 'delay scheduling', which highlights the fact that the group behaviour may be enhanced by increasing the delays. These steps lead to a control strategy to establish a desired group formation that guarantees spacing among the agents. Example case studies are presented to validate the underlying analytical derivations.

Keywords: consensus; CTCR; delay scheduling; formation; multi-agent systems; time delay

1. Introduction

The problem of decentralised (distributed) coordination of multi-agent systems is considered here, where the agents aim to agree upon a certain variable of interest. This general problem is treated in a ground-breaking work by Vicsek, Czirók, Ben-Jacob, Cohen, and Shochet (1995), which studies the agents trying to align their headings using discrete-time representation. Later on, Olfati-Saber and Murray (2004) investigated agents driven by first-order dynamics within a formal framework. That study considers directed communication topologies (both fixed and switching type) and introduces some graph theoretical results useful for the stability analysis of such agreement protocols. Under the simplifying features of first-order dynamics, time-delayed communications are also studied in the case of fixed topology. Several other researchers (Cepeda-Gomez & Olgac, 2011a, 2011b; Lin & Jia, 2009a, 2009b; Lin, Jia, Du, & Juan, 2008; Meng, Ren, Cao, & Zheng, 2011; Peng & Yang, 2009; Ren, 2007; Ren & Beard, 2005; Sun & Wang, 2009a, 2009b) have performed further extensions to this work, proposing consensus protocols for agents driven by second-order dynamics, including the study of the agreement problem under time-delayed communications (Cepeda-Gomez & Olgac, 20011a, 2011b; Lin & Jia, 2009a, 2009b; Lin et al., 2008; Meng et al., 2011; Peng & Yang, 2009; Sun & Wang, 2009a, 2009b). Meng et al. (2011) address a case with two time delays. For the stability analysis of such systems with uncertain but fixed time delays, almost all the previous works rely on either Lyapunov–Krasovskii- or Razhumikin-based methodologies (Lin & Jia, 2009a, 2009b; Lin et al., 2008; Meng et al., 2011; Peng & Yang, 2009; Sun & Wang, 2009a, 2009b) or generalised Nyquist criterion (Liu & Tian, 2009; Münz, Papachristodoulou, & Allgower, 2010). All of these treatments provide only sufficient conditions on the delays to achieve asymptotic stability. Consequently, they produce very conservative results, essentially confining the stability bounds within very small delays. Also, because these results are based on some solutions of an linear matrix inequality or graphical analysis, they are always imprecise. Therefore, these methods are, in general, not so practicable.

Differently from the earlier investigations, this paper presents an approach with several distinguishing features. First and most importantly, we follow a unique stability paradigm in this paper. It is based on the combination of a simplifying factorisation procedure over the characteristic equation of the system and the deployment of the crucial stability paradigm, which is called the cluster treatment of characteristic roots (CTCR) (Ergenc, Olgac, & Fazelinia,

* Corresponding author. Email: rudycepedagomez@mail.ustabuca.edu.co

2007; Fazelinia, Sipahi, & Olgac, 2007). This method creates exact, exhaustive and explicit stability regions in the domain of the delays. Second, CTCR provides both necessary and sufficient stability conditions in the domain of the delays. This unique knowledge enables the control designer to manipulate the delays (simply by enlarging them) such that the stability of the dynamics is guaranteed as well as the desired consensus reaching times. This delay manipulation process is named 'delay scheduling' (Olgac & Cavdaroglu, 2011). Liu and Tian (2009) also executes a similar manipulation using the generalised Nyquist method. Their results, therefore, are confined to fixed delay compositions, and short of providing an exhaustive parametric stability map of the system in the delay space which is offered here using CTCR philosophy. Cepeda-Gomez and Olgac (2011a, 2011b) analyses this capability for networks but with undirected topologies. The critical nuance between the undirected and directed cases is in the handling of the complex eigenvalues, which has been declared in the literature as the major hurdle (Münz et al., 2010, p. 1255). Although some earlier investigations have also attempted similar task for directed topologies (Liu & Tian, 2009; Meng et al., 2011), this is the first work, to the best of our knowledge, in which the exact stability boundaries are declared with truly large time delays.

The third distinguishing feature of this paper is in the formation control logic. Most approaches to this problem include control laws based on relative distances and the nearest-neighbour rules (Olfati-Saber, 2006; Ren, 2007), which impose highly nonlinear characteristics. Consequently, these systems make the stability assessment prohibitively complex.

The formation control problem requires an agreement of the agents on some feature(s) of the group behaviour. For instance, Olfati-Saber (2006) presents flocking algorithms which utilise a linear term to achieve a velocity agreement and a distance-based nonlinear term to guarantee separation and cohesion properties. Lin and Jia (2009b) extend a linear consensus algorithm to create formation behaviour by including the inter-agent distances into the control law of each agent. A similar approach is used by Liu and Tian (2009); they use the desired positions of the agents with respect to a common origin in the control logic, and analyse the convergence speed to the formation in the presence of time delay observing simulations of different case studies. For the same objective, in this paper we follow a different approach, which is part of the contributions of the work. While designing the control we take advantage of decoupling properties of the dynamics in modal coordinates introduced by a factorisation procedure. A constant forcing term is deployed on each agent, which provides the desired formation spacing among the agents. This procedure results in a simpler control logic than those previously proposed. Structured steps for the design of that constant forcing term are also presented.

In summary, the combined contributions of this paper over the existing literature are the following. For a group of second-order agents communicating under a directed structure affected by two rationally independent time delays:

(1) A simple new formation control logic is introduced, based on a consensus protocol and a constant inter-agent forcing term. Interestingly, this logic preserves linearity of the dynamics, in contrast to the competing routines.
(2) To assess the stability of the consensus protocol, and of the formation control algorithm under delayed communication schemes, a unique CTCR paradigm is deployed. This process yields an exact (non-conservative) and exhaustive declaration of stable regions in the domain of the delays.
(3) The 'delay scheduling' concept is introduced, which is a direct consequence of Equation (2).

The paper is organised as follows: Section 2 presents the consensus protocol, which is used as the basis for the formation control algorithm; Section 3 is devoted to the stability assessment in the domain of the delays, including a brief description of the CTCR paradigm and spectral delay space (SDS) concept; the extension of the consensus protocol to the formation control algorithm, including the design steps, is the topic of Section 4. Section 5 presents some case studies including the deployment of the delay scheduling concept.

Throughout the text, boldface notation is used for vector quantities, bold capital letters for matrices and italic symbols for scalars.

2. Problem statement

We consider a group of n autonomous agents, which are driven by second-order dynamics given by $\ddot{x}_j(t) = u_j(t)$, $j = 1, 2, \ldots, n$, where $x_j(t) \in \Re$ is taken as the scalar position and $u_j(t) \in \Re$ as the control law. The analysis is presented for a one-dimensional case for simplicity, although all the ensuing results are scalable to higher dimensions using the Kronecker product operation (Schaefer, 1996) as it is common in the literature (Cepeda-Gomez & Olgac, 2011a, 2011b; Lin & Jia, 2009a, 2009b; Lin et al., 2008; Meng et al., 2011). We declare that the consensus is achieved when all n agents approach to the same position, i.e., $\lim_{t\to\infty} x_j(t) - x_k(t) = 0$ for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, n$. Notice that this consensus definition does not impose any a priori restriction over the value of the final position.

We assume that the jth agent receives position and velocity information from a subset of agents, which consists of Δ_j agents, $\Delta_j < n$. The members of this set are called the *informers of the jth agent*, and the set is denoted by N_j. The number of informers is also known as the in-degree of the

jth node. Assuming unidirectional information channels, the communication network is described by a directed graph with n vertices. It is also assumed that all these communication channels have two constant and uniform time delays, τ_1 and τ_2, which affect the position and velocity information exchange, respectively. That is, agent j only knows the τ_1-seconds-earlier position and τ_2-seconds-earlier velocity of its Δ_j informers. The delay structure proposed here is based on a realistic assumption that two different classes of sensors are used for the measurement of position and velocity, which introduce different delays for each sensing channel. Also, considering that the swarm is made of the same type of agents, the selection of uniform time delays across the network is justified.

In order to create the consensus, we adopt a commonly used PD-type (proportional and derivative) decentralised control law for each agent:

$$\ddot{x}_j(t) = P\left(\sum_{k\in N_j}\frac{x_k(t-\tau_1)}{\Delta_j} - x_j(t)\right) + D\left(\sum_{k\in N_j}\frac{\dot{x}_k(t-\tau_2)}{\Delta_j} - \dot{x}_j(t)\right) \quad (1)$$

This logic tries to bring the agent's current position to the centroid of its informers, N_j, and its velocity to the mean velocity of the same set of informers (or the velocity of the same centroid), using the last known position and velocity of the informer agents. Notice that this protocol does not include self-delayed information, while all the data from the informers are delayed, positions by τ_1 and velocities by τ_2. The corresponding dynamics of the n-agent system in state space become

$$\dot{\mathbf{x}}(t) = \left(\mathbf{I}_n \otimes \begin{bmatrix} 0 & 1 \\ -P & -D \end{bmatrix}\right)\mathbf{x}(t) + \left(\mathbf{C}\otimes\begin{bmatrix} 0 & 0 \\ P & 0 \end{bmatrix}\right) \times \mathbf{x}(t-\tau_1) + \left(\mathbf{C}\otimes\begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix}\right)\mathbf{x}(t-\tau_2) \quad (2)$$

with the state vector $\mathbf{x} = \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 & \dots & x_n & \dot{x}_n \end{bmatrix} \in \Re^{2n}$ being a concatenation of the n positions and velocities. In Equation (2), $\otimes$ denotes Kronecker multiplication (Schaefer, 1996), $\mathbf{I}_n$ is the identity matrix of dimension n, $\mathbf{C} = \mathbf{\Delta}^{-1}\mathbf{A}_\Gamma$ is the *weighted adjacency matrix* created using the in-degree matrix, $\Delta = \mathrm{diag}(\Delta_1, \Delta_2, \dots, \Delta_n) \in \Re^{n\times n}$ and $\mathbf{A}_\Gamma$ is the adjacency matrix of the communication topology (Biggs, 1993). This dynamics can also be expressed in a compact form as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{x}(t-\tau_1) + \mathbf{B}_2\mathbf{x}(t-\tau_2) \quad (3)$$

with self-evident $\mathbf{A}$, $\mathbf{B}_1$ and $\mathbf{B}_2$ matrices from Equation (2).

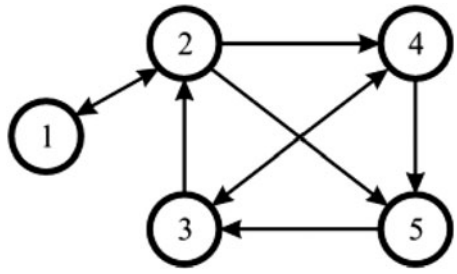

Figure 1. Example communication topology with five agents.

