

# Data-guided Control: Clustering, Graph Products, and Decentralized Control

Airlie Chapman, Andrés D. González, Mehran Mesbahi, Leonardo Dueñas-Osorio, and Raissa M. D’Souza

**Abstract**—This paper presents a novel approach to form an interdependent network model from time-varying system data. The research incorporates system meta-data using  $k$ -means clustering to form a layered structure within the dynamics. To compactly encode the layering, a Cartesian product model is fit to time-varying data using convex optimization. We show that under special situations a closed form solution of this model can be acquired. The Cartesian form is particularly conducive to reasoning about the role of the interdependent network layers within the dynamics. This is illustrated through the derivation of a distributed LQR controller which requires only knowledge of local layers in the network to apply. To demonstrate the applicability of this work, the proposed methods and analysis is applied to time-series data from a high-fidelity interdependent infrastructure network simulation.

**Index Terms**—Interdependent networks, Network clustering, Meta-data modeling, Distributed LQR, Dynamic mode decomposition

## I. INTRODUCTION

The complexity and scale of networked dynamic systems presents a challenge to direct modeling. Traditional approaches are a highly specialized activity involving tuning to the characteristics of the system. An alternative, advocated in this paper, is to form a model using time-series data coupled with limited knowledge of the network structure. When time-series data is sufficiently rich, this data-driven system identification approach produces a compact model which captures observed features and can be analyzed using traditional control-theoretic techniques.

System identification algorithms are typically applied to input/output data which is generated by experimental processes or high-fidelity simulations, so as to disseminate high-level features of the system [1]. Output time-series simulation data  $\{\phi(0), \dots, \phi(T)\}$ , or *snapshots*, can also be used to approximate a linear memory-less dynamics  $\phi(k+1) = A\phi(k)$  for

an unknown linear operator  $A$ . The data-driven process of recovering the eigenvectors and eigenvalues of operator  $A$  is referred to as dynamic mode decomposition (DMD) [2]. DMD has demonstrated its ability to extract essential features of a dynamic response, reliably approximating models with low complexity [3], [4]. Rowley *et al.* [3] connected the efficacy of DMD to its grounding in Koopman operator theory.

As we are examining the application of DMD to networked dynamic systems, we consider each time-varying state in the data as corresponding to a node in the network. The dynamic network abstraction implies that edges between nodes are encoded in the linear operator  $A$ . If there is knowledge of the dynamic interdependencies between states, that is the underlying edges between nodes, then constraints can be placed on the data-driven system identification to encode this network structure.

A specific form of network structure considered is when functional layering occurs within the dynamics. Layered network structures composed of many *interdependent network* layers are purposefully induced in many systems and also evolve naturally. Man-made structures, due to design advantages, often take on interdependent network forms. Infrastructure networks [5], [6], sensor and actuator placement in smart structures [7], [8] and supply chain networks [9] are just a few examples of this phenomenon. It has been previously demonstrated that graph product networks are particularly well suited at compactly describing interdependent network structures [10], [11], [12]. For special graph products, like the Cartesian and Kronecker graph products, system-theoretic properties can also be described in terms of these topological graph based representations. Examples are convergence rates, controllability and observability features and stabilizing feedback properties [12], [13], [14].

A focus of this work is to incorporate available classification data into the DMD to help encode network layers within the dynamics. One such classification data is coarse-known *functional* features. Examples include the designation of state’s measurement types such as a velocity state and temperature states in a fluid particle network, or known classes of states such as those which correspond to different types of vehicles in a transportation network. Another classification type is *static* real-valued data. Topological positions corresponding to physical states in stationary distributed sensor networks and ages of human subjects in a opinion network fall in this

The research of the authors has been supported by ARO grant W911NF-13-1-0340 and ONR grant N0001412-1-1002. A. Chapman is with the Department of Mechanical Engineering, University of Melbourne, Victoria 3010, Australia. Email: airlie.chapman@unimelb.edu.au. M. Mesbahi is with the Department of Aeronautics and Astronautics, University of Washington, WA 98195, USA. Email: mesbahi@uw.edu. A. González is with the School of Industrial and Systems Engineering at the University of Oklahoma, OA 73019, USA. Email: andres.gonzalez@ou.edu. L. Dueñas-Osorio is with the Department of Civil and Environmental Engineering, Rice University, TX 77005, USA. Email: leonardo.duenas-osorio@rice.edu. R. D’Souza is with the Complexity Sciences Center, Department of Mechanical and Aerospace Engineering and the Department of Computer Science, University of California, Davis, CA 95616, USA. Email: raissa@cse.ucdavis.edu.

data type. Ideally, data-driven modelling should be sympathetic to these classifications. We propose an approach to use these classifications to induce clustering of nodes into meta-nodes that have similar functional and static properties. We then proceed to fit an interdependent network system model with respect to these functional and static properties using a Cartesian graph product.

The organization of the paper is as follows. We begin by introducing background material on the Cartesian product and data-driven system identification in §II. §III proposes a novel clustering technique for generating meta-nodes with similar functional and static properties. These meta-nodes are then used to generate a Cartesian product model of the time-series data which encodes the interdependent layered network in §IV. To illustrate the benefit of the Cartesian product model, properties of the model are explored in §V including its ability to induce distributed controllers which are optimal in the linear quadratic regulator (LQR) sense. The paper is concluded with an illustrative example over an interdependent infrastructure network and some final remarks in §VI and §VII, respectively.

