Automatica 80 (2017) 243–252

Contents lists available at ScienceDirect

Automatica

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

Distributed methods for synchronization of orthogonal matrices over graphs[☆]

Johan Thunberg [a], Florian Bernard [a,b], Jorge Goncalves [a,c]

[a] Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg

[b] Centre Hospitalier de Luxembourg, Luxembourg City, Luxembourg

[c] Control Group, Department of Engineering, University of Cambridge, Cambridge, United Kingdom

ARTICLE INFO

Article history:
Received 23 October 2015
Received in revised form
24 January 2017
Accepted 25 January 2017
Available online 28 March 2017

Keywords:
Multi-agent systems
Distributed optimization
Sensor networks
Consensus algorithms
Robust estimation
Measurement and instrumentation

ABSTRACT

This paper addresses the problem of synchronizing orthogonal matrices over directed graphs. For synchronized transformations (or matrices), composite transformations over loops equal the identity. We formulate the synchronization problem as a least-squares optimization problem with nonlinear constraints. The synchronization problem appears as one of the key components in applications ranging from 3D-localization to image registration. The main contributions of this work can be summarized as the introduction of two novel algorithms; one for symmetric graphs and one for graphs that are possibly asymmetric. Under general conditions, the former has guaranteed convergence to the solution of a spectral relaxation to the synchronization problem. The latter is stable for small step sizes when the graph is quasi-strongly connected. The proposed methods are verified in numerical simulations.

1. Introduction

This paper introduces two new distributed algorithms for the problem of synchronizing orthogonal matrices over graphs. Synchronization means that compositions of transformations (multiplications of matrices) over loops in the graph equal the identity (matrix) (Bandeira, Singer, & Spielman, 2013; Bernard, Thunberg, Gemmar et al., 2015; Singer, 2011; Wang & Singer, 2013). Thus, "synchronization" does not refer to the related concepts of consensus (Olfati-Saber & Murray, 2004a) or rendezvous, e.g., attitude synchronization (Thunberg, Song, Montijano, Hong, & Hu, 2014). We formulate the problem as a nonlinear least-squares optimization with matrix variables (Absil, Mahony, & Sepulchre, 2009; Helmke & Moore, 2012). For symmetric communication topologies we provide an algorithm with strong convergence guarantees — the solution converges to the optimal solution of a spectral relaxation, which in turn is known to produce near-optimal solutions. For graphs that are possibly asymmetric we provide an algorithm with weaker convergence guarantees but with good performance in numerical simulations.

The synchronization problem appears as one of the key components in the following applications: the 3D-localization problem, where the transformations are obtained from camera measurements; the generalized Procrustes problem, where scales, rotations, and translations are calculated between multiple point clouds (Gower & Dijksterhuis, 2004); the image registration problem, where transformations are calculated between multiple images (Bernard, Thunberg, Husch et al., 2015). There are also many other interesting applications for the synchronization problem, see Section 1.2 in Boumal (2015). Due to sensor and communication limitations, there is often a need to use distributed protocols for the 3D-localization problem and several approaches have been proposed recently (Aragues, Sagues, & Mezouar, 2015; Montijano, Zhou, Schwager, & Sagues, 2014; Tron & Vidal, 2014).

If we exclude the requirement that the synchronization method shall be distributed, there is an extensive body of work. Govindu et al. have presented several approaches based on Lie-group averaging, where a first-order approximation in the tangent space is used (Govindu, 2004, 2006; Govindu & Pooja, 2014). Singer et al. have presented several optimization approaches (Bandeira et al., 2013; Chaudhury, Khoo, & Singer, 2013; Hadani & Singer, 2011a,b; Singer, 2011; Singer & Shkolnisky, 2011; Wang & Singer, 2013).

☆ The authors gratefully acknowledge the financial support from the Fonds National de la Recherche, Luxembourg (FNR8864515, FNR6538106). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Michael M. Zavlanos under the direction of Editor Christos G. Cassandras.