The complexity level of this dynamics increases rapidly as the number of agents gets larger, making the stability analysis numerically intractable. To observe this point, a very simple example is presented here. Consider five agents interacting under a very simple communication topology presented in Figure 1. The corresponding characteristic equation of the system is

$$Q(s, P, D, \tau_1, \tau_2) = \det\left(s\mathbf{I}_{2n} - \mathbf{A} - \mathbf{B}_1 e^{-\tau_1 s} - \mathbf{B}_2 e^{-\tau_2 s}\right) \quad (4)$$

Just to give the reader an idea of the enormity of the mathematics, we display the explicit form of Equation (4) for this particular case:

$$\begin{aligned} &s^{10} + 5Ds^9 + \left[5P + \left(10 - \frac{3}{4}e^{-2\tau_2 s}\right)D^2\right]s^8 \\ &+ \left[\left(20 - \frac{3}{2}e^{-(\tau_1+\tau_2)s}\right)DP + \left(10 - \frac{3}{8}e^{-3\tau_2 s} - \frac{9}{4}e^{-2\tau_2 s}\right)D^3\right]s^7 + \left[\left(10 - \frac{3}{4}e^{-2\tau_1 s}\right)P^2 \right. \\ &+ \left(30 - \frac{9}{4}e^{-2\tau_2 s} - \frac{9}{2}e^{-(\tau_1+\tau_2)s} - \frac{9}{8}e^{-(2\tau_2+\tau_1)s}\right)D^2P \\ &\left. + \left(5 - \frac{9}{4}e^{-2\tau_2 s} - \frac{3}{4}e^{-3\tau_2 s} + \frac{1}{16}e^{-4\tau_2 s}\right)\right]s^6 \\ &+ \left[\left(30 - \frac{9}{8}e^{-(\tau_2+2\tau_1)s} - \frac{9}{4}e^{-2\tau_1 s} - \frac{9}{2}e^{-(\tau_2+\tau_1)s}\right)DP^2 \right. \\ &+ \left(20 + \frac{1}{4}e^{-(3\tau_2+\tau_1)s} - \frac{3}{4}e^{-3\tau_2 s} - \frac{9}{4}e^{-(2\tau_2+\tau_1)s} \right. \\ &\left. - \frac{9}{2}e^{-(\tau_2+\tau_1)s} - \frac{9}{2}e^{-2\tau_2 s}\right)D^3P + \left(1 - \frac{3}{8}e^{-3\tau_2 s} \right. \\ &\left. + \frac{1}{16}e^{-5\tau_2 s} - \frac{3}{4}e^{-2\tau_2 s} + \frac{1}{16}e^{-4\tau_2 s}\right)D^5\Bigg]s^5 \\ &+ \left[\left(10 - \frac{9}{4}e^{-2\tau_1 s} - \frac{3}{8}e^{-3\tau_1 s}\right)P^3 \right. \\ &+ \left(30 - \frac{9}{4}\left(e^{-(\tau_2+2\tau_1)s} + e^{-2\tau_1 s} + e^{-2\tau_2 s} \right. \right. \\ &\left.\left. + e^{-(2\tau_2+\tau_1)s}\right) + \frac{3}{8}e^{-2(\tau_1+\tau_2)s} - 9e^{-(\tau_2+\tau_1)s}\right)D^2P^2 \end{aligned}$$

$$
\begin{aligned}
&+\Big(5-\frac{9}{8}e^{-(2\tau_2+\tau_1)s}-\frac{5}{16}e^{-(4\tau_2+\tau_1)s}+\frac{1}{16}e^{-4\tau_2 s}\\
&-\frac{3}{4}e^{-3\tau_2 s}-\frac{3}{2}e^{-(\tau_2+\tau_1)s}+\frac{1}{4}e^{-(3\tau_2+\tau_1)s}\\
&-\frac{9}{4}e^{-2\tau_2 s}\Big)D^4P\Big]s^4+\Big[\Big(20-\frac{3}{4}e^{-3\tau_1 s}-\frac{9}{2}e^{-(\tau_2+\tau_1)s}\\
&-\frac{9}{4}e^{-(\tau_2+2\tau_1)s}-\frac{9}{2}e^{-2\tau_1 s}+\frac{1}{4}e^{-(\tau_2+3\tau_1)s}\Big)DP^3\\
&+\Big(10+\frac{3}{8}e^{-2(\tau_2+\tau_1)s}+\frac{5}{8}e^{-(3\tau_2+2\tau_1)s}+\frac{1}{4}e^{-(3\tau_2+\tau_1)s}\\
&-\frac{3}{8}e^{-3\tau_2 s}-\frac{9}{4}e^{-2\tau_2 s}-\frac{3}{4}e^{-2\tau_1 s}-\frac{9}{8}e^{-(\tau_2+2\tau_1)s}\\
&-\frac{9}{4}e^{-(2\tau_2+\tau_1)s}-\frac{9}{2}e^{-(\tau_2+\tau_1)s}\Big)D^3P^2\Big]s^3\\
&+\Big[\Big(5-\frac{9}{4}e^{-2\tau_1 s}-\frac{3}{4}e^{-3\tau_1 s}+\frac{1}{16}e^{-4\tau_1 s}\Big)P^4\\
&+\Big(10+\frac{1}{4}e^{-(\tau_2+3\tau_1)s}-\frac{9}{4}\left(e^{-2\tau_1 s}+e^{-(\tau_2+2\tau_1)s}\right)\\
&-\frac{3}{8}\left(e^{-3\tau_1 s}-e^{-2(\tau_2+\tau_1)s}\right)-\frac{9}{2}e^{-(\tau_2+\tau_1)s}-\frac{3}{4}e^{-2\tau_2 s}\\
&+\frac{5}{8}e^{-(2\tau_2+3\tau_1)s}-\frac{9}{8}e^{-(2\tau_2+\tau_1)s}\Big)D^2P^3\Big]s^2\\
&+\Big[5-\frac{3}{2}e^{-(\tau_2+\tau_1)s}-\frac{3}{4}e^{-3\tau_1 s}+\frac{5}{16}e^{-(\tau_2+4\tau_1)s}\\
&-\frac{9}{8}e^{-(\tau_2+2\tau_1)s}+\frac{1}{4}e^{-(\tau_2+3\tau_1)s}-\frac{9}{4}e^{-2\tau_1 s}\\
&+\frac{1}{16}e^{-4\tau_1 s}\Big]DP^4s+\Big(1-\frac{3}{4}e^{-2\tau_1 s}+\frac{1}{16}e^{-4\tau_1 s}\\
&-\frac{3}{8}e^{-3\tau_1 s}+\frac{1}{16}e^{-5\tau_1 s}\Big)P^5 \qquad (5)
\end{aligned}
$$

This equation is a 10th-order quasi-polynomial with commensuracy degree 5 (i.e., up to $5\tau_1$ and $5\tau_2$ terms appear) and, more critically, the crosstalk between the time delays (i.e., terms like $\tau_1+\tau_2$ and $3\tau_1+\tau_2$) is present. It is clear that the complexity of Equation (5) increases rapidly as the number of agents gets larger or if a more elaborate communication topology is used. The stability problem of this general class of multiple time-delay systems (MTDSs) is also notoriously known to be NP-hard (Toker & Ozbay, 1996). It becomes numerically intractable very quickly as the order of the characteristic equation increases. Even the strongest of the present mathematical tools to analyse the stability of time-delay systems falls short to handle such complexities. To remedy this impasse, two procedures are combined in this paper. First, a factorisation operation is performed to break the characteristic equation into quasi-polynomial factors of reduced order and simpler but identical forms, as described in this section. Second, a method called the CTCR (Fazelinia et al., 2007; Ergenc et al., 2007) is deployed to analyse the stability of each factor, as explained in Section 3. This combination of factorisation and CTCR facilitates an efficient and novel mechanism to resolve the problem.

Lemma 1 (Factorisation property)**:** *The characteristic equation of system* (2) *can always be expressed as the product of a set of second- and fourth-order factors:*

$$
\begin{aligned}
&Q(s,P,D,\tau_1,\tau_2)\\
&\quad=\det\left(s\mathbf{I}_{2n}-\mathbf{A}-\mathbf{B}_1e^{-\tau_1 s}-\mathbf{B}_2e^{-\tau_2 s}\right)\\
&\quad=\prod_{j=1}^{\ell+m}q_j\left(s,P,D,\tau_1,\tau_2,\lambda_j\right)\\
&\quad=\prod_{j=1}^{\ell}\left[s^2+Ds+P-\lambda_j\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)\right]\\
&\qquad\times\prod_{j=\ell+1}^{m}\left[s^4+2Ds^3+\left(D^2+2P\right)s^2+2DPs+P^2\right.\\
&\qquad-2\mathrm{Re}\left(\lambda_j\right)\left(s^2+Ds+P\right)\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)\\
&\qquad\left.+\left|\lambda_j\right|^2\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)^2\right]=0 \qquad (6)
\end{aligned}
$$

where λ_j represents the eigenvalues of $\mathbf{C}=\mathbf{\Delta}^{-1}\mathbf{A}_\Gamma$. This matrix has ℓ real eigenvalues, denoted by $j=1,2,3,\ldots,\ell$, and m complex conjugate eigenvalue pairs $\left(\lambda_j,\lambda_j^*\right)$, $j=\ell+1,\ \ell+2,\ \ell+3,\ldots,\ell+m$. Then, $n=2m+\ell$ (see Appendix for the proof).

Remark 1: In Lemma 1, we have assumed that the real eigenvalues of the matrix $\mathbf{C}$ always create Jordan blocks of size 1. If a multiple real eigenvalue creates a Jordan block of size 2 or larger, the corresponding characteristic equation factor can be obtained and analysed as we demonstrate below. Jordan blocks of size larger than two are extremely rare. Although the procedure is discussed, we will ignore further pursuit along this line.

Remark 2: The factorisation idea presented here is similar to the one used before for consensus problems with undirected communication topologies (Cepeda-Gomez & Olgac, 2011a, 2011b). However, there is an extra difficulty added in the form of the second-order factors created by Jordan blocks of size 2.

Remark 3: The above factorisation feature has been recognised by many researchers earlier (Fax & Murray, 2004) and presented in different ways. However, we include the above segment for a more lucid proof of Lemma 1 and to better show the interplay between the spectra (the eigenvalues) of a special matrix $\mathbf{C}$ and the final form of the factors. This is critical for the deployment of CTCR and allows a more generic treatment of different communication structures with very large membership counts (more than 10 agents).

Lemma 1 simplifies the problem considerably, by transforming it from a $2n$-order system with time delays of commensuracy degree n and delay crosstalk (just like in the

example given by Equation (5)) into a combination of ℓ second-order and m fourth-order subsystems with maximum commensuracy degree of 2 (e.g., $e^{-2\tau_1 s}$) and a simple delay crosstalk term (e.g., $e^{-(\tau_1+\tau_2)s}$). Furthermore, since the *only* discriminating element from one factor to the other is the eigenvalue λ_j, the complete stability analysis can be handled very efficiently. For this, we first perform the stability analysis only once for a generic real eigenvalue and a generic complex one. Then, we simply repeat these results for $\ell + m$ different eigenvalues (i.e., subsystems). The superposition of the stability pictures for each factor eventually yields the global stability outlook for the complete system. This property makes the stability analysis virtually independent of the number of agents. The complexity in determining the stability of a quasi-polynomial such as Equation (5) is now considerably reduced to the determination of the eigenvalues of a known matrix **C** and repeated creation of the stability pictures of simple quasi-polynomials (which we call the 'factors' from this point forward), which are in the form given by Equation (6). The process to obtain the individual stability bounds can be automated, without the need of user intervention beyond the description of the communication structure.

In order to demonstrate the power provided by Lemma 1, we consider again the five-agent topology of Figure 1. The **C** matrix corresponding to this topology has the eigenvalue set $1, 0.38, -0.5, -0.44 \pm 0.37i$. The ensuing factorised characteristic equation, which displays the footprints of these eigenvalues, is

$$\begin{aligned}&\left[s^2 + Ds + P - \left(Dse^{-\tau_2 s} + Pe^{-\tau_1 s}\right)\right]\left[s^2 + Ds + P\right.\\&\left. - 0.38\left(Dse^{-\tau_2 s} - Pe^{-\tau_1 s}\right)\right] \times \left[s^2 + Ds + P\right.\\&\left. + 0.5\left(Dse^{-\tau_2 s} - Pe^{-\tau_1 s}\right)\right]\left[\left(s^4 + 2Ds^3\right)\right.\\&\left. + \left(D^2 + 2P\right)s + 2DPs + P^2 - 0.88\left(s^2 + Ds + P\right)\right.\\&\left. \times \left(Dse^{-\tau_2 s} - Pe^{-\tau_1 s}\right) + 0.33\left(Dse^{-\tau_2 s} - Pe^{-\tau_1 s}\right)^2\right]\\&= 0 \qquad (7)\end{aligned}$$

Obviously, the first three factors correspond to the three real eigenvalues and the last one emanates from the complex eigenvalue pair. The conversion of the characteristic equation from Equation (5) to Equation (7) represents a wonderful simplification.

Some key observations

We now wish to direct the discussions to some other features of the eigenvalues of the weighted adjacency matrix **C**. From the way in which this matrix is created, its diagonal consists of zeroes only (so-called *hollow matrix*), and any row elements always add up to 1. The latter property makes **C** a *row-stochastic matrix* of which the components are all non-negative (Marcus & Minc, 1996). Row-stochastic non-negative matrices possess a wonderful feature: the eigenvalues of such matrices are always within the unit disc of the complex plane. This feature arises as an extension to the Gershgorin's disc theorem (Bell, 1965). Furthermore, it can be proven (Agaev & Chebotarev, 2005) that if the topology is connected and has at least one spanning tree, $\lambda = 1$ is one of the eigenvalues of the weighted adjacency matrix **C** with multiplicity 1. Then, there is always a factor

$$\begin{aligned}q_1(s, P, D, \tau_1, \tau_2, \lambda = 1) &= s^2 + Ds\\&+ P - \left(Dse^{-\tau_2 s} + Pe^{-\tau_1 s}\right) = 0 \qquad (8)\end{aligned}$$

in the characteristic quasi-polynomial (5) which corresponds to $\lambda = 1$. Without loss of generality, we will assign this eigenvalue to the state ξ_1. It can be shown that the normalised eigenvector corresponding to ξ_1 is always $\mathbf{t}_1 = 1/\sqrt{n}\,[1 \;\; 1 \;\; 1 \;\; \ldots \;\; 1]^T \in \Re^n$, and it is selected as the first column of the transformation matrix **T** that converts the matrix **C** into its Jordan form: $\mathbf{\Lambda} = \mathbf{T}^{-1}\mathbf{C}\mathbf{T}$. This factor governs the dynamics of ξ_1, which can be shown to be a weighted average of the positions of the agents. The weights for the computation of ξ_1 arise from the first row of the inverse of the transformation matrix **T**. We call this quantity ξ_1, the *weighted centroid*, which is, obviously, topology dependent. It represents some sort of an *agreement dynamics* in the swarm, because if there is an agreement (consensus) we expect the behaviour of the weighted centroid to represent that consensus.

