22nd International Symposium on
Mathematical Theory of Networks and Systems
July 11-15, 2016. Minneapolis, MN, USA

Clustering-Based Model Order Reduction for Multi-Agent Systems with General Linear Time-Invariant Agents

Petar Mlinarić Sara Grundel Peter Benner

***Abstract*— In this paper, we extend our clustering-based model order reduction method for multi-agent systems with single-integrator agents to the case where the agents have identical general linear time-invariant dynamics. The method consists of the Iterative Rational Krylov Algorithm, for finding a good reduced order model, and the QR decomposition-based clustering algorithm, to achieve structure preservation by clustering agents. Compared to the case of single-integrator agents, we modified the QR decomposition with column pivoting inside the clustering algorithm to take into account the block-column structure. We illustrate the method on small and large-scale examples.**

I. Introduction

The study of consensus and synchronization for multi-agent systems has received considerable attention in recent years [1], [2], [3]. In brief, multi-agent systems are network systems that can consist of a very large number of simple and identical subsystems, called agents. This motivates research on clustering-based model order reduction (MOR) methods that would reduce the large network, simplifying analysis, simulation, and control, while preserving consensus and synchronization properties.

We have developed a clustering-based MOR method for multi-agent systems with single-integrator agents [4]. Here, we generalize this method to multi-agent systems where agents have identical, but general, linear time-invariant (LTI) dynamics.

There are several published papers related to the work presented here. The paper [5] extends the clustering-based MOR method based on θ-reducible clusters from [6] to networks of second-order subsystems, but not more general subsystems. The controller-Hessenberg form is the basis of the extended method and the $\mathcal{H}_\infty$-error bound. The authors of [7] propose a clustering method for networks of identical passive subsystems, although it is limited to networks with a tree structure. The reference [8] extends the expression for the clustering-based $\mathcal{H}_2$-error from [9] to a class of second-order physical network systems, when almost equitable partitions are used.

The outline of this paper is as follows. In Section II we introduce the necessary background. We explain the more general clustering method in Section III and demonstrate it on a few examples in Section IV. We conclude with Section V.

P. Mlinarić, S. Grundel, and P. Benner are with Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. E-mails: mlinaric@mpi-magdeburg.mpg.de, grundel@mpi-magdeburg.mpg.de, benner@mpi-magdeburg.mpg.de.

P. Mlinarić is also affiliated to the "International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg)".

II. Preliminaries

A. Multi-Agent Systems

We define a *multi-agent system* over an undirected, weighted, connected graph $G = (V_G, E_G, A_G)$ with the set of vertices $V_G = \{1, 2, \ldots, n_G\}$, the set of edges E_G and the adjacency matrix $A_G = [a_{ij}] \in \mathbb{R}^{n_G \times n_G}$. First, in every vertex of the graph we define an agent

$$\begin{aligned}\dot{x}_i(t) &= Ax_i(t) + Bz_i(t),\\ y_i(t) &= Cx_i(t),\end{aligned}$$

with its state $x_i(t) \in \mathbb{R}^{n_A}$, input $z_i(t) \in \mathbb{R}^{m_A}$, and output $y_i(t) \in \mathbb{R}^{p_A}$, for $i \in V_G$. A, B, and C are real matrices of appropriate sizes and identical for all agents, but they can be arbitrary (later, we will constrain this choice to guarantee the stability or synchronization of the multi-agent system).

Second, we define the inputs of individual agents, consisting of a coupling term and an external input to some agents called *leaders*. Let $V_L = \{v_1, v_2, \ldots, v_{m_G}\} \subseteq V_G$ be the set of leaders. Then, we define the input of the ith agent as

$$z_i(t) := \begin{cases} K \sum_{j=1}^{n_G} a_{ij}(y_j(t) - y_i(t)) + u_k(t), & \text{if } i = v_k,\\ K \sum_{j=1}^{n_G} a_{ij}(y_j(t) - y_i(t)), & \text{otherwise,}\end{cases}$$

where $K \in \mathbb{R}^{m_A \times p_A}$ and $u_k(t) \in \mathbb{R}^{m_A}$ is the kth external input, for $k \in \{1, 2, \ldots, m_G\}$. For simplicity, we assume that $p_A = m_A$ and $K = I_{m_A}$.