## II. BACKGROUND

In this section, we provide notation and a brief background on constructs and models that will be used throughout the paper. For a matrix  $A \in \mathbb{R}^{m \times n}$  with  $ij$ th element  $a_{ij}$ , the vectorization operator is  $\text{vec}(A) = [a_{11}, \dots, a_{m1}, a_{12}, \dots, a_{m2}, \dots, a_{1n}, \dots, a_{mn}]^T$ . The  $i$ th indicator vector denoted as  $e_i$  is defined as the  $i$ th column of the identity matrix. The Kronecker product of matrices  $M$  and  $N$  is  $M \otimes N$ , and the Kronecker sum for square matrices  $A \in \mathbb{R}^{n_a \times n_a}$  and  $B \in \mathbb{R}^{n_b \times n_b}$  is  $A \oplus B = A \otimes I_{n_b} + I_{n_a} \otimes B$ .

### A. Cartesian Product Graphs

A weighted digraph  $\mathcal{G} = (V, E, W)$  is characterized by a node set  $V$  with cardinality  $n$ , an edge set  $E$  comprised of ordered pairs of nodes with cardinality  $m$ , and a weight set  $W$  with cardinality  $m$ , where an edge exists from node  $i$  to  $j$  if  $(i, j) \in E$  with edge weight  $w_{ji} \in W$ . The adjacency matrix  $\mathcal{A}(\mathcal{G})$  is a  $n \times n$  matrix with  $[\mathcal{A}(\mathcal{G})]_{ij} = w_{ji}$  if  $(i, j) \in E$  and 0 otherwise.

There are a number of ways to synthesize large-scale networks from a set of smaller graphs [15]. The Cartesian product is one such method and is defined for a pair of factor graphs  $\mathcal{G}_1 = (V_1, E_1, W_1)$  and  $\mathcal{G}_2 = (V_2, E_2, W_2)$  and denoted by  $\mathcal{G} = \mathcal{G}_1 \square \mathcal{G}_2$ . The (Cartesian) product graph  $\mathcal{G}$  has the vertex set  $V_1 \times V_2$ , and there is an edge from vertex  $(i, p)$  to  $(j, q)$  in  $V_1 \times V_2$  if and only if either  $i = j$  and  $(p, q)$  is an edge in  $E_2$ , or  $p = q$  and  $(i, j)$  is an edge in  $E_1$ . The corresponding weight if an edge exists is  $w_{((j,q),(i,p))} = \delta_{pq}w_{ji} + \delta_{ij}w_{qp}$ , where  $\delta_{uv} = 1$  if  $u = v$  and 0 if otherwise. An example of a Cartesian product is displayed in Figure 1. The Cartesian product exhibits a distributive adjacency matrix property over its factors  $\mathcal{A}(\mathcal{G}_1 \square \mathcal{G}_2) = \mathcal{A}(\mathcal{G}_1) \oplus \mathcal{A}(\mathcal{G}_2)$  [15].

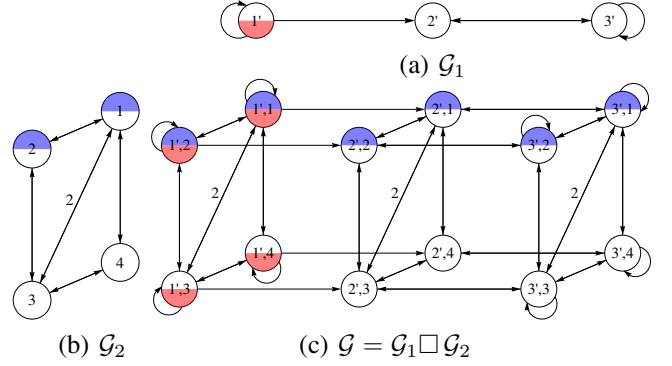


Figure 1: Factor graphs  $\mathcal{G}_1$  and  $\mathcal{G}_2$  and composite graph  $\mathcal{G}_1 \square \mathcal{G}_2$ . Edge weights of all graphs are 1 unless otherwise marked.

### B. Linear Operator Theory

An objective of DMD and Koopman operator theory is to obtain an infinite dimensional operator that describes a non-linear dynamic process  $x(k+1) = f(x(k))$  when provided with a set of observables  $\phi(k) = g(x(k)) \in \mathbb{R}^n$ . Truncating the infinite dimensional operator to create a finite dimensional operator  $A \in \mathbb{R}^{n \times n}$  forms a discrete-time linear time-invariant representation that approximates the time evolution of the observables as  $\phi(k+1) \approx A\phi(k)$ . The novelty of this approach is that  $A$  can be approximated using  $(T+1)$ -time state snapshots of the observables stored in the matrix  $\Phi = [\phi(0) \ \phi(1) \ \dots \ \phi(T)] \in \mathbb{R}^{n \times (T+1)}$ .

Constructing  $A$  from data can be restated with respect to a norm measure as a search for an  $A$  such that  $\|\phi(k+1) - A\phi(k)\|$  is small for all consecutive snapshot pairs  $(\phi(k), \phi(k+1))$  where  $\|\cdot\|$  is an arbitrary norm.<sup>1</sup> This is equivalent to the optimization problem

$$\min_{A \in \mathbb{R}^{n \times n}} \|\Phi_2 - A\Phi_1\|, \quad (1)$$

where  $\Phi$  is used to form  $\Phi_1 = [\phi(0), \phi(1), \dots, \phi(T-1)]$  and  $\Phi_2 = [\phi(1), \phi(2), \dots, \phi(T)]$ . With no additional structure on the problem, the optimal solution under the Frobenius norm can be solved in closed form using the least-squares identity  $A^* = \arg \min_{A \in \mathbb{R}^{n \times n}} \|\Phi_2 - A\Phi_1\|_F = \Phi_2 \Phi_1^T (\Phi_1 \Phi_1^T)^{-1}$ .