The other factors of the characteristic equation (6) must then be related to some disagreement dynamics of the system. When these disagreement dynamics are stable (i.e., their states asymptotically vanish), the agents will reach consensus among themselves.

It can be seen that $s = 0$ is always a stationary root of Equation (8) independent of the delays, τ_1 and τ_2, which implies that the dynamics of the weighted centroid are, at best, marginally stable. In fact, provided that the disagreement dynamics are stable, ξ_1 reaches a non-zero steady state, when the topology is connected and has at least one spanning tree (also known as arborescence) (Gabow & Myers, 1978). Regardless of the number of arborescences, each agent remains connected to the root of its arborescence.

In Lemma 1, we implicitly require every agent to have at least one informer, otherwise the matrix $\mathbf{\Delta}$ would be singular. Now, when the communication topology does not only have a spanning tree but *it is* a tree, the root node does not have any informer. This situation is useful to model leader–follower consensus cases (Meng et al., 2011), where that root node dictates the dynamics upon which the agents agree, regardless of the initial conditions. For leaderless consensus, everybody should listen to at least one peer, and the agreement takes place at a point related to the initial conditions of the group. Then, we need every agent to have at least one informer to reach consensus at the value dictated by the weighted centroid dynamics.

If the communication topology does not have a spanning tree, however, $\lambda = 1$ is a multiple eigenvalue of **C** and Equation (8) appears as a repeated factor within the characteristic equation (5). These factors represent the dynamics of the centroids of the subgroups generated by the subgraphs which contain different spanning trees. If all the disagreement factors are stable, the swarm members within a subgroup reach some stationary positions which are generally different. Thus, the consensus is not achieved. These facts are stated in the following lemmas, the proofs of which are in Appendix.

Lemma 2 (Group behaviour)**:** *Assume that the communication topology has at least one spanning tree and every agent has at least one informer. Then, the agents in the group reach a consensus if and only if the factor* (8) *is marginally stable and all the remaining factors of Equation* (6) *are stable. Furthermore, the consensus value will be* $\lim_{t\to\infty} x_j(t) = \lim_{t\to\infty} \sqrt{n}\xi_1(t = \infty)$*, whereas the other states tend to* $\xi_j(t = \infty) = 0$ *for* $j = 2,\ 3, \ldots, \ell + m$.

Lemma 3 (Topologies without spanning trees)**:** *If the given communication topology does not have a spanning tree, the control logic described by Equation* (1) *cannot result in consensus.*

Remark 4: The results presented in Lemmas 2 and 3 concur with the consensus literature. However, they offer two important distinctions. The treatment here considers second-order agents without self-delayed feedback construct, thus proper preparatory discussions are provided for the integrity of the document. Second, this paper brings a different but more lucid proof of the unsuccessful consensus in the presence of multiple spanning trees.

According to Lemma 2, the characteristic polynomial given by q_1 in Equation (8) is related to the motion of the topology-dependent centroid. That is, if the swarm behaviour creates a consensus, the agents will agree in the dynamics governed by this characteristic. The remaining factors which have the generic form of Equation (6) are related to the stability of the relative motion of the agents with respect to the expected consensus. Therefore, they represent the *disagreement dynamics*. A stable swarm consensus is reached if all the $q_j\left(s, P, D, \tau_1, \tau_2, \lambda_j\right)$, with $j = 2,\ 3,\ \ldots,\ \ell + m$, and the weighted centroid behaviour $q_1(s, P, D, \tau_1, \tau_2, \lambda_1 = 1)$ are all stable. Alternatively, if all the factors of Equation (6) are stable except q_1 in Equation (8), i.e., the weighted centroid of the group has an unstable behaviour, the agents will move in coherence but along an unstable trajectory.

3. Main contribution: stability analysis using CTCR paradigm and the SDS

Lemma 1 points that the stability of the swarm dynamics is only achieved provided that each of the factors given in Equation (6) represents stable system behaviour. For a given set of control parameters (P, D), these factors have the general formations of

$$q(s, \tau_1, \tau_2) = g_{11}(s) + g_{12}(s)\, e^{-\tau_1 s} + g_{13}(s)\, e^{-\tau_2 s} = 0 \tag{9}$$

for Equation (A7a), and

$$\begin{aligned} q(s, \tau_1, \tau_2) = {} & g_{21}(s) + g_{22}(s)\, e^{-\tau_1 s} + g_{23}(s)\, e^{-\tau_2 s} \\ & + g_{24}(s)\, e^{-(\tau_1+\tau_2)s} + g_{25}(s)\, e^{-2\tau_1 s} \\ & + g_{26}(s)\, e^{-2\tau_2 s} = 0 \end{aligned} \tag{10}$$

for Equation (A7b), where $g_{ij}(s)$ are some polynomials of s. Furthermore, the class of quasi-polynomials given in Equation (9) is a subset of those of Equation (10). Therefore, we will concentrate on the stability treatment of the latter generic class. Notice that this quasi-polynomial represents a system with multiple and rationally independent time delays with delay crosstalk and commensurate degree 2.

The stability analysis of the factors in Equation (10) is not a trivial task, as these quasi-polynomials possess infinitely many characteristic roots, and all of them should be in the stable left-half of the complex plane. This feature is further exacerbated due to the presence of two rationally independent delays. The complex task of assessing the stability is performed deploying a unique methodology called the CTCR (Ergenc et al., 2007; Fazelinia et al., 2007). The main philosophy behind it is the detection of the right-half plane characteristic roots (i.e., unstable roots), of a linear time invariant–multiple time-delay system (LTI-MTDS), such as Equation (10), similar to what Routh's array achieves for LTI systems with no delay. It is well known in the linear system literature that the number of unstable roots of a characteristic equation such as Equation (10) can change only along certain loci in the domain of the delays. The CTCR method requires the exhaustive knowledge of these so-called 'stability switching curves'. In this paper, we utilise a novel approach to obtain this knowledge in a surrogate domain called the SDS. The following paragraphs present some preparatory definitions and explain the key propositions of CTCR, leaving the details to the references (Ergenc et al., 2007; Fazelinia et al., 2007).

Definition 1 (Kernel hypercurves $\wp_0$)**:** The hypercurves that consist of *all* the points $(\tau_1,\ \tau_2) \in \Re^{2+}$exhaustively, which cause an imaginary root $s = \omega i,\ \ \omega \in \Re^+$ and satisfy the constraint $0 < \tau_j\omega < 2\pi$ are called the *kernel hypercurves*. The points on this hypercurve contain the smallest possible delay values that create the given imaginary root at the frequency ω.

Definition 2 (Offspring hypercurves $\wp$)**:** The hypercurves obtained from the kernel hypercurve by the following

pointwise nonlinear transformation:

$$\left\langle \tau_1 \pm \frac{2\pi}{\omega} j_1, \tau_2 \pm \frac{2\pi}{\omega} j_2 \right\rangle, \quad j_1, j_2 = 0, 1, 2, \ldots \tag{11}$$

are called the offspring hypercurves.

Definition 3 (Root tendency (RT))**:** The RT indicates the direction of transition of the imaginary root (to the right or to the left-half of the complex plane) as only one of the delays increases by ε, $0 < \varepsilon \ll 1$, while the other delay remains constant:

$$\mathrm{RT}|_{s=\omega i}^{\tau_j} = \mathrm{sgn}\left[\mathrm{Re}\left(\frac{\partial s}{\partial \tau_j}\bigg|_{s=\omega i}\right)\right] \tag{12}$$

Root tendencies are -1 for stabilising and $+1$ for destabilising root crossings across the imaginary axis.

There are two key propositions of the CTCR paradigm without which an exhaustive parametric stability assessment can be possible.

Proposition 1 (Small number of kernel hypercurves)**:** *The number of kernel hypercurves is manageably small: for an LTI-TDS of state dimension* n*, that number is bounded by* n^2 (*Ergenc et al.,* 2007).

Proposition 2 (Invariant RT property)**:** *Take an imaginary characteristic root,* ωi*, caused by any one of the infinitely many grid points on the kernel and offspring hypercurves in* $(\tau_1, \tau_2) \in \Re^{2+}$ *domain defined by expression* (11)*. The RT of these imaginary roots remains invariant so long as the grid points on different 'offspring hypercurves' are obtained by keeping one of the delays fixed. That is, the RT with respect to the variations of* τ_{j} *is invariant from the kernel to the corresponding offspring as the other delay* τ_k, $k \neq j$ *is fixed.*

Spectral delay space

We describe a new procedure for determining the kernel (and offspring) hypercurves. It is a formalised treatment from a recent thesis work (Fazelinia, 2007) and it is developed on a new domain called the SDS. SDS is defined by the coordinates $v_j = \tau_j \omega$ for every point $(\tau_1, \tau_2) \in \Re^{2+}$on the kernel and the offspring hypercurves. This is a conditional mapping: if a delay set $(\tau_1, \tau_2) \in \Re^{2+}$creates an imaginary root ωi, (i.e., if the point is on the kernel or the offspring hypercurves), then $(\tau_1\omega, \tau_2\omega)$ forms a point in the SDS. On the contrary, (τ_1, τ_2) points that do not generate an imaginary root have no representation in the SDS domain.

The main advantage of SDS is that the image of the kernel hypercurve in the SDS, denoted as $\wp_0^{\mathrm{SDS}}$ and named the *building hypercurve*, is confined to a square of edge length 2π (see Definition 1). Then, it is only necessary to explore a finite domain in SDS in order to find the representation of the building hypercurves in SDS. This finite domain is known as the *building block* (BB), i.e., a square of $2\pi \times 2\pi$. Similarly, we name the corresponding representation of offspring hypercurves in the SDS as the *reflection curves*. An important advantage of SDS domain is that the transition from the *building hypercurves* to the *reflection hypercurves* is achieved simply by translating the building hypercurve by 2π, as opposed to using the pointwise nonlinear transformation (11). SDS also prevents the undesirable shape distortion from kernel to offspring which occurs in the delay space. The reflection curves in the SDS, $\wp_{\mathrm{SDS}}$, are generated by simply stacking the BB squares with $\wp_0^{\mathrm{SDS}}$ in it one next to the other. We refer to this property as the *stackability feature* of the SDS.

