ARTICLE IN PRESS
JID: EJCON [m5G;September 22, 2017;16:47]
European Journal of Control 000 (2017) 1–11
Contents lists available at ScienceDirect
European Journal of Control
journal homepage: www.elsevier.com/locate/ejcon

Consensus in networks with arbitrary time invariant linear agents

Miguel Parada Contzen

Control Systems Group of the Technische Universität, Berlin, Germany

A R T I C L E I N F O

Article history:
Received 7 May 2017
Revised 26 June 2017
Accepted 25 August 2017
Available online xxx

Recommended by A.G. Aghdam

Keywords:
Consensus
Multiagents systems
Linear systems
Distributed control
Graph theory

A B S T R A C T

In this paper, we study the consensus problem from a general control theoretical perspective. For that we identify three entities: the multiagents network that constitutes the control plant, consensus as a control objective, and the consensus algorithm as a feedback controller for the network. Consensus is redefined through the idea of organization (a linear transformation) to define an error vector that resumes the characteristics of the network. With this formulation, we can translate the general consensus problem into a stability problem and, from there, use classical Control Theory to analyze the case of agents with arbitrary linear time invariant dynamics (and not only integrator dynamics) and Laplacian algorithms. The paper is complemented with numeric examples to illustrate the proposed analysis methodology.

1. Introduction

The topic of *consensus* in multi agents systems had gain much attention in the control society during the last decade. The analogy of a swarm of birds is a useful way to explain the main characteristic of the problem: a group of similar systems (or *agents*) agree to coordinate some important variables through a given information exchange strategy (or *algorithm*). The publication of books like [14,18,19,28] shows that the topic has already reached an advanced state. However, it remains a popular area of research as shown in the review papers [5,15] where more than three hundred references are quoted. Most of the work in the area is based on Graph Theoretical approaches to the problem and single or double integrators dynamics. Examples of this are the already quoted publication, and an increasing number of papers such as [1,9–11,17].

Consensus can be understood as a control objective in the same way as stability or robustness in classical control. That is, the definition of consensus is independent of the agent's dynamics or the methodology that the agents follow to reach this objective. It is however not an exception in the field, *e.g.* [8,15,19, etc.], to find definitions not only in terms of the output signals but also in terms of specific dynamics (usually integrators) and specific consensus algorithms (usually Laplacian algorithms). That is, not as a control objective used for synthesis of controllers, but as a property of particular control plants with particular controllers. Although some publications, *e.g.* [12,13,21–23,26,27,29], extend the graph theoretical approaches to systems with more general linear dynamics, the particular cases with which these deal, makes it difficult to extend the results to more general cases.

Furthermore, consensus can be intuitively compared with the equilibrium point of a system that resumes the characteristics of the whole network. However, explicitly reducing consensus to a stability problem, is not typically addressed in the existing works. Nevertheless, in some recent papers, *e.g.* [3,20,24,25,29], consensus is studied as the stability of a *differences vector* between the outputs of one of the agents and the rest of them.

In this paper, this idea is further exploited to formally translate the consensus problem into a classical stability one through the introduction of an analysis tool that we named *organization*. The idea was partially introduced by the author in a conference publication [16], but here it is enriched to obtain analytical conditions to verify if agents with arbitrary linear time-invariant dynamics can reach consensus over all of its outputs. This is done by extending the notion of weighted graphs to include matrix weights, and by identifying three different entities: A multiagents control plant, consensus as a control objective, and a consensus algorithm as a distributed feedback controller. From here, the problem can be studied by means of standard control theory allowing to drop restrictive assumptions on the dynamics of the agents.

After this introduction, Section 2 presents a summary of Graph Theoretical concepts that are needed to characterize the consensus problem. The following section formally defines the problem, while Section 4 analyzes consensus in three different cases. First the most studied case of integrator systems with coupled dynamics, then with arbitrary linear dynamics, and finally for networks where all agents have identical dynamical behavior. The paper is

E-mail addresses: parada-contzen@control.tu-berlin.de, miguelparadacontzen@gmail.com

http://dx.doi.org/10.1016/j.ejcon.2017.08.004
0947-3580/

then complemented with numeric examples that show some of the main characteristics of the proposed approach.