We find that the dynamics of the multi-agent system is

$$\dot{x}(t) = (I_{n_G} \otimes A - L \otimes BC)x(t) + (M \otimes B)u(t), \quad (1)$$

with state

$$x(t) = \begin{bmatrix} x_1(t)\\ x_2(t)\\ \vdots\\ x_{n_G}(t)\end{bmatrix} \in \mathbb{R}^{n_G n_A}$$

and input

$$u(t) = \begin{bmatrix} u_1(t)\\ u_2(t)\\ \vdots\\ u_{m_G}(t)\end{bmatrix} \in \mathbb{R}^{m_G m_A},$$

where $L \in \mathbb{R}^{n_G \times n_G}$ is the Laplacian matrix of the graph G defined by

$$[L]_{ij} := \begin{cases} \sum_{k=1}^{n_G} a_{ik}, & \text{if } i = j, \\ -a_{ij}, & \text{if } i \neq j, \end{cases}$$

and $M = \begin{bmatrix} e_{v_1} & e_{v_2} & \cdots & e_{v_{m_G}} \end{bmatrix} \in \mathbb{R}^{n_G \times m_G}$. For the output of the multi-agent system

$$y(t) = C_{\text{MAS}} x(t), \tag{2}$$

we consider two possibilities. The first is

$$C_{\text{MAS}} = W^{\frac{1}{2}} R^T \otimes I_{n_A}, \tag{3}$$

where $R \in \mathbb{R}^{n_G \times p_G}$ and $W \in \mathbb{R}^{p_G \times p_G}$ are the incidence and edge weights matrices of the graph G and p_G is the number of edges of the graph G. As in [9], the output in this case is a vector of weighted differences of agents' states across the edges. The second is

$$C_{\text{MAS}} = \begin{bmatrix} e_{h_1} & e_{h_2} & \cdots & e_{h_{p_G}} \end{bmatrix}^T \otimes I_{n_A}, \tag{4}$$

where $h_1, h_2, \ldots, h_{p_G} \in V_G$. Here, the output is a vector of a few agents' states. Other possibilities for the output include replacing I_{n_A} in (3) or (4) with C, such that the output depends of the agents' outputs, and not states.

B. Model Order Reduction via Projection

Petrov-Galerkin projection is a general framework for MOR techniques. Numerous methods, including balanced truncation and moment matching (see [10] for an overview), belong to the class of Petrov-Galerkin projection methods. We briefly introduce this framework here.

Let

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{5}$$

be an arbitrary LTI system of order n, where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$. Then, Petrov-Galerkin projection consists of choosing two full-rank matrices $V_r, W_r \in \mathbb{R}^{n \times r}$, for some $r < n$, and defining a reduced order model (ROM) of order r by

$$\begin{aligned} W_r^T V_r \dot{\widehat{x}}(t) &= W_r^T A V_r \widehat{x}(t) + W_r^T B u(t), \\ \widehat{y}(t) &= C V_r \widehat{x}(t). \end{aligned} \tag{6}$$

Note that multiplying V_r and W_r on the right by nonsingular matrices gives us an equivalent LTI system for the ROM. Therefore, the ROM is defined by $\operatorname{Im} V_r$ and $\operatorname{Im} W_r$, the subspaces generated by the columns of V_r and W_r.