With these definitions and propositions, we now return to the mentioned preparatory stage of CTCR method. It is the exhaustive determination of all the imaginary roots, $s = \omega i$, for the generic factor $q\,(s, \tau_1, \tau_2)$ of the characteristic equation, as in Equation (10) within the semi-infinite quadrant of $(\tau_1, \tau_2) \in \Re^{2+}$. For this, we follow the procedure given below. For the sake of simplicity, we present the steps for a generic second-order factor (9). The same procedure, slightly expanded, applies to (10). First, the exponential terms in Equation (9) are replaced by

$$e^{-\tau_k \omega\, i} = \cos(v_k) - i\ \sin(v_k), \quad v_k = \tau_k \omega \tag{13}$$

And the sine and cosine functions are expressed in terms of a half-angle tangent function:

$$\cos(v_k) = \frac{1 - z_k^2}{1 + z_k^2}, \quad \sin(v_k) = \frac{2z_k}{1 + z_k^2}, \quad z_k = \tan\left(\frac{v_k}{2}\right) \tag{14}$$

Now $q\,(s = \omega\, i, \tau_1, \tau_2)$ can be rewritten as a polynomial of ω with complex coefficients, which are parameterised in z_1 and z_2:

$$q_j(\omega, z_1, z_2) = \sum_{k=0}^{2} c_k\left(P, D, \lambda_j, z_1, z_2\right)(\omega\, i)^k = 0 \tag{15}$$

In this equation, we abuse the notation as the arguments of $q\,(s, \tau_1, \tau_2)$ are changed but the function's name q is kept the same. Please note that the subscript j establishes the connection between a specific eigenvalue and the corresponding factor. If there is an imaginary solution at $s = \omega\, i$, $\omega \in \Re^+$ of Equation (10), both the real and the imaginary parts of Equation (15) must be zero simultaneously:

$$\mathrm{Re}\left[q_j(\omega, z_1, z_2)\right] = \sum_{k=0}^{2} f_k(z_1, z_2)\,\omega^k = 0 \tag{16a}$$

$$\mathrm{Im}\left[q_j(\omega, z_1, z_2)\right] = \sum_{k=0}^{2} g_k(z_1, z_2)\,\omega^k = 0 \tag{16b}$$

The necessary condition for Equations (16a) and (16b) to have a common root, ω, is simply stated through a Sylvester's resultant matrix:

$$\mathbf{M} = \begin{bmatrix} f_2(z_1, z_2) & f_1(z_1, z_2) & f_0(z_1, z_2) & 0 \\ 0 & f_2(z_1, z_2) & f_1(z_1, z_2) & f_0(z_1, z_2) \\ g_2(z_1, z_2) & g_1(z_1, z_2) & g_0(z_1, z_2) & 0 \\ 0 & g_2(z_1, z_2) & g_1(z_1, z_2) & g_0(z_1, z_2) \end{bmatrix} \tag{17}$$

In order for Equation (16a, b) to be satisfied, $\mathbf{M}$ should be singular. This results in the following expression in terms of z_1 and z_2:

$$\det(\mathbf{M}) = F(z_1, z_2) = F\left(\tan\left(\frac{v_1}{2}\right), \tan\left(\frac{v_2}{2}\right)\right) \tag{18}$$

which constitutes a closed-form description of the *building hypercurves* in the SDS (v_1, v_2). To obtain its graphical depiction, one of the parameters, say v_2, can be scanned in the range of $[0, 2\pi]$ and the corresponding v_1 values are calculated again in $[0, 2\pi]$. Notice that every point (v_1, v_2) on these curves brings an imaginary characteristic root at $\pm\omega i$. That is, we have a continuous sequence of (v_1, v_2, ω) sets all along the building hypercurves. During this process, several key properties of the BB concept are also deployed: (1) the confinement of the BB within $0 < \tau_j\omega < 2\pi$ squares, and (2) the stackability of the BBs.

In order to assess the stability properties of the system in the space of the delays, we now back transform the building hypercurves from the (v_1, v_2) domain of SDS to the (τ_1, τ_2) delay space, using the inverse transformation of Equation (21). For every point in SDS, the knowledge of (v_1, v_2, ω) creates the corresponding (τ_1, τ_2, ω), and thus a point on the kernel or the offspring hypercurves. The RT invariance property follows, for the creation of the complete and exact stability outlook of the system, which is unique. Examples of this construction are presented in Section 5.

4. From consensus to formation control

The consensus protocol which is studied in the previous sections is deployed to drive the agents towards the agreed-upon consensus position. This behaviour, however, does not make much practical sense as the agents would ultimately collide. It is well known that consensus structure can be transformed into a formation. This transition has been investigated before, as recent survey article (Cao, Yu, Wei, & Cheng, 2013) summarises. Various approaches which were suggested to generate a group formation departing from consensus protocols are heavily nonlinear. We wish to list only a few of them here for comparison. Pavone and Frazzolli (2007) analyses kinematic agents and then extends the results to non-holonomic agents (unicycles). However, their results are limited to circular formations, in which each agent follows one of its neighbours. Sepulchre, Paley, and Leonard (2007, 2008) also study collective motion of linear, kinematic agents, however, the proposed control logic is highly nonlinear. Dimarogonas and Kyriakopoulos (2009) present a dispersion algorithm to guarantee spacing among members of a swarm, again bringing severe nonlinear effects into the group dynamics.

The formation generation algorithm presented in this paper is novel from two aspects: it is a very simple construct and it preserves the linearity of the dynamics. Furthermore, we provide a clear procedure to design any formation configuration.

We propose a modification to the control law (1), preserving the linearity of the dynamics in place, in such a way that the agents can maintain some distances among themselves. To create this 'formation', we take advantage of the decoupling property presented in Lemma 1. If a constant forcing term is introduced into the disagreement dynamics in the modal coordinates ξ_j, some of the inter-agent distances will be non-zero at the steady state. The question for synthesising the control is how to determine these forces in order to achieve a desired swarm formation. The relation between the forcing terms in the transformed domain$\xi_j$and the final swarm formation is obviously topology dependent. The rest of this section addresses this particular question.

We introduce an additional constant control term $\boldsymbol{\varphi}_j$, on each one of the disagreement subsystems ξ_j, $j = 2, 3, \ldots, \ell + m$. Notice that the consensus forming state of the centroid ξ_1 is not forced, as its final value depends only on the initial conditions. The dynamics of the disagreement subsystems are given by

$$\dot{\boldsymbol{\xi}}_j(t) = \mathbf{A}_j\boldsymbol{\xi}_j(t) + \mathbf{B}_{1j}\boldsymbol{\xi}_j(t-\tau_1) + \mathbf{B}_{2j}\boldsymbol{\xi}_j(t-\tau_2) + \boldsymbol{\varphi}_j, \quad j = 2, 3, \ldots, \ell + m \tag{19}$$

The system matrices $\mathbf{A}_j$, $\mathbf{B}_{1j}$ and $\mathbf{B}_{2j}$ are self-evident from Equations (A6a) and (A6b), for second- and fourth-order factors. The constant forcing vector has the form $\boldsymbol{\varphi}_j = \begin{bmatrix} 0 & \varphi_j \end{bmatrix}^T$ for second-order factors and $\boldsymbol{\varphi}_j = \begin{bmatrix} 0 & \varphi_{1j} & 0 & \varphi_{2j} \end{bmatrix}^T$ for fourth-order factors.

To study the influence of the forcing terms on the steady-state value of the disagreement dynamics, we take the Laplace transform of Equation (19):

$$\boldsymbol{\Xi}_j(s) = \left(s\mathbf{I} - \mathbf{A}_j - \mathbf{B}_{1j}e^{-\tau_1 s} - \mathbf{B}_{2j}e^{-\tau_2 s}\right)^{-1}\frac{\boldsymbol{\varphi}_j}{s} \tag{20}$$

where $\boldsymbol{\Xi}(s)$ represents the Laplace transformation of $\xi(t)$. Notice that we are considering zero initial conditions for the disagreement dynamics in Equation (19). This is based on the fact that the disagreement factors are assumed stable, and therefore, the effect of non-zero initial conditions should vanish after the transient behaviour. If we apply the

final value theorem to (20), it yields:

$$\begin{aligned}\boldsymbol{\xi}_{j,ss} &= \boldsymbol{\xi}_j\left(t=\infty\right) = \lim_{s\to 0} s\,\boldsymbol{\Xi}_j\left(s\right)\\ &= -\left(\mathbf{A}_j + \mathbf{B}_{1j} + \mathbf{B}_{2j}\right)^{-1}\boldsymbol{\varphi}_j\end{aligned} \tag{21}$$

where the subscript *ss* refers to the steady-state values. For the second-order factors, Equation (21) has the explicit form:

$$\boldsymbol{\xi}_{j,ss} = -\frac{\varphi_j}{P\left(\lambda_j - 1\right)}\begin{bmatrix}1\\0\end{bmatrix} \tag{22a}$$

whereas the fourth-order factors impart

$$\begin{aligned}\boldsymbol{\xi}_{j,ss} = &-\frac{1}{P\left(\left|\lambda_j\right|^2 + 1 - 2\mathrm{Re}\left(\lambda_j\right)\right)}\\ &\times\begin{bmatrix}\mathrm{Re}\left(\lambda_j\right)-1 & -\mathrm{Im}\left(\lambda_j\right)\\ 0 & 0\\ \mathrm{Im}\left(\lambda_j\right) & \mathrm{Re}\left(\lambda_j\right)-1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\varphi_{1j}\\ \varphi_{2j}\end{bmatrix}\end{aligned} \tag{22b}$$

Notice that Equations (22a) and (22b) imply constant relative positions among the agents and therefore zero relative velocity.

Using the back transformation, one can obtain the corresponding agent positions in the formation as $\mathbf{x}_{ss} = (\mathbf{T}\otimes\mathbf{I}_2)\,\boldsymbol{\xi}_{ss}$. Obviously, the formation design process does not start from the forcing terms $\boldsymbol{\varphi}_j$, but from the desired final configuration of the agents. Then, the actual algorithm takes the set of desired final configurations for the agents, $\mathbf{x}_{ss}$, and generates the required final values of the disagreement terms, $\boldsymbol{\xi}_{ss} = \left(\mathbf{T}^{-1}\otimes\mathbf{I}_2\right)\mathbf{x}_{ss}$. Once these values are known, the inverse operations of Equation (22) are applied to obtain the forcing terms $\boldsymbol{\varphi}_j$.

It is important to notice that this formation algorithm guarantees the desired *relative* distances that are achieved, as given in $\mathbf{x}_{ss}$. However, it does not assure the final *absolute* positions of the agents in formation. This is because the centroid-related term is not controlled, thus the agents create the formation around the centroid, motion of which is defined by the initial configuration of the agents. This nuance, i.e., the application of a constant forcing term to the centroid dynamics, introduces an extra degree of flexibility to the formation control algorithm. Since this factor exhibits marginal stability, as explained in Section 2, applying a constant forcing term simply results in a steady-state behaviour with constant velocity and linearly increasing position. This velocity is reflected in the motion of every single agent. Then, an extended coordinated motion can be achieved if ξ_1 is also controlled.

Finally, one must consider that the control $\boldsymbol{\Phi} = \begin{bmatrix}0 & 0 & \varphi_2^T & \cdots & \varphi_{\ell+m}^T\end{bmatrix}^T \in \Re^{2n}$ cannot be applied in the transformed coordinate $\boldsymbol{\xi}$. They have to be deployed in the actual displacement domain $\mathbf{x}$. Therefore, we use the transformation $\mathbf{F} = (\mathbf{T}\otimes\mathbf{I}_2)\,\boldsymbol{\Phi}$, where the vector $\mathbf{F} = \begin{bmatrix}0 & f_1 & 0 & f_2 & \ldots & 0 & f_n\end{bmatrix}^T \in \Re^{2n}$ contains the control terms applied to each agent in the actual displacement domain x. With the inclusion of the forcing term $\mathbf{F}$, the dynamics of the multi-agent system become

$$\begin{aligned}\dot{\mathbf{x}}\left(t\right) = &\left(\mathbf{I}_n\otimes\begin{bmatrix}0 & 1\\ -P & -D\end{bmatrix}\right)\mathbf{x}\left(t\right) + \left(\mathbf{C}\otimes\begin{bmatrix}0 & 0\\ P & 0\end{bmatrix}\right)\\ &\times\mathbf{x}\left(t-\tau_1\right) + \left(\mathbf{C}\otimes\begin{bmatrix}0 & 0\\ 0 & D\end{bmatrix}\right)\mathbf{x}\left(t-\tau_2\right) + \mathbf{F}\end{aligned} \tag{23}$$

Notice that this mechanism is very different from the formation control approach followed by Lin and Jia (2009b), where the authors include inter-agent distances in the control protocol. This inclusion introduces severe nonlinearities in the overall dynamics, whereas the present approach simply exerts a constant forcing term as a supplement over the consensus creating protocol. Therefore, the treatment does not affect the linearity of the dynamics. This is completely new and distinct from the common pathways adopted to date.

Examples of this design process are presented in the following section.

5. Illustrative examples

In this section, we present the simulation results that verify the theoretical analysis of previous sections. All of the examples use the communication topology of Figure 1. The weighted adjacency matrix, once again, has the eigenvalues: 1, 0.38, −0.5, $-0.44 \pm 0.37i$. The control gains used in the examples are $P = 2$ and $D = 0.8$.

5.1. *Stability analysis*

The deployment of the CTCR paradigm using the BB concept leads to Figures 2 and 3. In Figure 2, the SDS representation of the characteristic equation factors with their respective building and reflection hypercurves is shown. They correspond to the real and the complex eigenvalues. The properties of the SDS are clear: the reflection curves, presented in blue, are obtained by translating the building curves, which are depicted in red. Notice that the building curves are confined to a square of side length 2π, i.e., the building block.