## III. CLUSTERING

Each node  $i \in V$  in a networked system is assumed to have a state  $\phi_i(k) \in \mathbb{R}$  at time  $k$  which is dependent on all node states in the network and evolves dynamically through the model described in §II-B, where  $\phi(k) = [\phi_1(k) \ \phi_2(k) \ \dots \ \phi_n(k)]^T$  and  $\Phi \in \mathbb{R}^{n \times (T+1)}$  combines this data for  $T+1$  time steps. Each node is assumed to have a functional labelling which partitions the nodes into  $n_s$  disjoint nonempty sets  $\mathcal{S} = (S_1, S_2, \dots, S_{n_s})$  called *partitions* with  $V = \bigcup_{i=1}^{n_s} S_i$  and  $S_i \cap S_j = \emptyset$ . These functional classifiers are assumed to encode known heterogeneity within the network and form the basis of the interdependent network

<sup>1</sup>Unless otherwise stated assume that  $\|\cdot\|$  is the Frobenius norm.

structure, namely the layers in the network. Additional *functional* labelling, with associated partitions  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$  can also be added, forming multiplexed network layers, but for simplicity only one is formally presented. It is assumed that additional real-valued *static* data  $\psi_i \in \mathbb{R}^p$  is also associated with each node  $i$ . Concatenating this non-time varying data forms the matrix  $\Psi = [\psi_1 \ \psi_2 \ \dots \ \psi_n]^T \in \mathbb{R}^{n \times p}$ .

The objective of this work is to form a reduced-order dynamics for the network sympathetic to the data triplet  $(\Phi, \Psi, \mathcal{S})$ . The approach consists of two stages. The first stage clusters the nodes into *meta-nodes* based on the partition  $\mathcal{S}$  and the partition  $\mathcal{R}$  across the nodes which is formed using data clustering over the states  $\phi_i$  and  $\psi_i$ . The second stage takes the newly found meta-node states and forms a low-order approximation of the linear operator using a Cartesian product which aligns with the first stage clusters.

Dimensionality is often a challenge when reasoning about large-scale networks. A favorable approach is to formulate a compact mathematical representation of the system. A reduced order recovery operator may be formulated, for example using a data clustering technique. To include the static data into the modeling, nodes that are ‘similar’ with respect to the states  $\phi_i$  and  $\psi_i$  are grouped into  $n_r$  non-empty partitions  $\mathcal{R} = (R_1, R_2, \dots, R_{n_r})$ . This approach reduces the order of the dynamics from  $n$  to  $n_r$ . Selecting good measures of similarity is key to maintaining an accurate reduced-order representation of the system. One such approach approach is  $k$ -means clustering which, when provided with a classifier state  $\xi_j \in \mathbb{R}^s$  for each node  $j$ , constructs partitions  $\mathcal{R}$  so as to minimize the variance  $\text{Var}R_i$  of the node classifier within each cluster  $R_i$ . For the complete classifier matrix  $\Xi = [\xi_1 \ \xi_2 \ \dots \ \xi_n]^T \in \mathbb{R}^{n \times s}$ . The associated cost function is

$$J_{\Xi}(\mathcal{R}) = \sum_{i=1}^{n_r} |R_i| \text{Var}R_i = \sum_{i=1}^{n_r} \sum_{j \in R_i} \|\xi_j - \mu_i\|^2, \quad (2)$$

where  $\mu_i = \frac{1}{|R_i|} \sum_{j \in R_i} \xi_j$  is the mean of the points in  $R_i$ . The mean can be considered a representative classifier for the cluster. Finding the partitions  $\mathcal{R}$  that minimizes (2) is generally NP-hard with many proposed sub-optimal solutions in the literature [16]. The  $k$ -means algorithm is the most commonly applied heuristic method to provide a ‘favorable’  $\mathcal{R}$  for the cost function (2).

With the objective of clustering with respect to the static data  $\Psi$  and the time-scale data  $\Phi$ , a suitable vector is  $\xi_j(\gamma) = \begin{bmatrix} \gamma \psi_j \\ (1-\gamma) \phi_j \end{bmatrix} \in \mathbb{R}^{p+T+1}$  where  $\gamma \in [0, 1]$ . The aggregated classifier matrix is then  $\Xi(\gamma) = [\gamma \Psi \ (1-\gamma)\Phi] \in \mathbb{R}^{n \times (p+T+1)}$ . The role of the variable  $\gamma$  is as a tuning parameter which varies the importance of node similarity from a clustering based solely on the static data when  $\gamma = 1$  to one based wholly on the time-scale data when  $\gamma = 0$ . To improve sensitivity to  $\gamma$ , a cursory scaling can also be performed on  $\Psi$  and  $\Phi$ . One such scaling is to center each column of the matrix about 0 and then normalize the matrix using the Frobenius norm as  $h(\Psi) = (\Psi - \mathbf{1}^T \otimes \mathbb{E}(\Psi)) / \|\Psi - \mathbf{1}^T \otimes \mathbb{E}(\Psi)\|_F$ ,

where  $\mathbb{E}(\Psi)$  is a column vector of averages over the rows of  $\Psi$ . The subsequent classifier matrix is then  $\Xi(\gamma) = [\gamma h(\Psi) \ (1-\gamma)h(\Phi)] \in \mathbb{R}^{n \times (p+T+1)}$ . For the tuning parameter  $\gamma = 1/2$ , the  $k$ -means clustering has a roughly even contribution from the static and time-scale data classifiers with  $\|\gamma h(\Psi)\|_F = \|(1-\gamma)h(\Phi)\|_F$ . The output of the  $k$ -means clustering with classifier  $\Xi(\gamma)$  over a nominated  $n_r$  clusters induces a classifier layering encoded in  $\mathcal{R}$ .