C. Projection-Based Clustering

Let $\pi = \{C_1, C_2, \ldots, C_{r_G}\}$ be a partition of the graph G, i.e. of the vertex set V_G. The *characteristic matrix of the partition* π is the matrix $P(\pi) \in \mathbb{R}^{n_G \times r_G}$ defined by

$$[P(\pi)]_{ij} := \begin{cases} 1, & \text{if } i \in C_j, \\ 0, & \text{otherwise}, \end{cases}$$

for all $i \in V_G$ and $j \in \{1, 2, \ldots, r_G\}$ [9]. Analogously to [9] for single-integrator agents, we define the Petrov-Galerkin projection matrices

$$\begin{aligned} V_r &:= P(\pi) \otimes I_{n_A}, \\ W_r &:= P(\pi) \left(P(\pi)^T P(\pi) \right)^{-1} \otimes I_{n_A}, \end{aligned}$$

which achieve clustering for multi-agent systems with general linear dynamics. To see this, notice that the ROM of the multi-agent system (1), (2) is then

$$\begin{aligned} \dot{\widehat{x}}(t) = {} & \left(I_{n_G} \otimes A - \left(P^T P \right)^{-1} P^T L P \otimes BC \right) \widehat{x}(t) \\ & + \left(\left(P^T P \right)^{-1} P^T M \otimes B \right) u(t), \\ \widehat{y}(t) = {} & C_{\text{MAS}} (P \otimes I_{n_A}) \widehat{x}(t), \end{aligned} \tag{7}$$

where we use a shorter notation $P := P(\pi)$. [9] shows that the matrix

$$\widehat{L} := \left(P^T P \right)^{-1} P^T L P$$

is the Laplacian matrix of a directed, symmetric, connected graph, on which the reduced multi-agent system is defined. In this sense, the network structure is preserved in the ROM.

D. Stability and Synchronization

The paper [11] analyzes the stability and synchronization of systems such as (1). The system (1) is *stable* if the matrix $I_{n_G} \otimes A - L \otimes BC$ is Hurwitz, as is the usual definition. It is shown that the matrix $I_{n_G} \otimes A - L \otimes BC$ is Hurwitz if and only if $A - \lambda BC$ is Hurwitz for every eigenvalue λ of L. In fact, more generally, it is shown that

$$\sigma(I_{n_G} \otimes A - L \otimes BC) = \bigcup_{\lambda \in \sigma(L)} \sigma(A - \lambda BC).$$

The system (1) is *synchronized* if $x_i(t) - x_j(t) \to 0$ as $t \to \infty$, for all $i, j \in V_G$ and for all initial conditions when the input u is zero. This condition is clearly equivalent to the output stability, where the output represents the discrepancies among the agents, such as in (3). The following Lemma (Lemma 4.2 in [11]) gives the necessary and sufficient condition for synchronization.

Lemma 1: Let G be an undirected, weighted, connected graph. Then the system (1) is synchronized if and only if $A - \lambda BC$ is Hurwitz for all positive eigenvalues λ of L.

Here, we are interested in synchronized multi-agent systems and how to preserve synchronization in the ROM using clustering. Using Cauchy's interlacing theorem (as was done in [9]), it can be seen that the matrix $\widehat{L}$ has a simple zero eigenvalue and that the other eigenvalues are positive and lie between the positive eigenvalues of L. We note that Lemma 1 can be extended to graphs with Laplacian matrices having the same properties as $\widehat{L}$, namely the real spectrum and the simple zero eigenvalue with the corresponding right eigenvector of all ones. Therefore, it is necessary and sufficient that $A - \lambda BC$ is Hurwitz for every positive eigenvalue of $\widehat{L}$ for the ROM (7) to be synchronized.

Now we identify a sufficient condition for preserving synchronization, independent of the partition used. Let there be an open interval (α, β), $0 \leqslant \alpha < \beta \leqslant \infty$, such that $A - \lambda BC$ is Hurwitz for all $\lambda \in (\alpha, \beta)$. Then, if (α, β) contains all positive eigenvalues of L, then the original system (1), (2) and all the ROMs (7) are synchronized. In case $\alpha = 0$ and β is finite, we can move all the positive eigenvalues of L inside the interval $(0, \beta)$ by scaling down all the weights in the graph, which is a simple method to ensure synchronization of the original system and the ROMs.