Figure 3 shows the representation of the stability switching curves in the space of the time delays for each one of the factors. The kernel curves are represented in red, whereas the offspring are in blue. The shaded region represents the stability zone in the domain of the delays for each factor. The intersection of these individual regions results in the combined stability chart of Figure 4. Delay combinations inside the shaded portion of this figure result in a stable behaviour, i.e., agents reach consensus.

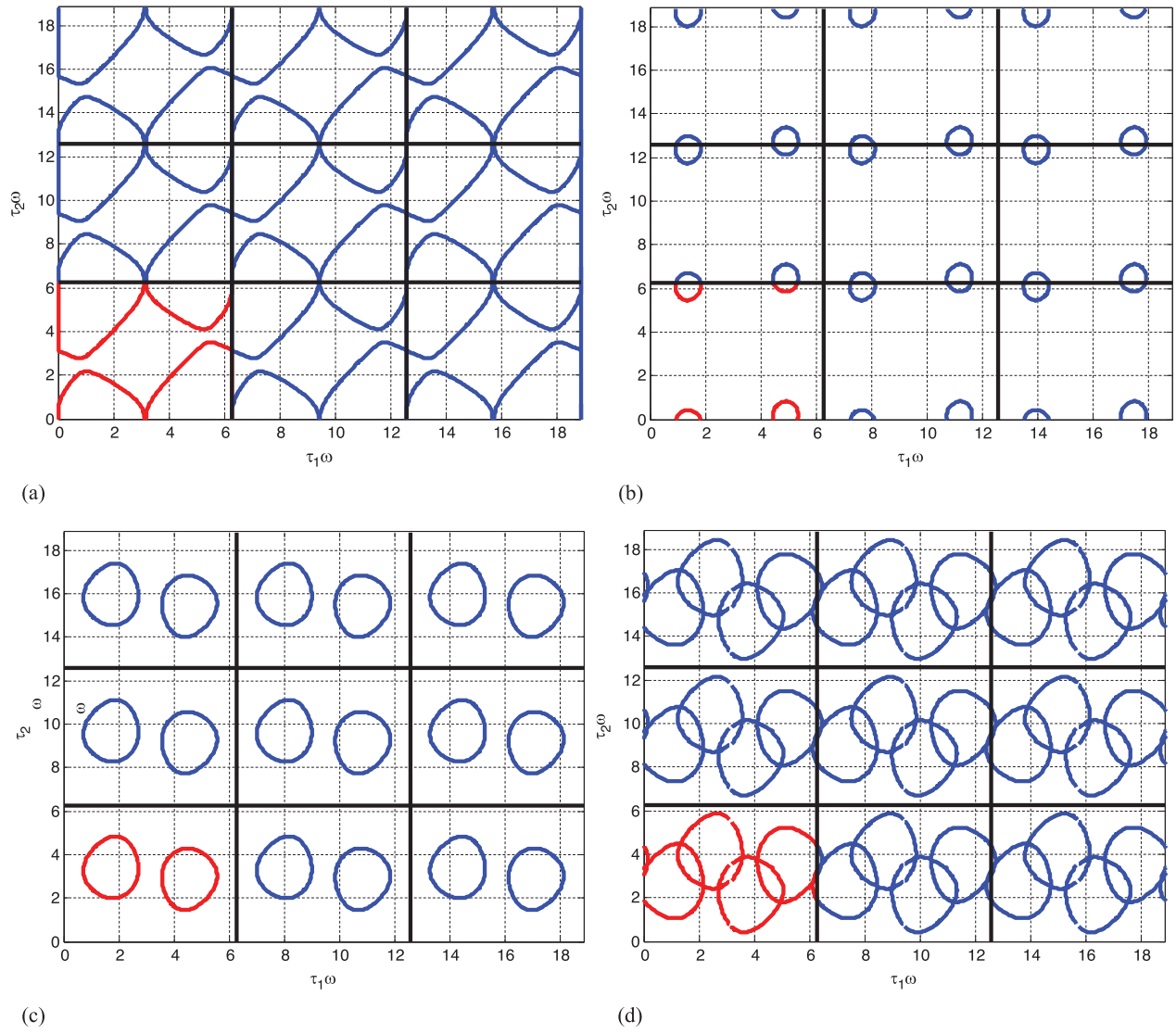

Figure 2. Spectral delay space representation of the stability switching curves generated by the communication topology of Figure 1 with $P = 2$ and $D = 0.8$. They correspond to (a) $\lambda = 1$, (b) $\lambda = 0.38$, (c) $\lambda = -0.5$ and (d) $\lambda = -0.44 \pm 0.37i$.

It is important to highlight that Figure 4 is an exact and exhaustive stability map. The shaded regions represent delay combinations that yield no characteristic root on the right-half of the complex plane. At any point on the kernel or offspring curves, there is a pair of purely imaginary roots which make the system marginally stable. There is no region of stability outside those marked in Figure 4; thus the method is exhaustive. There is no other methodology in the literature that allows the creation of such exact and exhaustive stability map in the domain of the delays.

These results are verified in Figure 5, in which panel (a) shows the simulation results for a delay combination of $(\tau_1, \tau_2) = (0.5,\ 0.5)$ seconds, corresponding to point **a** in Figure 4, panel (b) is for $(\tau_1, \tau_2) = (1,\ 2.5)$ (point **b** in Figure 4) and panel (c) is for $(\tau_1, \tau_2) = (1.3,\ 4.5)$, represented by point **c** in Figure 4. It is clear that delay combinations corresponding to points inside the shaded region (**a** and **c**) generate stable consensus, whereas points outside create divergent behaviour.

A more interesting and counter-intuitive conclusion we can make from Figures 4 and 5 is that larger time delays not always imply worse performance of the system. Both time delays are larger in **c** than in **b**, however, the system is stable in the first case and unstable in the second. This feature has been used in a control synthesis scheme named the *delay scheduling* (Olgac & Cavdaroglu, 2011). It offers wonderful options for the most effective control parameter selections, because the control designer can manipulate the feedback delay as a part of the control selection. Due to the causality principles, the delay can only be increased (i.e., one cannot sense the events before they happen). Anyway, the control designer can always intentionally prolong

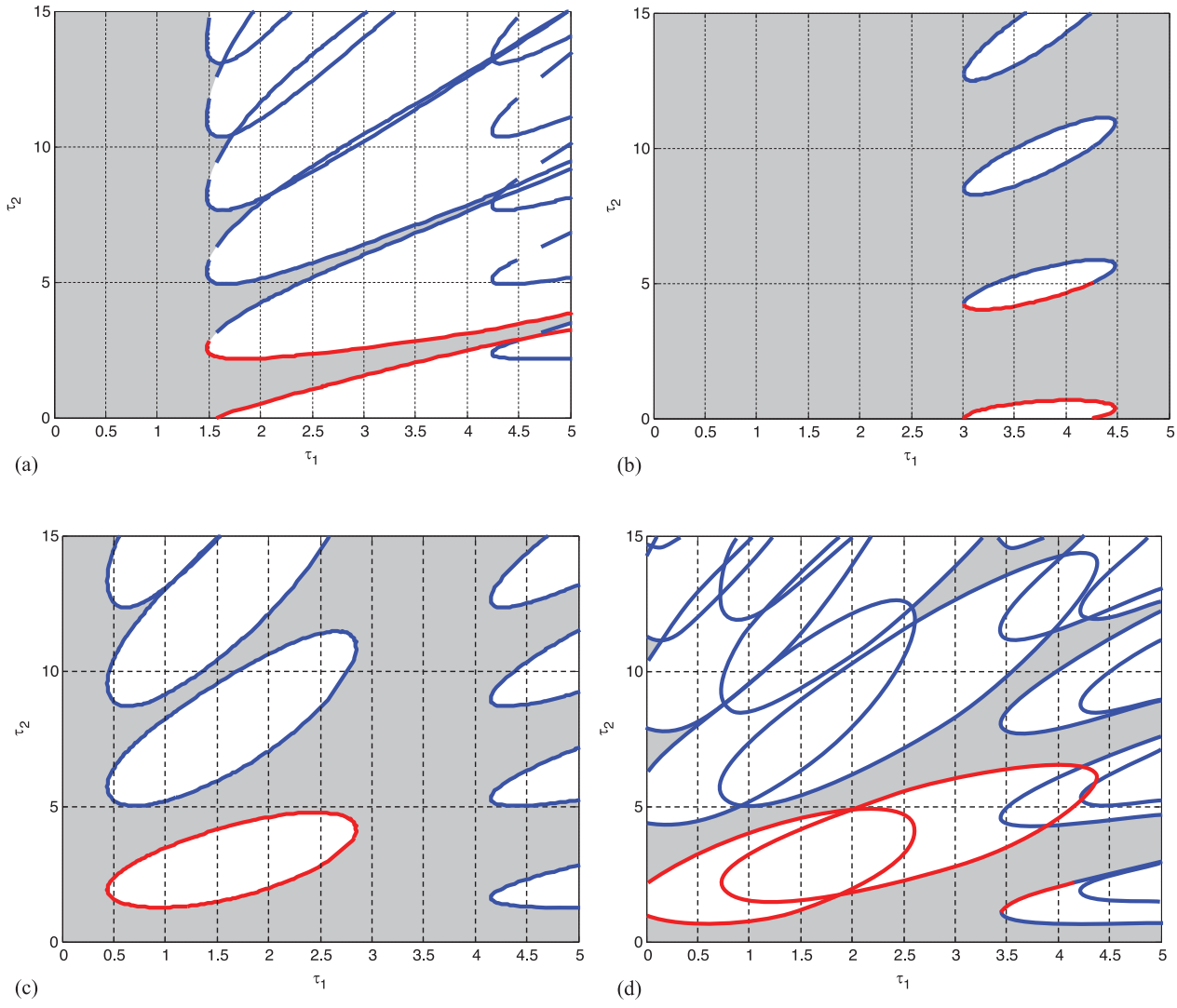

Figure 3. Stability regions in the domain of the time delays for the characteristic equation factors generated by the communication topology of Figure 1 with $P = 2$ and $D = 0.8$. (a) $\lambda = 1$, (b) $\lambda = 0.38$, (c) $\lambda = -0.5$ and (d) $\lambda = -0.44 \pm 0.37i$.

a given set of delays further. This is, however, a paradoxical and counter-intuitive proposition. Some authors have also reported cases in which introducing delays benefits the transient performance of the system. Liu and Tian (2009), for example, show that introducing and increasing some self-delays reduce the convergence time. However, they do not offer an analytical framework to do this, and reason their results on a pointwise trial-and-error effort at some fixed delay compositions. Differently here, we present a brief discussion, and a structured procedure, on how one should perform the selection of the stabilising delay prolongation, also known as the *delay scheduling*.

We first establish the stability chart of the particular topology following the steps described in the earlier sections (i.e., Figure 4). As we inquire the effective delay composition *in situ*, we can ascertain the instantaneous stability of the system from this chart. Let us assume the given delays at the moment render the operating point **b** in Figure 4, which is unstable. The controller can now impose an intentional delay to bring the combined delay composition to point **c**, where the stability is recovered. This delay adjustment operation is named the *delay scheduling* method.

The placement of the operating point by introducing prolonged delays (i.e., the choice of point **c**) offers an additional freedom to the control designer. As we make the selection for point **c**, we also wish to operate at a stable point which facilitates the fastest disturbance rejection capabilities to the swarm formation. As such, we query the operational question of the speed for reaching consensus. This information is crucial in tuning the delays in order to achieve the most desirable (expeditious) speed of reaching consensus. We explain next how the stability map, which is given in Figure 5, can be utilised for this purpose.

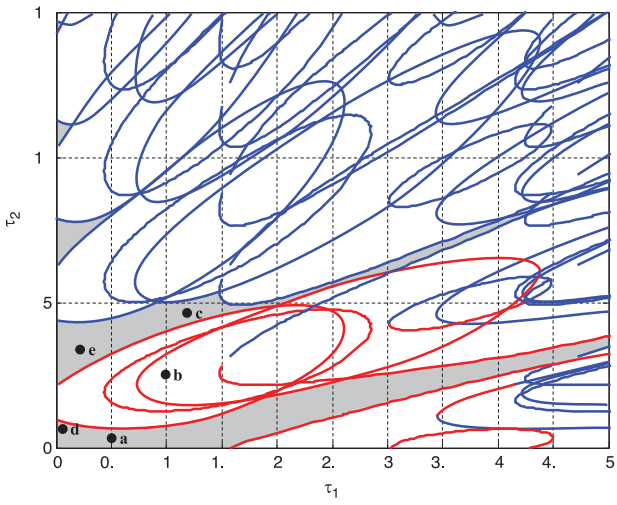

Figure 4. Complete stability picture for the communication topology of Figure 1, using $P = 2$ and $D = 0.8$.