E-mail addresses: johan.thunberg@uni.lu (J. Thunberg), florian.bernard@uni.lu (F. Bernard), jorge.goncalves@uni.lu (J. Goncalves).

http://dx.doi.org/10.1016/j.automatica.2017.02.025
0005-1098/

Pachauri et al. have addressed the special case where the matrices are permutation matrices (Pachauri, Kondor, & Singh, 2013). In Wang and Singer (2013), three types of relaxations of the problem are presented: semidefinite programming relaxation (see Boumal, 2015 for an extensive analysis of this approach); spectral relaxation; least unsquared deviation in combination with semidefinite relaxation. These three relaxations were evaluated in the probabilistic framework where the error to the ground truth was calculated in numerical experiments. The simulations showed that the first two approaches were on par, whereas the last approach performed slightly better. Furthermore, the last approach was significantly more robust to outliers. The first distributed algorithm we present has a connection to the second category of the three relaxations above, since the matrices in the algorithm converge to the optimal solution of the spectral relaxation. Our methods are extrinsic, in the sense that the matrices are calculated in $\mathbb{R}^{d\times d}$ and then projected onto the set of orthogonal matrices. The opposite to extrinsic methods are intrinsic methods where no projections from an ambient space occur. In Afsari, Tron, and Vidal (2013), intrinsic gradient descent methods are studied for the problem of finding the Riemannian center of mass.

The contributions of this work can be summarized as the introduction of two novel algorithms (Algorithms 1 and 2) for *distributed* synchronization of orthogonal matrices over *directed* graphs. For both algorithms we provide conditions for guaranteed convergence. The main result of the paper is the above-mentioned convergence in Algorithm 1 to the optimal solution of the spectral relaxation problem (Proposition 14). Previous works in the context of distributed algorithms have focused on undirected graphs and 3D rotations (Aragues et al., 2015; Montijano et al., 2014; Tron & Vidal, 2014). However, in this work we consider directed graphs and arbitrary dimensions. It should be noted that some of the existing algorithms can be extended to higher dimensions and are given for the 3D-case mostly for clarity of exposition.

The distributed approaches in this work bear a resemblance to linear consensus protocols (Jadbabaie & Morse, 2003; Mesbahi & Egerstedt, 2010; Olfati-Saber, Fax, & Murray, 2007; Olfati-Saber & Murray, 2004b). The methods also share similarities with the eigenvector method in Howard, Cochran, Moran, and Cohen (2010) and gossip algorithms (Boyd, Ghosh, Prabhakar, & Shah, 2006). The important states in our algorithms are matrices, and those combined converge to a tall matrix whose range space is a certain linear subspace. In the case of symmetric communication between agents, the proposed method can either be interpreted as an extension of the power method or the steepest descent method. In our methods, instead of using the graph Laplacian matrix (Mesbahi & Egerstedt, 2010), matrices similar to the graph connection Laplacian matrix (Singer & Wu, 2012) are used. These matrices can be seen as a generalizations of the graph Laplacian matrix, in which the scalars are replaced by matrix blocks.

The paper proceeds as follows. In Section 2 we introduce the definitions that are necessary in order to precisely state the problem, which is done in Section 3. Subsequently, the distributed method for the case of symmetric graphs (Algorithm 1) is introduced and analyzed in Section 4. In Section 5, the distributed method for the case of directed and possibly asymmetric graphs (Algorithm 2) is introduced and analyzed. In Section 6, the paper is concluded. For those propositions and lemmas that appear without proofs (if nothing else is mentioned), the proofs are found in an extended version of this paper available at arXiv (Thunberg, Bernard, & Goncalves, 2017).