One interesting example of an agent is the undamped oscillator, given by the matrices

$$A = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}.$$

It is easy to check that A is not Hurwitz, but that $A - \lambda BC$ is Hurwitz for all $\lambda > 0$ if $k > 0$, $c_1 \geqslant 0$, and $c_2 > 0$. Therefore, in this case $\alpha = 0$ and $\beta = \infty$.

III. Clustering Method

A. $\mathcal{H}_2$-Optimal Model Order Reduction

In Section II-B, we introduced Petrov-Galerkin projection as a general MOR framework, without describing how to choose good projection matrices V_r and W_r. Here, we formulate the $\mathcal{H}_2$-optimal MOR problem and refer to an efficient method for solving it.

The $\mathcal{H}_2$-norm $\|\cdot\|_{\mathcal{H}_2}$ is defined for any stable, strictly proper transfer function H by

$$\|H\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \|H(i\omega)\|_F^2 \, \mathrm{d}\omega,$$

where $\|\cdot\|_F$ is the Frobenius norm. Let H and $\widehat{H}$ denote the transfer functions of the LTI system (5) and its ROM (6). The $\mathcal{H}_2$-optimal MOR problem is

$$\min_{V_r, W_r \in \mathbb{R}^{n \times r}} \quad \|H - \widehat{H}\|_{\mathcal{H}_2},$$

which is known to be intractable. The Iterative Rational Krylov Algorithm (IRKA) finds a local optimum efficiently, and it often finds the global optimum [12].

As Section II-D indicates, we are interested in synchronized systems, which means that H can have unstable poles (poles in the closed right complex half-plane). Considering the $\mathcal{H}_2$-optimal MOR problem is possible also in this case. Clearly, for $\|H - \widehat{H}\|_{\mathcal{H}_2}$ to be defined, $H - \widehat{H}$ is necessarily stable. This implies that $\widehat{H}$ needs to preserve the unstable poles of H. It follows that

$$\|H - \widehat{H}\|_{\mathcal{H}_2} = \|H_{\text{stab}} - \widehat{H}_{\text{stab}}\|_{\mathcal{H}_2},$$

where H_{stab} and $\widehat{H}_{\text{stab}}$ are the stable parts of H and $\widehat{H}$, obtained by removing the unstable poles from H and $\widehat{H}$. We conclude that the $\mathcal{H}_2$-optimal ROM $\widetilde{H}$ for an unstable system H preserves the unstable poles of H and its stable part $\widetilde{H}_{\text{stab}}$ is an $\mathcal{H}_2$-optimal ROM for H_{stab}.

In particular, the (absolute) $\mathcal{H}_2$-error is defined, but the same is not true for the relative $\mathcal{H}_2$-error, since $\|H\|_{\mathcal{H}_2}$ is not defined. By abuse of terminology, we will refer to

$$\frac{\|H - \widehat{H}\|_{\mathcal{H}_2}}{\|H_{\text{stab}}\|_{\mathcal{H}_2}}$$

as the relative $\mathcal{H}_2$-error, which is actually the relative $\mathcal{H}_2$-error between H_{stab} and $\widehat{H}_{\text{stab}}$. This is an appropriate measure of error, since the $\mathcal{H}_2$-optimal MOR problem for an unstable system reduces to an $\mathcal{H}_2$-optimal MOR problem for the stable part of the system.

B. $\mathcal{H}_2$-Suboptimal Clustering

In [4], we proposed an $\mathcal{H}_2$-suboptimal clustering MOR method for multi-agent systems with single-integrator agents. The method combines IRKA and a QR decomposition-based clustering algorithm introduced in [13].