The identity and zero matrices are respectively denoted $\mathbf{I}$ and $\mathbf{0}$. A matrix composed by N identity matrices stacked in a column is denoted $\mathbf{1} = \mathrm{col}\{\mathbf{I}\}_{i=1}^{N}$. If necessary, the dimensions of these matrices will be denoted as an index. For example, $\mathbf{I}_q$ is the identity matrix in $\mathbb{R}^{q\times q}$ and $\mathbf{0}_{m\times n}$ the zero matrix in $\mathbb{R}^{m\times n}$.

2. Graph theory

Most of the consensus works use intensely graph theoretical methods for description and analysis of networks. In this section, basic notions of the subject are presented based on the quoted works and specialized books as [4,6,7].

An *undirected graph* is a tuple $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} = \{1, 2, \dots N\}$ is a set of N *nodes* or *vertices*, and $\mathscr{E} \subseteq \{(i, j) \in \mathscr{V} \times \mathscr{V}\}$ is a set of *edges*.

We interpret that the edge denoted $(i, j) \in \mathscr{E}$ is the same as the edge $(j, i) \in \mathscr{E}$. This is a slightly abuse of notation as we represent an unordered edge by an ordered pair (i, j). With this notation we mean that an unordered edge between nodes i and j of an undirected graph can be equivalently specified either by the pair (i, j) or the pair (j, i). Which is not the same as the graph having two different ordered edges.

In the context of this paper, the nodes correspond to *agents*, and the existence of an edge means that two agents interact with each other either through input and output signals, or by a "hierarchical relationship" (see Section 3.2). An arbitrary indexation of the edges, which is independent of the labeling of the nodes, can be introduced so that $e_k = (i_k, j_k) \in \mathscr{E} = \{e_1, e_2, \dots, e_{|\mathscr{E}|}\}$. In this case, $i_k \in \mathscr{V}$ and $j_k \in \mathscr{V}$ represent the two nodes associated with the kth edge, $e_k \in \mathscr{E}$.

We focus on *loopless* graphs, *i.e.* graphs where $(i, i) \notin \mathscr{E}$, $\forall i \in \mathscr{V}$. A *path* is an ordered sequence of nodes in an undirected graph such that any pair of consecutive nodes is connected by an edge. An undirected graph is *connected* if there is a path between every two nodes and unconnected otherwise. A (spanning) *tree* $\mathscr{T}$ is an undirected graph over a set of nodes $\mathscr{V}$ that is connected and has $N-1$ edges, where $N = |\mathscr{V}|$ is the number of nodes.

An *undirected weighted graph* is a tuple $\mathscr{G}_w = (\mathscr{G}, w_q)$ where $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is an undirected graph, and $w_q : \mathscr{E} \to \mathscr{M} \subseteq \mathbb{R}^{q\times q}\backslash\{\mathbf{0}\}$ is a function that associates a non-zero positive definite *weight* matrix to each edge.

This last definition is a generalization of the usual one because we introduce matrix weights. This consideration is done to model multiple input/output signals of the agents. *e.g.* three-dimensional position or speed of a vehicle; active and reactive power of an electric generation unit; etc. The *dimension* of a weighted graph is the dimension q of the image matrix space $\mathscr{M}$ of the weight function w_q. An *unweighted* graph is a special case of weighted graphs where $w_q((i, j)) = \mathbf{I}_q$, $\forall (i, j) \in \mathscr{E}$.

A *strictly directed graph*, or strict digraph, is an unweighted graph where the edge set $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ is redefined so that each edge has an unique orientation. That is, $(i, j) \in \mathscr{E} \Rightarrow (j, i) \notin \mathscr{E}$. In this case the notation $(i, j) \in \mathscr{V} \times \mathscr{V}$ and $(j, i) \in \mathscr{V} \times \mathscr{V}$ represent two different edges that cannot be simultaneously part of a strict digraph. An arbitrary strict digraph generated by giving orientations to the edges of an undirected unweighted graph $\mathscr{G}$, will be denoted by $\mathscr{G}^o$.

Because of the inclusion of matrix weights, the usual definitions of graph related matrices also needs to be generalized. The *Incidence Matrix*, denoted $D(\mathscr{G}^o)$, of a strict digraph $\mathscr{G}^o$ of $\dim\{\mathscr{G}^o\} = q$ is defined as a matrix where each block $o_{ik} = [D(\mathscr{G}^o)]_{ik}$ takes either the value $o_{ik} = -\mathbf{I}_q$ if the edge e_k has its origin in i, $o_{ik} = \mathbf{I}_q$ if node i is the destination of edge e_k or $o_{ik} = \mathbf{0}_{q\times q}$ otherwise. Note that this definition assumes that the edges are labeled by the index k. Different labeling systems would lead to different incidence matrices.