The speed of group consensus is dictated by the dominant characteristic root of the infinite dimensional system given in Equation (2). The dominant root of the characteristic equation (4) is the rightmost of the dominant roots of all the factors given by Lemma 1. These factors, $q_j(s, P, D, \tau_1, \tau_2, \lambda_j)$, however, are quasi-polynomials and their roots can only be determined by some numerical approximations (Breda, Maset, & Vermiglio, 2006; Vyhlidal & Zítek, 2009). We deploy the QPmR (Quasi-Polynomial, mapping-based Root finding) algorithm (Vyhlidal & Zítek, 2009) here. For some selected grid points (τ_1, τ_2), we determine the dominant root and its real part, $\mathrm{Re}(s_{\mathrm{dom}})$. For the example case study, the variation of $\mathrm{Re}(s_{\mathrm{dom}})$ is displayed in Figure 6.

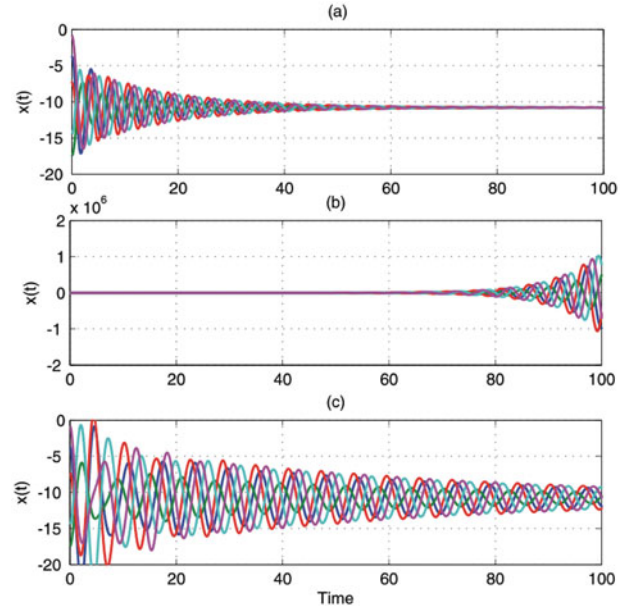

Figure 5. Example agents' behaviour for three different points in Figure 4.

We wish to stress several observations over this figure:

(1) The stable operating zones of Figure 4 are re-created by the grid point evaluations of the dominant roots over a (τ_1, τ_2) space of interest. Obviously, within these regions $\mathrm{Re}(s_{\mathrm{dom}}) < 0$ is satisfied.

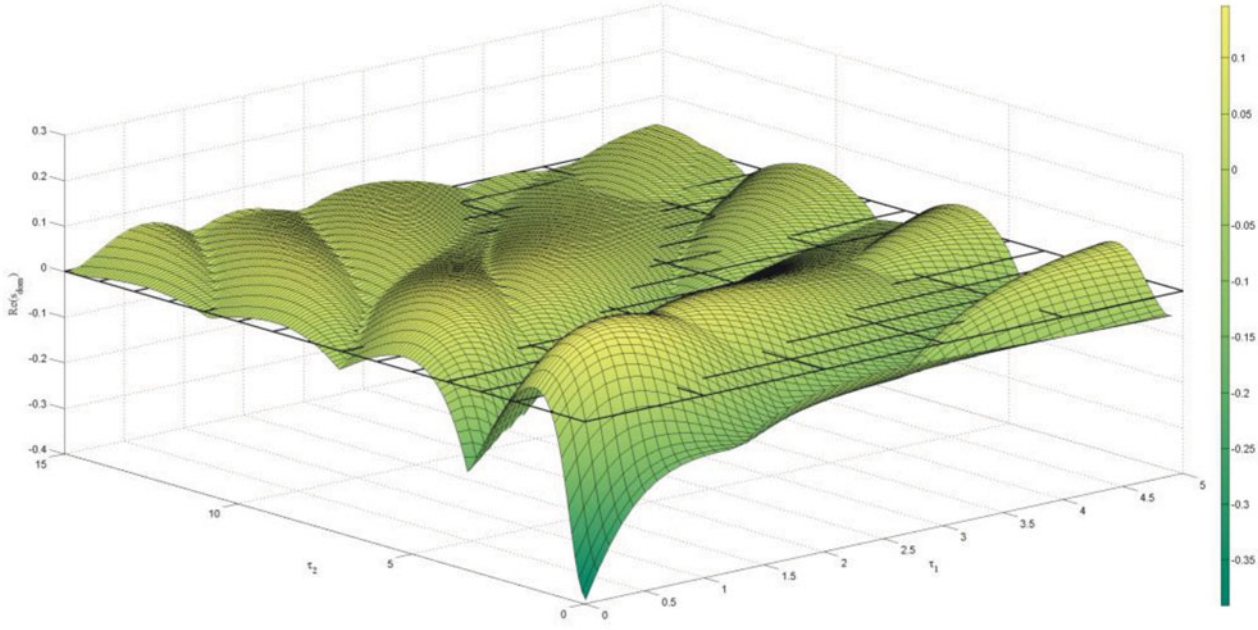

Figure 6. Variation of the real part of the dominant root for the complete system. Grid represents the zero real part partitioning.

(2) The dominant time constant, $-1/\mathrm{Re}(s_{\mathrm{dom}})$, i.e., the basis for the 'speed of reaching consensus', in the example case **a** of Figure 4 concurs with the location of s_{dom}. For this point $\mathrm{Re}(s_{\mathrm{dom}}) = -0.06$, which corresponds to a time constant of 16.4 seconds, and a settling time (the speed of consensus) of approximately 66 seconds, as observed in Figure 5.

(3) Although the fastest consensus is achieved at $\tau_1 = \tau_2 = 0$, larger delays do not always mean longer consensus time. Compare, for instance, points **d** and **e** in Figure 4. For the first one, $(\tau_1, \tau_2) = (0.05,\ 0.8)$ and $\mathrm{Re}(s_{\mathrm{dom}}) = -0.04$, whereas for **e** $(\tau_1, \tau_2) = (0.1,\ 3.5)$ and $\mathrm{Re}(s_{\mathrm{dom}}) = -0.05$. Obviously, the latter makes a better choice from the point of speed of reaching consensus, although both delays are larger in the second case.

This example captures the core concept of *delay scheduling* (Olgac & Cavdaroglu, 2011). In order to trigger the delay selections under this methodology, one has to have the complete system stability tableau in the delay space, i.e., Figures 4 and 6. The procedures discussed in this paper to obtain the stability outlook are very efficient. Just to give an idea to the reader, in order to create one of the stability tableaus in Figure 3 for an arbitrary factor in Equation (6), we encounter an average computational cost, of 1 second of CPU time (including display operations) on a 2.93 GHz Intel Core 2 Duo based computer with 4 GB RAM, running Matlab 2010a without an elaborate code optimisation for speed.

Another test is conducted over this example case study, by simply demonstrating the real-time control repercussions of the delay scheduling phenomenon. Figure 7 shows the time traces of the agent behaviour under delay combinations of (0.5, 0.5), (1, 2.5) and (1.3, 4.5) seconds in the time intervals A (from 0 to 6 seconds), B (from 60 to 100 seconds)

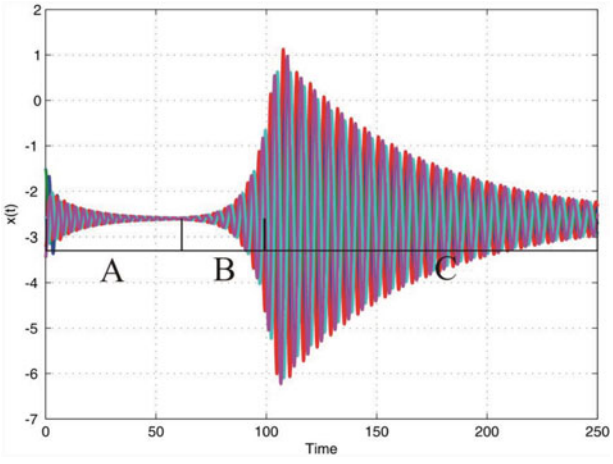

Figure 7. Behaviour of the agents when the delay changes, showing the effect of delay scheduling.

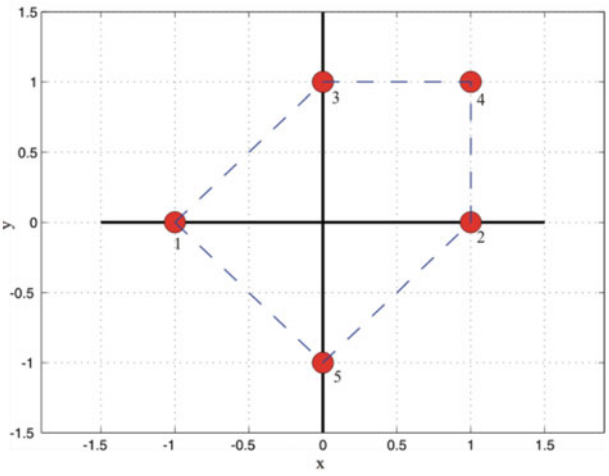

Figure 8. Desired formation for the example case.

and C (beyond 100 seconds), respectively. In duration of A, a perfectly stable group dynamic is observed. As both delays increase in the time interval B, the dynamics become unstable only to be recovered by prolonging the delays even further for the time interval C. This intriguing phenomenon, the *delay scheduling*, deserves much deeper understanding than the space allows. Interesting discussions on the steps of control decisions for delay scheduling, selection standardisation, the transition behaviour during scheduling and alike are reserved for future investigations and reporting.

5.2. *Formation control*

In order to test the formation control algorithm of Section 4, we start by defining a desired formation, which is presented in Figure 8. If these final positions (steady state) in x and y coordinates were selected as

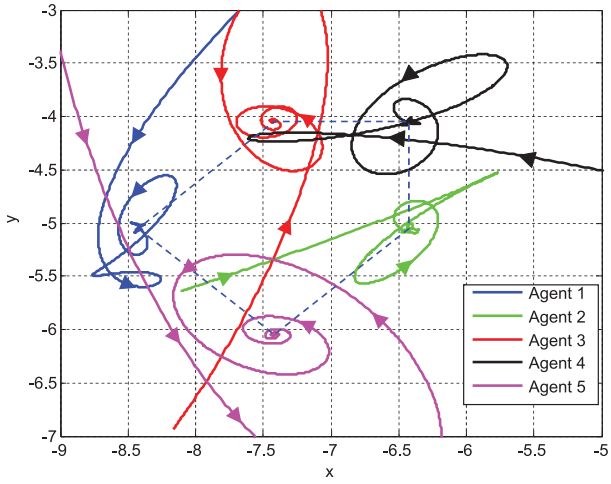

Figure 9. Traces of the motion of the agents for the formation control example case (centroid is uncontrolled).

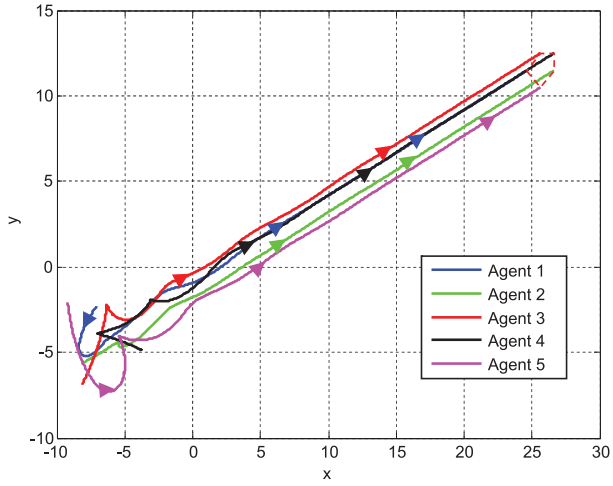

Figure 10. Traces of the motion of the agents for the formation control example case with centroid forcing (centroid is controlled).

