Automatica 93 (2018) 83–91

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Consensus control for linear systems with optimal energy cost ☆

Han Zhang *, Xiaoming Hu

Department of Mathematics, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 22 November 2016
Received in revised form 28 November 2017
Accepted 6 March 2018

Keywords:
Consensus control
Multi-agent systems
Optimal control
Semi-definite programming
Distributed optimization

ABSTRACT

In this paper, we design an optimal energy cost controller for linear systems asymptotic consensus given the topology of the graph. The controller depends only on relative information of the agents. Since finding the control gain for such controller is hard, we focus on finding an optimal controller among a classical family of controllers which is based on Algebraic Riccati Equation (ARE) and guarantees asymptotic consensus. Through analysis, we find that the energy cost is bounded by an interval and hence we minimize the upper bound. In order to do that, there are two classes of variables that need to be optimized: the control gain and the edge weights of the graph and are hence designed from two perspectives. A suboptimal control gain is obtained by choosing $Q = 0$ in the ARE. Negative edge weights are allowed, and the problem is formulated as a Semi-definite Programming (SDP) problem. Having negative edge weights means that "competitions" between the agents are allowed. The motivation behind this setting is to have a better system performance. We provide a different proof compared to Thunberg and Hu (2016) from the angle of optimization and show that the lowest control energy cost is reached when the graph is complete and with equal edge weights. Furthermore, two sufficient conditions for the existence of negative optimal edge weights realization are given. In addition, we provide a distributed way of solving the SDP problem when the graph topology is regular.

1. Introduction

Consensus is an important topic in the research of cooperative control for multi-agent. The goal is to let the states or the outputs of all agents become the same by control laws that depend on the information of the agent and its neighbours. In this paper, we consider the case of asymptotic consensus for linear systems. The goal of this paper is to design distributed controllers using only the relative information between the agents and with minimal control energy cost such that all the system states will eventually become the same as time goes infinity.

In this paper, we focus on designing the optimal energy cost controller for the linear systems so that they can reach asymptotic consensus. It is well-known that the asymptotic consensus for linear systems is equivalent to regulating $N-1$ systems with the dynamics $\dot{x}_i = Ax_i + \lambda_i Bu_i$, $2 \leq i \leq N$, where λ_i is the ith smallest eigenvalue of the Laplacian matrix, see Fax and Murray (2004) and Zhang, Lewis, and Das (2011). The area is well-studied, for example, in Borrelli and Keviczky (2008), the authors design linear quadratic regulators for identical linear systems when the graph topology is given. Though it is quite similar to what we undertake here, but as mentioned above, the equivalent problem is to regulate $N-1$ different systems with minimum energy, and hence is not the same. The authors of Rogge, Suykens, and Aeyels (2010) consider the problem as a quadratic optimal control problem on a ring network while we consider the case of a graph with arbitrary topologies. Augmented Lagrangian approach is used in Lin, Fardad, and Jovanovic (2011) to design a structured distributed controller so that the H_2 norm of the noisy systems is minimized. Deshpande, Menon, Edwards, and Postlethwaite (2011) consider a similar problem and use a two-step approach to design the control law. But their controller does not only use relative information of the agents, but also use the agents' own states. Cao and Ren (2010) study the optimal consensus of the single-integrators for both discrete-time and continuous-time cases. In Lin, Fardad, and Jovanović (2013), alternating direction method of multipliers (ADMM) is used to minimize the H_2 norm so that an optimal sparse feedback gain is obtained. In Thunberg and Hu (2016), a "topology free" control energy minimization problem is considered and the distributed energy-optimal control corresponds to a complete graph with equal edge weights.

On the other hand, distributed optimization has attracted great attention these years due to the wide applications in the network. Compared to the abundant results on distributed optimization in

☆ This work is supported by China Scholarship Council. The material in this paper was partially presented at the 36th Chinese Control Conference, July 26–28, 2017, Dalian, China. This paper was recommended for publication in revised form by Associate Editor Bert Tanner under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: hanzhang@kth.se (H. Zhang), hu@kth.se (X. Hu).

https://doi.org/10.1016/j.automatica.2018.03.044
0005-1098/© 2018 Elsevier Ltd. All rights reserved.

real vector spaces, for instances, Annergren, Pakazad, Hansson, and Wahlberg (2014), Lou, Shi, Johansson, and Hong (2014), Nedic and Ozdaglar (2009), Nedic, Ozdaglar, and Parrilo (2010) and Yi, Hong, and Liu (2015), the results on distributed SDP still remain limited. Dall'Anese, Zhu, and Giannakis (2013) considered an optimal power flow problem in power grids and relax it into an SDP problem. Then ADMM is used to solve the problem in a distributed manner. In Simonetto and Leus (2014), a sensor localization problem is considered. Relaxation towards an SDP and ADMM is used to solve the problem distributedly as well. Pakazad, Hansson, Andersen, and Rantzer (2014) analyse the robustness of interconnected uncertain systems and linear matrix inequalities (LMIs) are reformulated in SDP. Chordal sparsity structure is assumed among the data matrices of the SDP and hence the problem is decomposed and solved distributedly using proximal splitting method. In Pakazad, Hansson, Andersen, and Rantzer (2015), coupled SDPs with tree structures are considered. A distributed primal–dual interior point method is proposed to solve the coupled SDPs. The aforementioned work all utilize the idea of "decomposition" somehow but in this paper we treat the SDP in a different manner: reaching an optimal consensus in the intersection of convex feasible sets. Also, what makes our work different from existing distributed optimization problem is that our problem motivates from optimizing a parameter of a graph and the communication network of the distributed optimization algorithm is actually the physical network itself, while most distributed optimization algorithms relax and decompose the original problem and "design" the communication network according to the structure of the decomposed problem, see Pakazad et al. (2015) as an example.