We apply the clustering algorithm to the Petrov-Galerkin projection matrices obtained from IRKA. The motivation for this comes from the constraint

$$\operatorname{Im} V_r = \operatorname{Im} P(\pi)$$

which the ROM needs to satisfy. An equivalent constraint is

$$V_r = P(\pi) Z,$$

with some nonsingular $Z \in \mathbb{R}^{r_G \times r_G}$. Observing a simple example for $P(\pi)$ with $\pi = \{\{1\}, \{2, 3\}, \{4, 5, 6\}\}$ ($n_G = 6$, $r_G = 3$):

$$P(\pi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

we find that

$$V_r = P(\pi) Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_2 \\ z_3 \\ z_3 \\ z_3 \end{bmatrix},$$

where $z_i \in \mathbb{R}^{1 \times 3}$, $i \in \{1, 2, 3\}$, are the rows of $Z \in \mathbb{R}^{3 \times 3}$. From this, we see that the rows of V_r are equal if and only if the corresponding agents are in the same cluster. This motivates the idea to cluster the rows of V_r obtained from IRKA to find a $\mathcal{H}_2$-suboptimal partition π. Furthermore, the rows of Z are linearly independent, which suggests using QR decomposition with column pivoting on V_r^T. This clustering algorithm was introduced in [13] and is given in Algorithm 1.

Now we try to see if the same reasoning can give us a clustering method for multi-agent systems with general agents. Using the same example as before, with agents of arbitrary order n_A, we have

$$\begin{aligned} V_r &= (P(\pi) \otimes I_{n_A}) Z \\ &= \begin{bmatrix} I_{n_A} & 0 & 0 \\ 0 & I_{n_A} & 0 \\ 0 & I_{n_A} & 0 \\ 0 & 0 & I_{n_A} \\ 0 & 0 & I_{n_A} \\ 0 & 0 & I_{n_A} \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_2 \\ Z_3 \\ Z_3 \\ Z_3 \end{bmatrix}, \end{aligned}$$

Algorithm 1 Clustering using QR decomposition with column pivoting [13], [4]

Input: Matrix $V \in \mathbb{R}^{n \times r}$ of rank r
Output: Partition π such that $\operatorname{Im} P(\pi) \approx \operatorname{Im} V$

1: Compute the QR decomposition of V^T, i.e. find an orthogonal $Q \in \mathbb{R}^{r \times r}$, an upper-triangular $R \in \mathbb{R}^{r \times n}$, and a permutation matrix $P \in \mathbb{R}^{n \times n}$ such that $V^T P = QR$
2: Let $R = \begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$, where $R_{11} \in \mathbb{R}^{r \times r}$ is upper-triangular and $R_{12} \in \mathbb{R}^{r \times (n-r)}$
3: Solve a triangular system $\widetilde{R}_{12} = R_{11}^{-1} R_{12}$
4: Compute $\widetilde{V} = P \begin{bmatrix} I_r & \widetilde{R}_{12} \end{bmatrix}^T = [\widetilde{v}_{ij}] \in \mathbb{R}^{n \times r}$
5: Find a partition $\pi = \{C_1, C_2, \ldots, C_r\}$ such that $i \in C_j$ if and only if $j = \arg\max_k |\widetilde{v}_{ik}|$
6: Return π

Algorithm 2 QR decomposition with column pivoting for matrices with block-columns

Input: Matrix $X \in \mathbb{R}^{kr \times kn}$ of full rank, where $n, r, k \in \mathbb{N}$ and $r < n$
Output: Orthogonal matrix Q, upper-triangular matrix R, and permutation matrix P such that $XP = QR$

1: Denote $X = \begin{bmatrix} X_1 & X_2 & \cdots & X_n \end{bmatrix}$, where $X_i \in \mathbb{R}^{kr \times k}$
2: Find a block-column X_i with the largest Frobenius norm and swap it with X_1
3: Perform QR decomposition with column pivoting on X_1, i.e. find an orthogonal $Q_1 \in \mathbb{R}^{kr \times kr}$, an upper-triangular $R_1 \in \mathbb{R}^{kr \times k}$, and a permutation matrix $P_1 \in \mathbb{R}^{k \times k}$ such that $X_1 P_1 = Q_1 R_1$
4: Multiply all block-columns in X from the right by P_1
5: Multiply X from the left by Q_1^T
6: Repeat the procedure for $X(k+1 : kr, k+1 : kn)$, which computes the matrices Q_i, R_i, and P_i, for $i \in \{2, 3, \ldots, r\}$
7: Return $Q = Q_1 Q_2 \cdots Q_r$, $R = X$, and P with all of the column permutations recorded