$\mathbf{x}_{ss} = [-1 \;\; 1 \;\; 0 \;\; 1 \;\; 0]$ and $\mathbf{y}_{ss} = [0 \;\; 0 \;\; 1 \;\; 1 \;\; -1]$, they would impose the final values for the disagreement terms as $\boldsymbol{\xi}^x_{ss} = [0.88 \;\; 0.28 \;\; 1.51 \;\; 1.11 \;\; 0.39]$ in x-direction and $\boldsymbol{\xi}^y_{ss} = [0.68 \;\; -1.63 \;\; 2.20 \;\; -2.21 \;\; 0.61]$ in y-direction. This leads to the forcing terms in the transformed domain $\boldsymbol{\Phi}^x = [0 \;\; 0 \;\; 0 \;\; -0.31 \;\; 0 \;\; 4.55 \;\; 0 \;\; 3.32 \;\; 0 \;\; 0.48]^T$ and $\boldsymbol{\Phi}^y = [0 \;\; 0 \;\; 0 \;\; -0.633 \;\; 0 \;\; 5.13 \;\; 0 \;\; -0.63 \;\; 0 \;\; 0.75]^T$ as evaluated from Equation (22). The actual forcing terms to be applied to the agents are found as discussed in Section 4: $\mathbf{F}^x = [0 \;\; -4 \;\; 0 \;\; 2 \;\; 0 \;\; -2 \;\; 0 \;\; 2 \;\; 0 \;\; -1]^T$ and $\mathbf{F}^y = [0 \;\; 0 \;\; 0 \;\; -1 \;\; 0 \;\; 1 \;\; 0 \;\; 2 \;\; 0 \;\; -3]^T$. These forcing terms with a delay combination $(\tau_1, \tau_2) = (0.1, \; 0.1)$ yield the results presented in Figure 9. It is clear that the agents reach the desired formation.

As mentioned in the previous section, by having $\boldsymbol{\varphi}^x_1 = \boldsymbol{\varphi}^y_1 = 0$, the agents end in a static formation, with the centroid at a position dictated by the initial conditions. This is the behaviour displayed also in Figure 9. If the forcing terms corresponding to the centroid are also set to constant values, the steady-state value of the factor corresponding to $\lambda = 1$ presents constant velocity and linearly increasing position, due to the static root at the origin. This results in a coordinated motion of the agents, with constant velocity while keeping the formation (i.e., the marginal stability). An example of this behaviour is presented in Figure 10. For this case, the centroid forcing terms were set to $\boldsymbol{\varphi}^x_1 = 1$ and $\boldsymbol{\varphi}^y_1 = 0.5$, which create a consensus in the group with a constant speed of 2.5 along the vector direction (2, 1).

6. Conclusions

In this paper, we study an n-agent system, considering directed communication topologies and multiple time delays. A consensus protocol for this group of agents is presented and extended to a formation control algorithm. Two constant and rationally independent time delays are taken into account, one affecting the position and the other the velocity information exchange channels.

The stability analysis of the consensus protocol and the ensuing formation control are based on an enabling factorisation of the characteristic equation of the system. The factors involved are only distinguished from one another by a set of n eigenvalues of a particular system matrix. This factorisation procedure reduces the complexity of the stability analysis considerably and converts it into a repeated and efficient application of a reduced-order core analysis n times. The CTCR methodology and the new concept of SDS are utilised to obtain the exact stability bounds in the domain of the time delays. As the number of agents increases, the only remaining procedural complexity is shown to be in determining the set of eigenvalues of a given system matrix. Numerical example cases verify these claims.

An extension on this consensus protocol to create a formation is also presented. It is achieved by introducing some additional forcing terms that keep the agents apart from one another. A new algorithm for the selection of these forcing terms is also detailed. Example case studies show the viability of this idea.

We also look at the parametric variations of the speed of reaching consensus (as well as disturbance rejection capabilities) as the two delays in the communication channels vary. It becomes clear that the control designer may have the option to intentionally prolong the delays to achieve a faster consensus (a procedure called the *delay scheduling*). This ability requires, however, the precise knowledge of the parametric stability picture of the system (which is a major contribution of this paper) as well as the corresponding consensus speed variations.

Acknowledgements

The authors wish to express their gratitude for the thorough reviews on this manuscript by the anonymous reviewers. Their remarks matured the text considerably.

Funding

This work was supported in part by Army Research Office (ARO) [grant number W911NF-07-1-0557].

Notes on contributors

Rudy Cepeda-Gomez received the BSc degree from Industrial University of Santander (UIS), Bucaramanga, Colombia, in 2006, the MSc degree from Pontifical Xaverian University, Bogotá, Colombia, in 2010, both in electronic engineering, and the PhD degree in mechanical engineering from the University of Connecticut, Storrs, CT, in 2012. He was a full-time lecturer and leader of the research group with the Mechatronic Engineering Department at Santo Tomás University, Bucaramanga, Colombia, and a part-time lecturer at UIS (2012–2013). In January 2014, he

joined the Institute for Automation at the University of Rostock, Germany. Dr Cepeda-Gomez is a member of ASME, IEEE and the honour society of Phi-Kappa-Phi.

Nejat Olgac received the MSc degree from Technical University of Istanbul, Turkey in 1972 and the Dr. Eng. Sci. degree from Columbia University, New York, NY, in 1976, both in Mechanical Engineering. He is a professor with the Mechanical Engineering Department of the University of Connecticut (since 1981). Dr Olgac holds four patents (1995-96-99-2011) on the delayed resonator active vibration suppression technique and on a method for facilitating chatter stability mapping in a simultaneous machining application. He was visiting professor at INRIA (Sophia Antipolis, France, 1988–1989), SEW Eurodrive Fellow–guest professor at Technical University of Munich (Germany, 1995–1996) and visiting professor at Harvard University (2002–2003). He presently sits on the Editorial Boards of *IET Control Theory and Applications*, *International Journal of Mechatronics and Manufacturing Systems* (2006–present), *Journal of Vibration and Control* (2005–present) and *Bulletin of Applied Mechanics* (2010–present). He was the general chair of ASME-Dynamic Systems and Control Conference 2013 and IFAC Time Delay Systems Workshop (TDS 2012). Prof. Olgac is a member of the Connecticut Academy of Science and Engineering, fellow of ASME and senior member of IEEE.

References

Agaev, R., & Chebotarev, P. (2005). On the spectra of nonsymmetric Laplacian matrices. *Linear Algebra and its Applications, 399*, 157–168.

Bell, H.E. (1965). Gershgorin's theorem and zeros of polynomials. *American Mathematical Monthly, 74*, 292–295.

Biggs, N. (1993). *Algebraic graph theory*. New York, NY: Cambridge University Press.

Breda, D., Maset, S., & Vermiglio, A.R. (2006). Pseudo spectral differencing methods for characteristic roots of delay differential equations. *SIAM Journal on Scientific Computing, 27*, 482–495.

Cao, Y., Yu, W., Wei, R., & Cheng, G. (2013). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics, 9*, 427–438.

Cepeda-Gomez, R., & Olgac, N. (2011a). An exact method for the stability analysis of linear consensus protocols with time delay. *IEEE Transactions on Automatic Control, 56*, 1734–1740.

Cepeda-Gomez, R., & Olgac, N. (2011b). Exhaustive stability analysis in a consensus system with time delay and irregular topologies. *International Journal of Control, 84*, 746–757.

Dimarogonas, D.V., & Kyriakopoulos, K.J. (2009). Inverse agreement protocols with application to distributed multi-agent dispersion. *IEEE Transactions on Automatic Control, 54*, 657–663.

Ergenc, A.F., Olgac, N., & Fazelinia, H. (2007). Extended Kronecker summation for cluster treatment of LTI systems with multiple delays. *SIAM Journal on Control and Optimization, 46*, 143–155.

Fax, A., & Murray, R. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control, 49*, 1465–1476.

Fazelinia, H. (2007). *A novel stability analysis of systems with multiple time delays and its applications to high speed milling chatter* (PhD dissertation). Department of Mechanical Engineering, University of Connecticut, Storrs, CT.

Fazelinia, H., Sipahi, R., & Olgac, N. (2007). Stability robustness analysis of multiple time-delayed systems using building block concept. *IEEE Transactions on Automatic Control, 52*, 799–810.

Gabow, H.N., & Myers, E.W. (1978). Finding all spanning trees of directed and undirected graphs. *SIAM Journal on Computing, 7*, 280–287.

Lin, P., & Jia, Y. (2009a). Further results on decentralized coordination in networks of agents with second order dynamics. *IET Control Theory and Applications, 3*, 957–970.

Lin, P., & Jia, Y. (2009b). Consensus of second order discrete multi-agent systems with nonuniform time-delay and dynamically changing topologies. *Automatica, 45*, 2154–2158.

Lin, P., Jia, Y., Du, J., & Juan, J. (2008). Distributed control of multi-agent systems with second order dynamics and delay-dependent communications. *Asian Journal of Control, 10*, 254–259.

Liu, C.-L., & Tian, Y.-P. (2009). Formation control of multi-agent systems with heterogeneous communication delays. *International Journal of Systems Science, 40*, 627–636.

Marcus, M., & Minc, H. (1996). *A survey of matrix theory and matrix inequalities*. New York, NY: Dover.

Meng, Z., Ren, W., Cao, Y., & Zheng, Y. (2011). Leaderless and leader-follower consensus with communication and input delays under a directed network topology. *IEEE Transactions on Systems, Man and Cybernetics – Part B, 41*, 75–88.

Münz, U., Papachristodoulou, A., & Allgower, F. (2010). Delay robustness in consensus problems. *Automatica, 45*, 1252–1256.

Olfati-Saber, R. (2006). Flocking for multi-agent systems: Algorithms and theory. *IEEE Transactions on Automatic Control, 51*, 401–420.

Olfati-Saber, R., & Murray, R. (2004). Consensus problems in networks of agents with switching topology and time delays. *IEEE Transactions on Automatic Control, 49*, 1520–1533.

Olgac, N., & Cavdaroglu, M.E. (2011). Full state feedback controller design with delay scheduling for cart and pendulum dynamics. *Mechatronics, 21*, 38–47.

Pavone, M., & Frazzolli, E. (2007). Decentralized policies for geometric pattern formation and path coverage. *Journal of Dynamic Systems, Measurement, and Control, 129*, 633–643.

Peng, K., & Yang, Y. (2009). Leader follower consensus problem with a varying velocity leader and time-varying delays. *Physica A, 388*, 193–208.

Ren, W. (2007). Consensus strategies for cooperative control of vehicle formations. *IET Control Theory and Applications, 1*, 505–512.

Ren, W., & Beard, R. (2005). Consensus seeking under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control, 50*, 655–661.

Schaefer, R.D. (1996). *An introduction to nonassociative algebras*. New York, NY: Dover.

Sepulchre R., Paley D.A., & Leonard N.E. (2007). Stabilization of planar collective motion: All-to-All communication. *IEEE Transactions on Automatic Control, 52*, 811–824.

Sepulchre R., Paley D.A., & Leonard N.E. (2008). Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control, 53*, 706–719.

Sun, Y., & Wang, L. (2009a). Consensus problems in networks of agents with double integrator dynamics and time varying delays. *International Journal of Control, 82*, 1937–1945.

Sun, Y., & Wang, L. (2009b). Consensus of multi-agent systems in directed networks with nonuniform time varying delays. *IEEE Transactions on Automatic Control, 54*, 1607–1617.
Toker, O., & Ozbay, H. (1996). Complexity issues in robust stability of linear delay-differential systems. *Mathematics of Control, Signals and Systems, 9*, 386–400.
Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters, 76*, 1226–1229.
Vyhlidal, T., & Zítek, P. (2009). Mapping based algorithm for large scale computation of quasi-polynomial zeros. *IEEE Transaction on Automatic Control, 54*, 171–177.

Appendix. Proofs of the lemmas

Lemma 1 (Factorisation property)**:** *The characteristic equation of system* (2) *can always be expressed as the product of a set of second- and fourth-order factors:*

$$\begin{aligned}
&Q(s,P,D,\tau_1,\tau_2)\\
&=\det(s\mathbf{I}_{2n}-\mathbf{A}-\mathbf{B}_1e^{-\tau_1 s}-\mathbf{B}_2e^{-\tau_2 s})\\
&=\prod_{j=1}^{\ell+m} q_j(s,P,D,\tau_1,\tau_2,\lambda_j)=\prod_{j=1}^{\ell}\left[s^2+Ds+P-\lambda_j\right.\\
&\times\left.\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)\right]\times\prod_{j=\ell+1}^{m}\left[s^4+2Ds^3\right.\\
&+(D^2+2P)s^2+2DPs+P^2-2\mathrm{Re}(\lambda_j)\\
&\times(s^2+Ds+P)(Dse^{-\tau_2 s}+Pe^{-\tau_1 s})+|\lambda_j|^2\\
&\times\left.(Dse^{-\tau_2 s}+Pe^{-\tau_1 s})^2\right]=0
\end{aligned}\tag{A1}$$

where λ_j *represents the eigenvalues of* $\boldsymbol{C}=\boldsymbol{\Delta}^{-1}\boldsymbol{A}_\Gamma$. *This matrix has* ℓ *real eigenvalues, denoted by* $j=1,2,3,\ldots,\ell$ *and* m *complex conjugate eigenvalue pairs* (λ_j,λ_j^*), $j=\ell+1,\ \ell+2,\ \ell+3,\ldots,\ell+m$. *Then,* $n=2m+\ell$.