The main contribution of this paper is the construction of an optimal energy controller that depends only on the relative information between the agents. The controller has two classes of variables that need to be determined: the control gain and the edge weights of the graph. Similar to Borrelli and Keviczky (2008), computing the optimal control gain for the controller is hard, thus we focus on finding an optimal controller among a classical family of controller designs based on ARE and guarantees asymptotic consensus. Through analysis, we found that the energy cost is bounded by an interval and hence we minimize the upper bound. A suboptimal control gain is obtained by choosing $Q = 0$ in the ARE; the edge weights of the graph is optimized by solving an underlying SDP. The controller that we designed enjoy several favourable properties:

(1) The controller coincides with the optimal control in Thunberg and Hu (2016) when the graph is complete. It has been pointed out in Thunberg and Hu (2016) that any other distributed control laws constructed by Laplacian matrices that do not correspond to complete graphs with equal edge weights are suboptimal.
(2) When optimizing the edge weights, "competitions" are allowed between the connected agents. By doing so, the feasible region of the optimization problem is enlarged, and hence a smaller control energy cost might be obtained. We offer two sufficient conditions for when will "competitions" happen between agents. These two conditions help to determine whether the two agents will compete if we add a connection between them based on the old optimal solution.
(3) When the graph topology is regular, namely, every node has the same number of neighbours, the controller can be calculated in a distributed manner.

The rest of this paper is organized as follows. In Section 2, some preliminaries and notations are presented. In Section 3 we introduce the problem formulation. The design of the control gain is presented in Sections 4.1 and 4.2. In Section 4.3, the edge weights design is formulated as an SDP problem and we provide a different proof compared to Thunberg and Hu (2016) from the angle of optimization and show that the lowest control energy cost is reached when the graph is complete and with equal edge weights. Two sufficient conditions on the existence of the negative optimal edge weight are presented. In Section 4.4, a distributed way of computing the optimal edge weights is presented when the graph is regular. Finally, we conclude the paper and describe some future work in Section 5.

2. Preliminaries

We denote $\mathbf{1}$ as an N dimensional all-one column vector. $\mathbf{0}$ is denoted as an N dimensional all-zero matrix. The element on the ith row and jth column of any matrix D is expressed as $[D]_{ij}$. $D_1 \succeq D_2$ and $G_1 \preceq G_2$ mean that $D_1 - D_2$ and $G_2 - G_1$ are positive semi-definite. $\otimes$ denotes the Kronecker product. $\|\cdot\|$ denotes 2-norm of matrices or vectors. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We use $|\cdot|$ to denote the number of the elements of a set. And any notation with the superscript $*$ denotes the optimal solution to the corresponding optimization problem. $tr(\cdot)$ denotes the trace of a matrix. If $D, G \in \boldsymbol{S}_+^n$ are positive definite matrices, then $tr(DG)$ is the inner-product between D and G.

An edge-weighted undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ is composed of a node set $\mathcal{V} = \{1, 2, \ldots, N\}$, an edge set $(i, j) \in \mathcal{E},\ i, j \in \mathcal{V}$ which describes the connection topology between the nodes and the edge weight set $w_{ij} \in \mathcal{W},\ i, j \in \mathcal{V}$ which includes all the weights of the corresponding edges. To abbreviate the notation, we label the edges with numbers. For example, an edge with label l is denoted as $l \in \mathcal{E}$. On the other hand, seen from the nodes' perspective, the set of edges that is connected to node i is denoted as $\mathcal{E}(i)$, which can be interpreted as communication channels of node i. Note that $\mathcal{E} = \bigcup_{i\in\mathcal{V}}\mathcal{E}(i)$. Similarly, the set of the edge weights belong to node i is denoted as $\mathcal{W}(i)$. $\mathcal{N}(i)$ denotes the neighbour vertices set of node i.

Note that in this paper, we consider undirected graphs, hence the edge-weighted Laplacian matrix L_w is symmetric and defined as

$$[L_w]_{ij} = \begin{cases} \sum_l w_{il} & \text{if } i = j \text{ and } (i, l) \in \mathcal{E} \\ -w_{ij} & \text{if } i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$
$$\Leftrightarrow L_w = \sum_{k\in\mathcal{E}} w_k E_k,$$

where k is the label of the edges, $w_k \in \mathcal{W},\ \forall k \in \mathcal{E}$ are the edge weights. If node i and j are connected via edge k, then $[E_k]_{ii} = [E_k]_{jj} = 1, [E_k]_{ij} = [E_k]_{ji} = -1$, and the other elements of E_k are zero. For a connected graph, the eigenvalues of L_w is denoted as $0 = \lambda_1 < \lambda_2 \le \cdots \le \lambda_N$.

For any symmetric matrix $G \in \boldsymbol{S}^n$, $svec(G)$ is defined as $svec(G) = [G_{11}, \sqrt{2}G_{21}, \ldots, \sqrt{2}G_{n1}, G_{22}, \sqrt{2}G_{32}, \ldots, \sqrt{2}G_{n1}, \ldots, G_{nn}]^T$. It follows from the above definition that $tr(DG) = svec(D)^T svec(G)$, $\forall D, G \in \boldsymbol{S}^n$.

The *Symmetric Kronecker Product* between two matrices G and D is defined by the following identity

$$(R_1 \otimes_s R_2) svec(G) = \frac{1}{2} svec(R_2 G R_1^T + R_1 G R_2^T),$$

where $G \in \boldsymbol{S}^n$, but R_1 and R_2 is not necessarily symmetric. For more details about $svec(\cdot)$ and Symmetric Kronecker Product can be found in Alizadeh, Haeberly, and Overton (1998) and Schäcke (2013).