#### IV. CARTESIAN NETWORK MODEL: INTRA AND INTER DYNAMIC DEPENDENCIES

A network node’s membership in the functional partition  $\mathcal{S}$  and classifier partition  $\mathcal{R}$  describe the ‘character’ of the node. Similar nodes share similar memberships and as such model order reductions that group nodes with the same functional and classifier memberships can be applied. The partitions  $\mathcal{S}$  and  $\mathcal{R}$  can be used to create a *finer* partition  $\mathcal{T} = \{T_{ij}\}$  with  $T_{ij} = S_i \cap R_j$  for each  $S_i \in \mathcal{S}$  and  $R_j \in \mathcal{R}$ . The partition  $\mathcal{T}$  is a disjoint clustering of nodes of size  $n_s n_r$ . It is assumed that  $\mathcal{S}$  and  $\mathcal{R}$  are coarse enough such that no element of  $\mathcal{T}$  is empty. By fusing node states in the same cluster, a fused meta-node can be formed. The  $ij$ th partition then corresponds to a fused meta-node with accumulative time-series state  $\bar{\phi}_{ij}(k) = \sum_{h \in T_{ij}} \phi_h(k)$ . Isolating the dynamic features both within layers and between layers provides an ordered state vector that can be used to reason about the character of the network as a whole. Focusing on the network layers based on the partition  $\mathcal{S}$ , the state vector  $\bar{\phi}(k)$  can be ordered with respect to  $\mathcal{S}$  into  $n_s$  layers of  $\mathcal{S}$ . The time-series snapshot  $k$  becomes the stacked vectors of  $n_s$  layered state vectors with  $\bar{\phi}_{\mathcal{S}}(k) = [\bar{\phi}_{S_1}(k)^T, \bar{\phi}_{S_2}(k)^T, \dots, \bar{\phi}_{S_{n_s}}(k)^T]^T$  where  $\bar{\phi}_{S_i}(k) = [\bar{\phi}_{i1}(k), \dots, \bar{\phi}_{in_r}(k)]^T$ . The mapping of  $\phi(k)$  to  $\bar{\phi}_{\mathcal{S}}(k)$  with the order prescribed by  $\mathcal{S}$  can be compactly represented through the matrix  $P^{\mathcal{S}}$  with its right inverse  $Q^{\mathcal{S}}$  defined as

$$P_{p_{ij}h}^{\mathcal{S}} = \begin{cases} 1 & \text{if } h \in T_{ij} \\ 0 & \text{otherwise,} \end{cases} \quad Q_{h p_{ij}}^{\mathcal{S}} = \begin{cases} 1/|T_{ij}| & \text{if } h \in T_{ij} \\ 0 & \text{otherwise,} \end{cases}$$

and  $p_{ij} = (i-1)n_r + j$ , where  $P^{\mathcal{S}}Q^{\mathcal{S}} = I$ . For compactness, unless otherwise stated, the triple  $(P^{\mathcal{S}}, Q^{\mathcal{S}}, \bar{\phi}_{\mathcal{S}}(k))$  shall be written as  $(P, Q, \bar{\phi}(k))$ .<sup>2</sup> The meta-node states can be concatenated to form a time-series block matrix  $\tilde{\Phi}$ , as per §II-B, which can be represented in terms of the non-clustering state  $\Phi$  as  $\tilde{\Phi} = P\Phi$ , and  $\tilde{\Phi}_i = P\Phi_i$ , for  $i \in \{1, 2\}$ . The quality of the cluster partition  $\mathcal{T}$  evaluated using the  $k$ -means cost (2) is  $J_{\Phi}(\mathcal{T}) = \|\Phi - Q\tilde{\Phi}\|_F^2 = \|(I - QP)\Phi\|_F^2$ . A non-structured reduced order linear operator  $A$  can be formed by solving (1), which is sympathetic to the partition structure of  $\mathcal{S}$ . Consider  $A$  as a block matrix composed of  $|S_i| \times |S_k|$  subblocks  $A_{S_i S_k}$  which encodes the coupling between partition  $S_i$  and  $S_k$ . The main diagonal blocks capture the intra-layer coupling and the off-diagonal elements describe the inter-layer dependencies leading to the  $\mathcal{S}$ -layered dynamics

<sup>2</sup>A similar triple  $(P^{\mathcal{R}}, Q^{\mathcal{R}}, \bar{\phi}_{\mathcal{R}}(k))$  can be constructed if  $\bar{\phi}(k)$  is ordered by partitions  $\mathcal{R}$  rather than  $\mathcal{S}$ .

$\bar{\phi}_{ij}(k+1) = e_j^T \sum_{k=1}^{n_s} A_{S_i S_h} \bar{\phi}_{S_i}(k)$ . Permuting the rows and columns of  $A$  to correspond to the linear operator for the state vector  $\bar{\phi}_{\mathcal{R}}(k)$  rather than  $\bar{\phi}_{\mathcal{S}}(k)$  forms a similar  $\mathcal{R}$ -layered dynamics  $\bar{\phi}_{ij}(k+1) = e_i^T \sum_{k=1}^{n_r} A_{R_j R_h} \bar{\phi}_{R_j}(k)$  where  $A_{R_j R_k}$  encodes the coupling between partition  $R_j$  and  $R_k$ .