where $Z_i \in \mathbb{R}^{n_A \times 3n_A}$, $i \in \{1, 2, 3\}$, are the block-rows of $Z \in \mathbb{R}^{3n_A \times 3n_A}$. Here, we conclude that the block-rows of V_r determine the clusters. This motivates us to modify the method in Algorithm 1 such that it clusters the block-rows of V_r. We see that we need to modify the QR decomposition algorithm with column pivoting used in line 1 of Algorithm 1, since applying column permutations can break the block-column structure we found in $[(P(\pi) \otimes I_{n_A})Z]^T$. Therefore, we have to limit the possible column permutations that are performed on V_r^T. This modified method is presented in Algorithm 2. Additionally, in line 5 of Algorithm 1, the absolute value needs to be replaced by a matrix norm (we use the Frobenius norm) of the $n_A \times n_A$ blocks in $\widetilde{V} = P \begin{bmatrix} I_{r_G n_A} & \widetilde{R}_{12} \end{bmatrix}^T$ and indices i, j, k need to represent the positions of the blocks.

Algorithm 1 returns the correct partition when $\operatorname{Im} V_r = \operatorname{Im} P(\pi)$ (Lemma 1 in [4]). Analogously, it can be proved

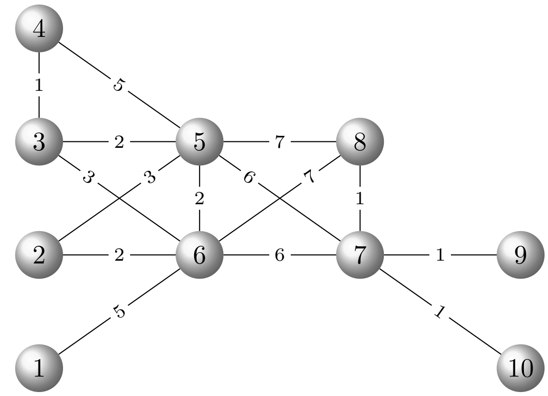

Fig. 1. Example of a multi-agent system defined on an undirected, weighted, connected graph. Vertices 6 and 7 are leaders. [9]

that the modified algorithm returns the correct partition when $\operatorname{Im} V_r = \operatorname{Im}(P(\pi) \otimes I_{n_A})$. Therefore, we expect for the modified algorithm to find a partition close to optimal when $\operatorname{Im} V_r$ is close, in some sense, to $\operatorname{Im}(P(\pi) \otimes I_{n_A})$.

We proved in [4] that Algorithm 1 is of linear complexity in the number of agents and quadratic in the number of clusters. Since the QR decomposition is computationally the most expensive part, we conclude that the same is true for the modified algorithm, except that it is further of cubic complexity in the order of the agent, since V_r is of size $n_G n_A \times r_G n_A$. Therefore, if agents are large-scale systems, it is sensible to apply MOR to agents before clustering. We will not consider agent reduction here, but it is an interesting problem for future work.

IV. Numerical Examples

A. Small-Scale Example

We use the example from [9], shown in Figure 1, except that the agents are undamped oscillators:

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ C = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

Therefore, the multi-agent system (1), (3) has $n = n_G n_A = 10 \cdot 2 = 20$ states, $m = m_G m_A = 2 \cdot 1 = 2$ inputs, and $p = p_G n_A = 15 \cdot 2 = 30$ outputs.