2. Preliminaries

2.1. Directed graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. Throughout the paper, the notation $\mathcal{A} \subset \mathcal{B}$ means that every element in $\mathcal{A}$ is contained in $\mathcal{B}$. The set $\mathcal{N}_i$ is the set of neighboring nodes of node i and defined by $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$. The adjacency matrix $A = [A_{ij}]$ for the graph $\mathcal{G}$ is defined by $A_{ij} = 1$ if $(i,j) \in \mathcal{E}$ and $A_{ij} = 0$ if $(i,j) \notin \mathcal{E}$. The graph Laplacian matrix is defined by $L = \mathrm{diag}(A1_n) - A$, where $1_n \in \mathbb{R}^n$ is a vector with all entries equal to 1. In order to emphasize that the adjacency matrix A, the graph Laplacian matrix L and the $\mathcal{N}_i$ sets depend on the graph $\mathcal{G}$, we may write $A(\mathcal{G})$, $L(\mathcal{G})$ and $\mathcal{N}_i(\mathcal{G})$ respectively. For simplicity however, we mostly omit this notation and simply write A, L, and $\mathcal{N}_i$.

Definition 1 (*Connected Graph, Undirected Path*)**.** The directed graph $\mathcal{G}$ is connected if there is an undirected path from any node in the graph to any other node. An undirected path is defined as a (finite) sequence of unique nodes such that for any pair (i,j) of consecutive nodes in the sequence it holds that $((i,j) \in \mathcal{E})$ or $((j,i) \in \mathcal{E})$.

Definition 2 (*Quasi-strongly Connected Graph, Center, Directed Path*)**.** The directed graph $\mathcal{G}$ is quasi-strongly connected (QSC) if it contains a center. A center is a node in the graph to which there is a directed path from any other node in the graph. A directed path is defined as a (finite) sequence of unique nodes such that any pair of consecutive nodes in the sequence comprises an edge in $\mathcal{E}$.

Definition 3 (*Strongly Connected Graph*)**.** The directed graph $\mathcal{G}$ is strongly connected if for all pairs of nodes $(i,j) \in \mathcal{V} \times \mathcal{V}$, there is a directed path from i to j.

Definition 4 (*Symmetric Graph*)**.** The directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is symmetric if $((i,j) \in \mathcal{E}) \Rightarrow ((j,i) \in \mathcal{E})$ for all $(i,j) \in \mathcal{V} \times \mathcal{V}$.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the graph $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ is the graph constructed by reversing the direction of the edges in $\mathcal{E}$, i.e., $(i,j) \in \bar{\mathcal{E}}$ if and only if $(j,i) \in \mathcal{E}$. It is easy to see that $A(\bar{\mathcal{G}}) = (A(\mathcal{G}))^T$ and $L(\bar{\mathcal{G}}) = \mathrm{diag}((A(\mathcal{G}))^T 1_n) - A(\mathcal{G})^T$.

2.2. Synchronization or transitive consistency of matrices

The set of invertible matrices in $\mathbb{R}^{d\times d}$ is $GL(d,\mathbb{R})$ and the group of orthogonal matrices in $\mathbb{R}^{d\times d}$ is $O(d) = \{R \in \mathbb{R}^{d\times d} : R^T R = I_d\}$. The set $SO(d)$ comprises those matrices in $O(d)$ whose determinants are equal to 1.

Definition 5 (*Transitive Consistency*)**.**

(1) The matrices in the collection $\{R_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ are transitively consistent for the complete graph if $R_{ik} = R_{ij}R_{jk}$ for all i, j and k.
(2) Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the matrices in the collection $\{R_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ are *transitively consistent for* $\mathcal{G}$ if there is a collection $\{R_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{R_{ij}\}_{(i,j)\in\mathcal{E}}$ such that $\{R_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ is transitively consistent for the complete graph.

If it is apparent by the context, sometimes we will be less strict and omit to mention which graph a collection of transformations is transitively consistent for. Another word for transitive consistency is synchronization. We will use the two interchangeably. A sufficient condition for synchronization of the R_{ij}-matrices for any graph is that there is a collection $\{R_i\}_{i\in\mathcal{V}}$ of matrices in $GL(d,\mathbb{R})$ such that

$$R_{ij} = R_i^{-1}R_j \quad (1)$$

for all $(i,j) \in \mathcal{E}$. Lemma 7 below provides additional important information. The result is similar to that in Tron and Vidal (2014). For the statement of the lemma, the following definition is needed.