We propose a model to encode the partitions  $\mathcal{S}$  and  $\mathcal{R}$  as *factors* of a Cartesian linear operator. The model assumes a Cartesian graph product relationship structure based on a node's membership in  $\mathcal{S}$  and  $\mathcal{R}$ , with  $\bar{\phi}_{ij}(k)$  directly coupled to  $\bar{\phi}_{pq}(k)$  if and only if either  $i = p$  or  $j = q$ . Further, the model assumes a common  $\mathcal{S}$  (and  $\mathcal{R}$ ) intra-layer dependency, i.e.,  $A_{S_i S_i} = A_{\mathcal{S}}$  and  $A_{R_i R_i} = A_{\mathcal{R}}$  for all  $i$ , and no inter-layered dependency, i.e.,  $A_{S_i S_h} = 0$  and  $A_{R_i R_h} = 0$  for all  $i \neq h$ . The corresponding form of the meta-node dynamics is then  $\bar{\phi}_{ij}(k+1) = e_j^T A_{\mathcal{S}} \bar{\phi}_{S_i}(k) + e_i^T A_{\mathcal{R}} \bar{\phi}_{R_j}(k)$ . Motivated by the similarity with the form of the adjacency matrix in the Cartesian graph product, we refer to these dynamics as *Cartesian dynamics*. The Cartesian dynamics can be compactly represented as

$$\bar{\phi}_{\mathcal{S}}(k+1) = (A_{\mathcal{R}} \otimes I_{n_r} + I_{n_s} \otimes A_{\mathcal{S}}) \bar{\phi}_{\mathcal{S}}(k) = A_{\mathcal{R}} \oplus A_{\mathcal{S}} \bar{\phi}_{\mathcal{S}}(k), \quad (3)$$

and similarly  $\bar{\phi}_{\mathcal{R}}(k+1) = A_{\mathcal{S}} \oplus A_{\mathcal{R}} \bar{\phi}_{\mathcal{R}}(k)$ . This reduced order representation of the network highlights the role of each layer within the dynamics as well as providing a compact representation of the functional flow through the dynamics. The optimal Cartesian linear operator  $A_{\oplus} = A_{\mathcal{R}} \oplus A_{\mathcal{S}}$  can be found by solving the convex optimization  $\beta_{\oplus} = f(A_{\oplus}) =$

$$\min_{A_{\mathcal{R}} \in \mathbb{R}^{n_s \times n_s}, A_{\mathcal{S}} \in \mathbb{R}^{n_r \times n_r}} \left\| \tilde{\Phi}_2 - A_{\mathcal{R}} \oplus A_{\mathcal{S}} \tilde{\Phi}_1 \right\|. \quad (4)$$

Additional convex constraints can be added to problem (4) to provide more structure to  $A_{\oplus}$ , for example element-wise positivity restrictions and spectral convergence bounds. Another option is to augment the cost function with a regularizer term such as the LASSO method  $L_1$ -norm sparsification of  $A_{\oplus}$ . These modifications retain the convexity of the problem and hence its solution efficiency. In its basic form, the optimization (4) can be reformulated as a least-squares problem with a closed form solution for  $A_{\oplus}$ .

**Theorem 1.** *The solution of (4) can be found by reshaping the vector  $a = [\text{vec}(A_{\mathcal{R}}^T)^T \text{vec}(A_{\mathcal{S}})^T]^T = (M^T M)^{-1} M^T \text{vec}(\tilde{\Phi}_2)$  where*

$$M = \begin{bmatrix} I_{n_r} \otimes V(\bar{\phi}(0)) & V(\bar{\phi}(0))^T \otimes I_{n_s} \\ I_{n_r} \otimes V(\bar{\phi}(1)) & V(\bar{\phi}(1))^T \otimes I_{n_s} \\ \vdots & \vdots \\ I_{n_r} \otimes V(\bar{\phi}(T-1)) & V(\bar{\phi}(T-1))^T \otimes I_{n_s} \end{bmatrix}.$$

Here,  $V(\cdot) : \mathbb{R}^{n_r n_s} \rightarrow \mathbb{R}^{n_r \times n_s}$  is the inverse operator of  $\text{vec}(\cdot)$ .

*Proof:* Applying the identity  $(B \otimes A) \text{vec}(X) = \text{vec}(AXB^T)$  then  $(B \otimes I) \text{vec}(X) = \text{vec}(XB^T) = (I \otimes X) \text{vec}(B^T)$  and  $(I \otimes A) \text{vec}(X) = (X^T \otimes I) \text{vec}(A)$  [17]. Decomposing  $f(A_{\oplus})^2$  into a sum of norms then the result follows. ■

The Cartesian linear operator is optimized using the partitioned time-series states  $\tilde{\Phi}$ , with the accuracy of the fit as  $\beta_{\oplus}$ . Ideally, the operator  $A_{\oplus}$ , once projected into the correct dimension, is also a favorable operator on the unmodified time-series data  $\Phi$ . Consider the approximate model generated by  $A_{\oplus}$  applicable to the full  $\phi(k)$  state. For every node  $h \in T_{ij} = S_i \cap R_j$ , its state  $\phi_h(k)$  is identical and follows the model

$$\phi_h(k+1) = e_j^T A_{\mathcal{S}} \left( \frac{1}{|S_i|} \sum_{p \in S_i} \phi_p(k) \right) + e_i^T A_{\mathcal{R}} \left( \frac{1}{|R_j|} \sum_{q \in R_j} \phi_q(k) \right).$$

In compact form, the approximate model is then  $\phi(k+1) = QA_{\oplus}P\phi(k)$ . The following theorem bounds the performance of the operator  $QA_{\oplus}P$ , measured against the original cost function (1) by the accuracy of the reduced order data fit  $\beta_{\oplus}$  and the cluster quality measure  $J_{\Phi}(\mathcal{T})$ . The result indicates that with better reduced order fit, improved cluster quality and finer clusters the more accurate  $A_{\oplus}$  can be used to describe the time evolution of the states.