Let us fix the number of clusters to $r_G = 5$. Thus, the reduced order is $r = r_G n_A = 10$. The matrix $I_{n_G} \otimes A - L \otimes BC$ is not Hurwitz, but unstable poles are unobservable. Therefore, we can directly use IRKA, which converges to a ROM of order r in under 30 iterations, with the relative $\mathcal{H}_2$-error of $7.149 \cdot 10^{-3}$. Block-row clustering of the projection matrix V_r generated by IRKA returns the partition

$$\{\{1\}, \{2, 3, 4, 8, 9, 10\}, \{5\}, \{6\}, \{7\}\},$$

where the corresponding ROM produces the relative $\mathcal{H}_2$-error of 0.2130. The $\mathcal{H}_2$-optimal partition with five clusters (there are 42 525 partitions of the set $\{1, 2, \ldots, 10\}$ with five clusters) is

$$\{\{1, 2, 3, 4\}, \{5, 8\}, \{6\}, \{7\}, \{9, 10\}\},$$

with the relative $\mathcal{H}_2$-error of 0.1395.

Since all the multi-agent systems here are synchronized and not stable, we had to remove unstable states, which are also unobservable, before computing the $\mathcal{H}_2$-norms. We can see that $\{\mathbb{1} \otimes e_1, \mathbb{1} \otimes e_2\}$ spans the unstable subspace of $I \otimes A - L \otimes BC$, where $\mathbb{1}$ is a vector of ones and $e_1, e_2 \in \mathbb{R}^2$ are canonical vectors. Therefore, we find that the following sparse projection matrices $V_{\text{stab}}, W_{\text{stab}} \in \mathbb{R}^{n \times (n-2)}$

$$V_{\text{stab}} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}, \; W_{\text{stab}} = \begin{bmatrix} I \\ -\mathbb{1}^T \otimes e_1^T \\ -\mathbb{1}^T \otimes e_2^T \end{bmatrix}$$

remove unstable states.

B. Large-Scale Example

We randomly generated an undirected, unweighted, connected graph using the following Python 2.7.10 code (with NetworkX 1.10, NumPy 1.10.4, and SciPy 0.16.1 modules)

```
import networkx as nx
G = nx.powerlaw_cluster_graph(1000, 2, 0.5, seed=0)
L = nx.laplacian_matrix(G)
```

where the Holme-Kim algorithm [14] is utilized. The resulting graph has 1000 vertices and 1996 edges. We decided for the multi-agent system with the dynamics in (1), where the agents are undamped oscillators as in the previous example and the leaders are the first three agents. For the output, we chose (4), containing the states of the fourth and fifth agents, i.e.

$$y(t) = \left(\begin{bmatrix} e_4 & e_5 \end{bmatrix}^T \otimes I_2 \right) x(t).$$

Thus, the number of states, inputs, and outputs are $n = 2000$, $m = 3$, and $p = 4$.

We notice that the unstable states are now observable. Therefore, to apply IRKA, we need to remove the unstable states. We achieve this using sparse projection matrices $V_{\text{stab}}, W_{\text{stab}} \in \mathbb{R}^{n \times (n-2)}$ defined above. Let $V_{\text{IRKA}}, W_{\text{IRKA}} \in \mathbb{R}^{(n-2) \times r}$ denote the projection matrices computed by IRKA. Instead of applying the clustering algorithm to $V_{\text{stab}} V_{\text{IRKA}}$, where the last two rows are always zero, we computed the SVD decomposition of

$$\begin{bmatrix} V_{\text{stab}} V_{\text{IRKA}} & W_{\text{stab}} W_{\text{IRKA}} \end{bmatrix} \in \mathbb{R}^{n \times 2r}$$

and applied the clustering algorithm to the first r left singular vectors, since they span the dominant r-dimensional subspace.

We observed that IRKA does not converge (in under 100 iterations) and even returns unstable ROMs for larger reduced orders. Despite this, we noticed that using two iterations of IRKA already returns a good partition and that using more iterations does not significantly improve the $\mathcal{H}_2$-error associated with the resulting partition (we observed the same behavior when using output (3), with $2 \cdot 1996$ outputs). Figure 2 reports relative $\mathcal{H}_2$-errors due to clustering for different numbers of clusters. All $\mathcal{H}_2$-norms are computed with respect to the stable parts.