The *adjacency matrix*, denoted $A(\mathscr{G}_w)$, of a weighted graph $\mathscr{G}_w$ is constructed so that each block $\mathbf{W}_{ji} := [A(\mathscr{G}_w)]_{ij}$ takes the value $\mathbf{W}_{ji} = w_q((j, i)) \in \mathscr{M}$ if $(j, i) \in \mathscr{E}$ or $\mathbf{W}_{ji} = \mathbf{0}$ otherwise. Note that this matrix is symmetric.

The matrix *degree of node i*, Δ_i, in an undirected loopless weighted graph is defined as the sum of all elements of the respective block column or block row of the adjacency matrix. *i.e.* $\Delta_i = \sum_{j=1}^{N} \mathbf{W}_{ij} = \sum_{i=1}^{N} \mathbf{W}_{ij}$. The *degree matrix* is $\Delta(\mathscr{G}_w) = \mathrm{diag}\{\Delta_1, \dots, \Delta_N\}$.

The *Laplacian matrix* of an undirected loopless weighted graph $\mathscr{G}_w$ is $\hat{L}(\mathscr{G}_w) := \Delta(\mathscr{G}_w) - A(\mathscr{G}_w)$. Each column and row of $\hat{L}(\mathscr{G}_w)$ sums up to zero. This can respectively be written as $\mathbf{1}'\hat{L}(\mathscr{G}_w) = \mathbf{0}$ and $\hat{L}(\mathscr{G}_w)\mathbf{1} = \mathbf{0}$. It can be shown that for weighted graphs, $\hat{L}(\mathscr{G}_w) = D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o)$, where $\mathbf{W} = \mathrm{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} = \mathrm{diag}\{w_q(e_k)\}_{k=1}^{|\mathscr{E}|}$ and $\mathscr{G}^o$ is an arbitrary strict digraph defined from $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. From this property is immediate that the Laplacian matrix is positive semidefinite.

Furthermore, it can also be shown that $\mathrm{rank}\{\hat{L}(\mathscr{G}_w)\} = (N-1)q$ if and only if the loopless undirected weighted graph $\mathscr{G}_w$ is connected. From here, if the eigenvalues of the Laplacian matrix of an undirected weighted graph are ordered in an increasingly, it is clear that the first q of them are identically zero. The algebraic connectivity, $a(\mathscr{G}_w) := \lambda_{q+1}$, is the $(q+1)$-th element of the increasingly ordered set $\mathrm{eig}\{\hat{L}(\mathscr{G}_w)\}$. If it is zero, then the graph is not connected.

3. The consensus problem

3.1. Multiagents systems

Even though consensus based control is formulated for Multi Agents Systems, it is not easy to find a general description of such a system in the related works. Typically, the plant over which control is performed is considered to be a set of simple integrators in continuous time. This is a very particular case and therefore it can be difficult to generalized it to other relevant configurations.

In a realistic scenario in a control theoretical framework, the different components of the network can be defined according to their physical characteristics or functions. The *agents* correspond to the controlled machines, with possible non linear dynamics in continuous or discrete time or discrete states. For example, electric generators in a grid or mobile vehicles. For simplicity, we will consider that each of the N agents in a set $\mathscr{V} = \{1, 2, \dots, N\}$ are described by linear continuous dynamics. Not making the linearity assumption would lead to complications that are not due to the multi agent plant or the consensus problem, but due to the modeling of the components.

Definition 3.1. A linear *autonomous agent* (AA) is an agent $i \in \mathscr{V}$ with individual dynamics given by:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{C}_i\mathbf{x}_i\end{aligned} \tag{1}$$

Where $\mathbf{A}_i \in \mathbb{R}^{n_i\times n_i}$, $\mathbf{B}_i \in \mathbb{R}^{n_i\times q}$, and $\mathbf{C}_i \in \mathbb{R}^{q\times n_i}$.

Note that the number of outputs does not depend on the agent but is always q. We assume that $\mathbf{C}_i\mathbf{B}_i \neq \mathbf{0}$ is full rank. We also assume that each agent has the same number of outputs as inputs. The aggregation of all these components defines a *network* of agents.

Definition 3.2. An *autonomous agents network* (AAN) is the aggregation of all N autonomous agents in a set $\mathscr{V}$. The dynamics of such a network are described by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$