**Theorem 2.** *Consider the pair  $(\beta_{\oplus}, A_{\oplus})$  and cost  $J_{\Phi}(\mathcal{T})$  defined in (4) and (2), respectively. Let  $\beta = \|QA_{\oplus}P\tilde{\Phi}_1 - \tilde{\Phi}_2\|$ , then  $(\max |T_{ij}|)^{-1/2} \beta_{\oplus} \leq \beta \leq (\min |T_{ij}|)^{-1/2} \beta_{\oplus} + J_{\Phi}(\mathcal{T})$ .*

*Proof:* As  $PQ = I$ ,  $A_{\oplus} \tilde{\Phi}_1 - \tilde{\Phi}_2 = (PQ)A_{\oplus}(P\tilde{\Phi}_1) - P\tilde{\Phi}_2 = P(QA_{\oplus}P\tilde{\Phi}_1 - \tilde{\Phi}_2)$ . Let  $Z = QA_{\oplus}P\tilde{\Phi}_1 - \tilde{\Phi}_2$  and observe that  $PP^T$  is a diagonal matrix with  $[PP^T]_{kk} = |T_k|$  then

$$\begin{aligned} \beta_{\oplus}^2 &= \|PZ\|^2 = \text{tr}(Z^T P^T P Z) \leq \sigma_{\max}(P^T P) \text{tr}(Z^T Z) \\ &= \sigma_{\max}(PP^T) \|Z\|^2 = \max |T_k| \beta^2. \end{aligned}$$

This corresponds to the lower bound in the result. Applying a similar approach to solve for the upper bound,  $Q^T Q$  is a diagonal matrix with diagonal elements  $1/|T_k|$  and noting that  $\|\tilde{\Phi}_2 - Q\tilde{\Phi}_2\| \leq \|\tilde{\Phi} - Q\tilde{\Phi}\|$  then from the triangle inequality, the result follows. ■

Additional relationship can be drawn from Theorem 2 by noting that partition  $\mathcal{T}$  is always finer than partitions  $\mathcal{S}$  and  $\mathcal{R}$ . This hierarchical partitioning implies that the cluster quality is bounded as  $J_{\Phi}(\mathcal{T}) \leq \min(J_{\Phi}(\mathcal{S}), J_{\Phi}(\mathcal{R}))$ . Further, the size of the partitions in  $\mathcal{T}$  is bounded by  $|T_{ij}| \leq \max |S_i|$  and  $|T_{ij}| \leq \max |R_j|$  and a nontrivial intersection of  $S_i$  and  $R_j$  implies that  $|T_{ij}| \geq 1$ . A relaxation of the inequality in Theorem 2 is  $\beta \in [(\min \{\max |S_i|, \max |R_j|\})^{-1/2} \beta_{\oplus}, \beta_{\oplus} + \min(J_{\Phi}(\mathcal{S}), J_{\Phi}(\mathcal{R}))]$ . The bounds indicate that the accuracy of  $A_{\oplus}$  is related to the cluster quality for the optimized classifier partition  $\mathcal{R}$ , with tighter bounds generated with smaller clusters.

## V. DISTRIBUTED ANALYSIS AND CONTROL

In this section we describe features of and techniques to study the Cartesian dynamics (3). Existing work typically embeds the Cartesian state matrix  $A_{\mathcal{R}} \oplus A_{\mathcal{S}}$  into continuous time dynamics opposed to the discrete time form (3) explored in §IV. Applying the approximate transform from discrete dynamics  $\bar{\phi}(k+1) = A_{\mathcal{R}} \oplus A_{\mathcal{S}} \bar{\phi}(k)$  sampled at time intervals  $\Delta t$  to

the continuous dynamics  $\dot{x}(t) = Ax(t)$  where  $x(t)$  is the continuous form of  $\bar{\phi}(k)$  then  $A_{\mathcal{R}} \oplus A_{\mathcal{S}} = e^{A\Delta t} \approx I + A\Delta t$ . As such the continuous dynamics for the discrete dynamics (3) can be approximated with a Cartesian state matrix as

$$A \approx \left[ \frac{1}{\Delta t} (A_{\mathcal{R}} - (1 - \gamma)I) \right] \oplus \left[ \frac{1}{\Delta t} (A_{\mathcal{S}} - \gamma I) \right] := A_1 \oplus A_2,$$

with  $\gamma \in \mathbb{R}$ . For notational convenience we assume that  $A_1$  and  $A_2$  are square matrices of dimension  $n_1 = n_s$  and  $n_2 = n_r$ , respectively. In this section, decompositional features are presented as well as the controlled (and observed) form of the dynamics. An attraction of the Cartesian dynamics (3) is that features of the full dynamics are often decomposable into similar features on the layers. Consider an  $n$ -node graph abstraction of an arbitrary state matrix  $A \in \mathbb{R}^{n \times n}$  formed by assuming the edge  $(i, j)$  is present with weight  $w_{ji}$  in  $\mathcal{G}(A)$  only if  $[A]_{ji} = w_{ji}$  is nonzero. The Cartesian state matrix can be decomposed as  $\mathcal{G}(A_{\mathcal{R}} \oplus A_{\mathcal{S}}) = \mathcal{G}(A_{\mathcal{R}}) \square \mathcal{G}(A_{\mathcal{S}})$ . The graph layers  $\mathcal{G}(A_{\mathcal{R}})$  and  $\mathcal{G}(A_{\mathcal{S}})$  represent to the partition  $\mathcal{R}$  and  $\mathcal{S}$ , respectively.