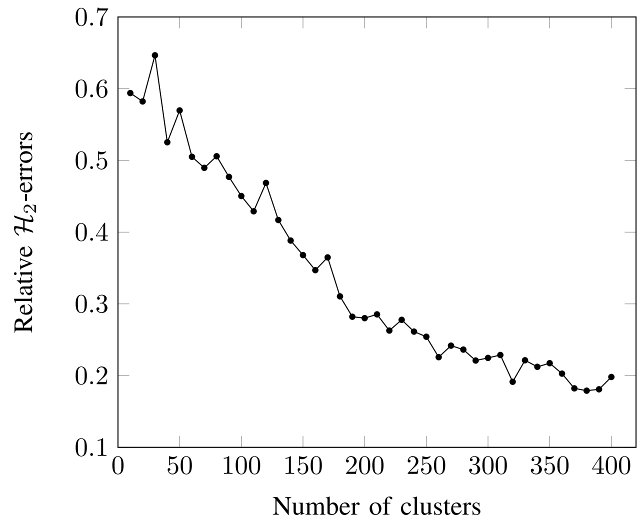

Fig. 2. Relative $\mathcal{H}_2$-errors when clustering a multi-agent system with 1000 agents.

V. Conclusion

We presented an extension of our method, combining IRKA and a clustering algorithm, for clustering-based MOR of multi-agent systems where agents have identical general LTI dynamics. Heuristically, it appears that this method finds a partition close to the optimal. We demonstrated this on a small-scale example, where the obtained partition results in the $\mathcal{H}_2$-error of the same order of magnitude as the optimal. Furthermore, we showed that this method is applicable to multi-agent systems with a large number of agents of small to medium order. We illustrated this on a large-scale example with 1000 agents of second order. A theoretical explanation that shows when the algorithm finds a partition close to optimal remains an open problem for future work. Combining the clustering method with the MOR of agents is an interesting problem for future work.

References

[1] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM J. Cont. Optim.*, vol. 48, no. 1, pp. 162–186, 2009.

[2] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Trans. Automat. Control*, vol. 53, no. 6, pp. 1503–1509, July 2008.

[3] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 57, no. 1, pp. 213–224, Jan. 2010.

[4] P. Mlinarić, S. Grundel, and P. Benner, "Efficient model order reduction for multi-agent systems using QR decomposition-based clustering," in *54th IEEE Conference on Decision and Control (CDC), Osaka, Japan*, Dec. 2015, pp. 4794–4799.

[5] T. Ishizaki and J.-I. Imura, "Clustered model reduction of interconnected second-order systems," *Nonlinear Theory and Its Applications, IEICE*, vol. 6, no. 1, pp. 26–37, 2015.

[6] T. Ishizaki, K. Kashima, J.-I. Imura, and K. Aihara, "Model reduction and clusterization of large-scale bidirectional networks," *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 48–63, Jan. 2014.

[7] B. Besselink, H. Sandberg, and K. H. Johansson, "Model reduction of networked passive systems through clustering," in *Proc. European Control Conf. ECC 2014, Strasbourg*, June 2014, pp. 1069–1074.

[8] N. Monshizadeh and A. van der Schaft, "Structure-preserving model reduction of physical network systems by clustering," in *53rd IEEE Conference on Decision and Control (CDC)*, Dec. 2014, pp. 4434–4440.

[9] N. Monshizadeh, H. L. Trentelman, and M. K. Çamlibel, "Projection-based model reduction of multi-agent systems using graph partitions," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 145–154, June 2014.

[10] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Philadelphia, PA: SIAM Publications, 2005.

[11] N. Monshizadeh, H. L. Trentelman, and M. K. Çamlibel, "Stability and synchronization preserving model reduction of multi-agent systems," *Systems Control Lett.*, vol. 62, no. 1, pp. 1–10, 2013.

[12] S. Gugercin, A. C. Antoulas, and C. A. Beattie, "$\mathcal{H}_2$ model reduction for large-scale dynamical systems," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 609–638, 2008.

[13] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral relaxation for k-means clustering," in *NIPS*, 2001, pp. 1057–1064.

[14] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Phys. Rev. E*, vol. 65, p. 026107, Jan. 2002.