Proof: Let $\mathbf{T}$ be the non-singular similarity transformation matrix that converts $\mathbf{C}$ into its Jordan canonical form: $\boldsymbol{\Lambda}=\mathbf{T}^{-1}\mathbf{C}\mathbf{T}$. The matrix $\boldsymbol{\Lambda}\in\Re^{n\times n}$is block diagonal in the form

$$\boldsymbol{\Lambda}=\begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0\\
0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & 0 & \lambda_\ell & 0 & \cdots & 0\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{\ell+1} & \cdots & \mathbf{0}\\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_{\ell+m}
\end{bmatrix}\tag{A2}$$

where λ_j, $j=1,2,3,\ldots,\ell$, are the size-1 Jordan blocks corresponding to the real eigenvalues and

$$\mathbf{J}_j=\begin{bmatrix}\mathrm{Re}(\lambda_j) & -\mathrm{Im}(\lambda_j)\\ \mathrm{Im}(\lambda_j) & \mathrm{Re}(\lambda_j)\end{bmatrix},\qquad j=\ell+1,\ \ell+2,\ldots,\ell+m\tag{A3}$$

are the size-2 (2×2) Jordan blocks corresponding to the m complex conjugate eigenvalue pairs. If we implement the state transformation $\mathbf{x}(t)=(\mathbf{T}\otimes\mathbf{I}_2)\boldsymbol{\xi}(t)$ in Equation (3), it becomes

$$\begin{aligned}
\dot{\boldsymbol{\xi}}(t)&=\left(\mathbf{T}^{-1}\otimes\mathbf{I}_2\right)\left(\mathbf{I}_n\otimes\begin{bmatrix}0 & 1\\ -P & -D\end{bmatrix}\right)(\mathbf{T}\otimes\mathbf{I}_2)\boldsymbol{\xi}(t)\\
&+\left(\mathbf{T}^{-1}\otimes\mathbf{I}_2\right)\left(\mathbf{C}\otimes\begin{bmatrix}0 & 0\\ P & 0\end{bmatrix}\right)(\mathbf{T}\otimes\mathbf{I}_2)\boldsymbol{\xi}(t-\tau_1)\\
&+\left(\mathbf{T}^{-1}\otimes\mathbf{I}_2\right)\left(\mathbf{C}\otimes\begin{bmatrix}0 & 0\\ 0 & D\end{bmatrix}\right)(\mathbf{T}\otimes\mathbf{I}_2)\boldsymbol{\xi}(t-\tau_2)
\end{aligned}\tag{A4}$$

A convenient property of the Kronecker product is $(\mathbf{U}\otimes\mathbf{V})(\mathbf{W}\otimes\mathbf{Z})=\mathbf{U}\mathbf{W}\otimes\mathbf{V}\mathbf{Z}$ where the pairs of square matrices $(\mathbf{U},\mathbf{W})$ and $(\mathbf{V},\mathbf{Z})$ are of the same dimensions (Schaefer, 1996). Using this property in Equation (A4), one obtains

$$\begin{aligned}
\dot{\boldsymbol{\xi}}(t)&=\left(\mathbf{I}_n\otimes\begin{bmatrix}0 & 1\\ -P & -D\end{bmatrix}\right)\boldsymbol{\xi}(t)+\left(\boldsymbol{\Lambda}\otimes\begin{bmatrix}0 & 0\\ P & 0\end{bmatrix}\right)\boldsymbol{\xi}(t-\tau_1)\\
&+\left(\boldsymbol{\Lambda}\otimes\begin{bmatrix}0 & 0\\ 0 & D\end{bmatrix}\right)\boldsymbol{\xi}(t-\tau_2)
\end{aligned}\tag{A5}$$

Since $\mathbf{I}$ and $\boldsymbol{\Lambda}$ are diagonal and block diagonal matrices, respectively, Equation (A5) is block-diagonalised, thus it can be represented as a set of $\ell+m$ dynamically decoupled subsystems with the following dynamics:

$$\begin{aligned}
\dot{\xi}_j(t)&=\begin{bmatrix}0 & 1\\ -P & -D\end{bmatrix}\boldsymbol{\xi}_j(t)+\lambda_j\begin{bmatrix}0 & 0\\ P & 0\end{bmatrix}\boldsymbol{\xi}_j(t-\tau_1)\\
&+\lambda_j\begin{bmatrix}0 & 0\\ 0 & D\end{bmatrix}\boldsymbol{\xi}_j(t-\tau_2),\quad j=1,2,\ldots,\ell
\end{aligned}\tag{A6a}$$

$$\begin{aligned}
\dot{\xi}_j(t)&=\left(\mathbf{I}_2\otimes\begin{bmatrix}0 & 1\\ -P & -D\end{bmatrix}\right)\boldsymbol{\xi}_j(t)\\
&+\left(\mathbf{J}_j\otimes\begin{bmatrix}0 & 0\\ P & 0\end{bmatrix}\right)\boldsymbol{\xi}_j(t-\tau_1)\\
&+\left(\mathbf{J}_j\otimes\begin{bmatrix}0 & 0\\ 0 & D\end{bmatrix}\right)\boldsymbol{\xi}_j(t-\tau_2)\\
&j=\ell+1,\ell+2,\ldots,m
\end{aligned}\tag{A6b}$$

The common characteristic equation corresponding to the subsystems with dynamics given by Equation (A6a) is

$$\begin{aligned}
&q_j(s,P,D,\tau_1,\tau_2,\lambda_j)\\
&=s^2+Ds+P-\lambda_j(Dse^{-\tau_2 s}+Pe^{-\tau_1 s})=0
\end{aligned}\tag{A7a}$$

whereas Equation (A6b) generates the characteristic equation as

$$\begin{aligned}
&q_j\left(s,P,D,\tau_1,\tau_2,\lambda_j\right)\\
&=s^4+2Ds^3+\left(D^2+2P\right)s^2+2DPs+P^2\\
&-2\mathrm{Re}\left(\lambda_j\right)\left(s^2+Ds+P\right)\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)\\
&+\left|\lambda_j\right|^2\left(Dse^{-\tau_2 s}+Pe^{-\tau_1 s}\right)^2=0
\end{aligned}\tag{A7b}$$

The characteristic equation of the complete system is then formed by the product of these $\ell+m$ individual factors.

Lemma 2 (Group behaviour)**:** *Assume that the communication topology has at least one spanning tree and every agent has at least one informer. Then, the agents in the group reach a consensus if and only if the factor* (8) *is marginally stable and all the remaining factors of Equation* (6) *are stable. Furthermore, the consensus value will be* $\lim_{t\to\infty} x_j(t) = \lim_{t\to\infty} \sqrt{n}\xi_1 (t = \infty)$*, whereas the other states tend to* $\xi_j (t = \infty) = 0$ *for* $j = 2, 3, \ldots, n$.

Proof: First, we prove the necessity condition. From the definition of the state $\boldsymbol{\xi}$, we have $[\,\xi_1\ \xi_2\ \cdots\ \xi_n\,]^T = \mathbf{T}^{-1}[\,x_1\ x_2\ \cdots\ x_n\,]^T$. If consensus is reached, the agents have a common steady-state value which we denote by $\bar{x} = \lim_{t\to\infty} x_j(t)$, $j = 1, 2, \ldots, n$. Then,

$$\lim_{t\to\infty} \left[\,\xi_1(t)\ \xi_2(t)\ \ldots\ \xi_n(t)\,\right]^T = \bar{x}\,\mathbf{T}^{-1}\left[\,1\ 1\ \ldots\ 1\,\right]^T \quad \text{(A8)}$$

Since the communication topology is assumed to have a spanning tree, $\lambda = 1$ is a simple eigenvalue of $\mathbf{C}$, corresponding to the eigenvector $\mathbf{t}_1 = 1/\sqrt{n}[\,1\ 1\ \ldots\ 1\,]^T$, which is also the first column of the transformation matrix $\mathbf{T}$. Since $\mathbf{T}^{-1}[\,1\ 1\ \ldots\ 1\ \ldots]^T = \sqrt{n}\mathbf{T}^{-1}\mathbf{t}_1 = \sqrt{n}\,[\,1\ 0\ \ldots\ 0\,]^T$, Equation (A8) leads to $\lim_{t\to\infty} \xi_1(t) = \sqrt{n}\bar{x}$, which indicates marginal stability for Equation (8), while $\lim_{t\to\infty} \xi_j(t) = 0$ for $j = 2, 3, \ldots, n$, indicating asymptotic stability in the other factors of Equation (6).

Next, the sufficiency clause is proven. If Equation (8) is marginally stable (due to the characteristic root at the origin) and all the other factors in Equation (6) are stable, the steady-state value of ξ_1 will be constant whereas the remaining $\xi_j(t)$, $j = 2, 3, \ldots, n$, tends to zero. Then, $\lim_{t\to\infty}[\,\xi_1(t)\ \xi_2(t)\ \ldots\ \xi_n(t)\,]^T = [\,\sqrt{n}\bar{x}\ 0\ \ldots\ 0\,]^T$. Using the inverse transformation, the original states become $\lim_{t\to\infty}[\,x_1(t)\ x_2(t)\ \cdots\ x_n(t)\,]^T = \mathbf{T}[\,\sqrt{n}\bar{x}\ 0\ \ldots\ 0\,]^T = \sqrt{n}\,\bar{x}\,\mathbf{t}_1 = [\,\bar{x}\ \bar{x}\ \ldots\ \bar{x}\,]^T$ implying that the agents reach consensus.

Lemma 3 (Topologies without spanning trees)**:** *If the given communication topology does not have a spanning tree, the control logic described by Equation* (1) *cannot result in consensus.*

Proof: Assume that the communication topology does not have a spanning tree but it can be separated into $r < n$ components, i.e., disjoint subgraphs within the graph that have spanning trees themselves. Then, the matrix $\mathbf{C}$ has a repeated eigenvalue 1 with multiplicity of r, i.e., one for each component. Since $\mathbf{C}$ is stochastic, the eigenvalue 1 is always semi-simple (Marcus & Minc, 1996), there are r linearly independent eigenvectors corresponding to this repeated eigenvalue. These eigenvectors are assigned to the first r columns of $\mathbf{T}$ matrix for convenience. These columns should have the form $\mathbf{t}_j = [a_{j1}\ a_{j2}\ldots a_{jn}]^T, j = 1, 2, \ldots, r$, where a_{jk} is 1, if agent k belongs to the component j and 0 otherwise (Biggs, 1993). Each one of these r eigenvalues creates a factor of form (9) in the characteristic equation of the system. These factors represent the dynamics of the transformed states $\xi_j, j = 1, 2, 3, \ldots, r$, which are at best marginally stable (for marginality, recall the stationary zero characteristic root of Equation (8)).

If the remaining $n-r$ factors, which represent the disagreement dynamics, are all stable, the steady-state value of the system in the transformed domain is $\lim_{t\to\infty}[\xi_1(t)\ \xi_2(t)\ldots\xi_r(t)\ \xi_{r+1}(t)\ldots\xi_n(t)]^T = [\bar{y}_1\ \bar{y}_2\ldots\bar{y}_r\ 0\ldots 0]^T$, where $\bar{y}_j \neq \bar{y}_k$ in general. One can recover the original states using the inverse transformation, $\lim_{t\to\infty}[x_1(t)\ x_2(t)\ldots x_n(t)]^T = \mathbf{T}\,[\bar{y}_1\ \bar{y}_2\ldots\bar{y}_m\ 0\ldots 0]^T$. This leads to $\lim_{t\to\infty}[x_1(t)\ x_2(t)\ldots x_n(t)]^T = \bar{y}_1\mathbf{t}_1 + \bar{y}_2\mathbf{t}_2 + \cdots + \bar{y}_r\mathbf{t}_r$. Due to the orthogonal construction of the $\mathbf{t}_j$ vectors and the $\bar{y}_j \neq \bar{y}_k$ condition, the agents typically do not reach a common state. For the degenerate case of $\bar{y}_j = \bar{y}_k$, then jth and kth subgroups have a joint steady-state behaviour, but the rest of the group does not. Therefore, consensus is not achieved.