For situations where an external control is applied to the network, decompositional properties are also possible if the inputs are sympathetic to the network layers. One example occurs when a common input structure is applied to each layer, represented by the dynamics  $\dot{x}(t) = A_1 \oplus A_2 x(t) + B_1 \otimes I_{n_2} u(t)$ . A decomposed form of these dynamics on only one layer is  $\dot{x}_1(t) = A_1 x_1(t) + B_1 u_1(t)$ . Figure 1(a) depicts an example of these dynamics on  $\mathcal{G}_1 = \mathcal{G}(A_1)$  with the input (and output) applied to lower-shaded node 1' encoded in the input matrix  $B_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ . Figure 1(c) is the associated dynamics on  $\mathcal{G}(A_1 \oplus A_2)$  with  $B_1 \otimes I_{n_2}$  representing the lower-shaded nodes which are at a common location for each repeated layer of  $\mathcal{G}_1$ .

An attraction of the Cartesian operator form is that distributed controllers (observers) with performance guarantees can be formed. A controller of the form  $u(t) = (K \otimes M)x(t)$ , where  $K \in \mathbb{R}^{n_1 \times n_2}$  and  $M \in \mathbb{R}^{n_2 \times n_2}$  are diagonal matrices, can be implemented distributively with the local layer feedback appearing as  $u_i(t) = [M]_{ii} K x_i(t)$  where  $x_i(t)$  is the states in layer  $i$ . Consider the linear quadratic regulator (LQR) problem which involves finding the optimal control  $u$  for the following optimization problem

$$\begin{aligned} u^* &= \operatorname{argmin} \int_0^\infty x^T Q x + u^T R u + 2x^T N u \, dt \quad (5) \\ \text{s.t.} \quad \dot{x} &= A x + B u, \end{aligned}$$

where  $R \succ 0$  and  $Q - N R^{-1} N^T \succeq 0$ . The problem framework is fully defined by the 5-tuple  $(A, B, Q, R, N)$ . The closed form algebraic solution to (5) corresponds to a feedback law  $u = -Kx = -R^{-1}(B^T P + N^T)x$  [18]. The matrix  $P \succeq 0$  is the unique solution to the algebraic Riccati equation  $0 = Q + A^T P + P A - (P B + N) R^{-1} (P B + N)^T$ . The solution is characterized by the pair  $(K, P)$ . The following theorem describes a *distributed* LQR solution on the Cartesian dynamics for a special class of triple  $(Q, R, N)$ .

**Theorem 3.** *The solution pair for the optimal LQR controller on the 5-tuple  $(A_1 \oplus A_2, B_1 \otimes I, Q, R, N)$  is  $(K_1 \otimes M_2, P_1 \otimes M_2)$  where  $Q = Q_1 \otimes M_2 - P_1 \otimes F_2$ ,  $R = R_1 \otimes M_2$ ,  $N = N_1 \otimes M_2$ ,  $M_2$  is a positive diagonal matrix,  $F_2 = A_2 M_2 + M_2 A_2^T \preceq 0$  and  $(K_1, P_1)$  is the solution pair for the optimal LQR controller on the 5-tuple  $(A_1, B_1, Q_1, R_1, N_1)$ .*

*Proof:* The result follows from the distributive properties of the Kronecker product. ■

Theorem 3 can be considered as describing a type of inverse-LQR problem, whereby if each layer applies a scaled version of the *layered* LQR controller  $u_i(t) = [M]_{ii} K_1 x_i(t)$  with  $K_1$  the optimal gain for the triple  $(Q_1, R_1, N_1)$  then the full controller  $u(t) = K_1 \otimes M_2 x(t)$  is the optimal controller for the *global* LQR problem with triple  $(Q, R, N)$ . The appearance of  $A_2$  in the matrix  $Q$  is accredited to the fact the global controller is formed independently of the graph layer  $\mathcal{G}(A_2)$ . The extended version of this paper provides sufficient condition on  $\gamma$  for the existence of a negative semidefinite  $F_2(\gamma)$  required in Theorem 3 and guarantees that the triple  $(Q, R, N)$  satisfies the positive semidefinite requirements of the LQR optimization problem.

## VI. EXAMPLE

This section applies the clustering and Cartesian product model creation techniques to a high-fidelity infrastructure simulation. The simulation models the dynamic recovery of nodes in an infrastructure network in response to a damaging earthquake event. A detailed description of the simulation model is described in the publication by González and Dueñas-Osorio [19]. The simulation was applied to the water, gas and power interdependent infrastructure networks in Shelby County, TN.

The water state output of the simulation  $w(k) \in \mathbb{R}^{49}$  represents the states of the 49 water nodes at time  $k \in [0, \dots, T]$ . If  $w_i(k) = 0$  then water node  $i$  at time  $k$  is functional; if  $w_i(k) = 1$  then water node  $i$  is damaged. Similarly output is the 16 gas node state vector  $g(k) \in \mathbb{R}^{16}$  and the 60 power node state vector  $p(k) \in \mathbb{R}^{60}$ . Concatenating these over multiple recovery runs, the complete state of the network at time  $k$  is  $\phi(k) = \begin{bmatrix} w(k)^T & g(k)^T & p(k)^T \end{bmatrix}^T$ . Also available is the state  $\psi_i \in \mathbb{R}^2$  corresponds to the physical location of infrastructure node  $i$ , displayed in Figure 2a for the water and gas nodes.

The clustering approach from §III is applied using time-series repair data  $\Phi$  and static position data  $\Psi$  to form a classifier partition  $\mathcal{R}$ . Figure 3 visualizes the effect of  $\gamma$  as the clustering shifts from a time-series focused clustering with  $\gamma = 0$  to a position focused clustering with  $\gamma = 1$  for a partition set  $\mathcal{R}$  of dimension 8.

Following the method of §IV, with  $\gamma = 0.66$ , a functional partition  $\mathcal{S}$ , formed from the utility type, and the classifier partition  $\mathcal{R}$  is used to generate a finer partition  $\mathcal{T}$ . Constructing  $\bar{\phi}_{\mathcal{R}}(k)$  then the optimization (4) was solved to form the approximate Cartesian dynamics  $\bar{\phi}_{\mathcal{R}}(k + 1) =$

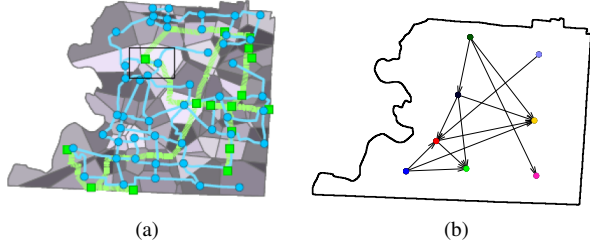


Figure 2: (a) Infrastructure network topology for the water (circles) and gas (square) networks in Shelby County, TN. (b)  $\mathcal{G}(A_S)$  with self loops omitted and edges with magnitude less than 0.01.

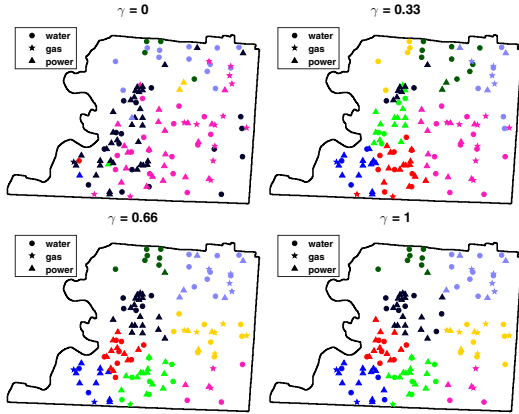


Figure 3: Weighted clustering balancing the importance of recovery proximity and spatial proximity.

$A_S \oplus A_{\mathcal{R}} \bar{\phi}_{\mathcal{R}}(k)$ . An  $L1$  regularizer term  $\|A_S\|_1 + \|A_{\mathcal{R}}\|_1$  was added to the cost function (4) to promote a sparse  $A_S$  and  $A_{\mathcal{R}}$ . The resultant  $A_{\mathcal{R}}$  is  $\begin{bmatrix} 0.208 & 0.013 & 0.005 \\ 0 & 0.116 & 0 \\ 0 & 0.010 & 0.202 \end{bmatrix}$  and the graph representation of  $A_S$  is shown in Figure 2b.

The Cartesian decomposition serves to isolate the effect of a node's utility type and its classifier cluster in its dynamics evolution. Examining  $A_{\mathcal{R}}$  with  $[A_{\mathcal{R}}]_{1,2}$  and  $[A_{\mathcal{R}}]_{3,2}$  nonzero, for example, describes the dependence of the water and power utilities on the gas utility. This can be attributed to the fact that gas nodes are dangerous to repair crews and must be preferentially repaired before power and water nodes. Similarly, the top left classifier cluster in Figure 2b has a significant impact on many of the other nodes due to the large number of water distribution stations in the area which are necessary to supply the large city of Memphis in the bottom left of the map.

## VII. CONCLUSION

This paper explores the inclusion of classification data into the time-series data model fitting for an interdependent networked system. Applying  $k$ -means clustering to the classification data, we were able to form meta-nodes that grouped similar network states together. The benefit of this grouping is that a compact Cartesian product model can be fit to the

time-series data. Furthermore, this can be formed efficiently using convex optimization, or in some cases directly through a closed form solution of the problem. The Cartesian product model presents a particularly attractive form for analysis and design, separating the role of each interdependent network layer in the system. Directions for future work are the exploration of Cartesian product modeling without the addition of classification data, and the examination of the distributed LQR formulation without the requirement that the cost matrices must conform to a layered structure.

## REFERENCES

- [1] L. Ljung, *System Identification (2nd Ed.) - Theory For the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- [2] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, no. 4, pp. 5–28, 2010.
- [3] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.
- [4] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, "Applications of the dynamic mode decomposition," *Theoretical and Computational Fluid Dynamics*, vol. 25, no. 1–4, pp. 249–259, 2011.
- [5] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 420–429.
- [6] X. Liu, H. Peng, and J. Gao, "Vulnerability and controllability of networks of networks," *Chaos, Solitons and Fractals*, vol. 80, pp. 125–138, 2015.
- [7] T. L. Rocha, S. Silva, and V. Lopes, "Optimal location of piezoelectric sensor and actuator for flexible structures," in *11th International Congress on Sound and Vibration*, no. 1998, 2004, pp. 1807–1814.
- [8] V. Gupta, M. Sharma, and N. Thakur, "Optimization criteria for optimal placement of piezoelectric sensors and actuators on a smart structure: a technical review," *Journal of Intelligent Material Systems and Structures*, vol. 21, no. 12, pp. 1227–1243, 2010.
- [9] J. Gong, J. E. Mitchell, A. Krishnamurthy, and W. A. Wallace, "An interdependent layered network model for a resilient supply chain," *Omega*, vol. 46, pp. 104–116, 2014.
- [10] J. Leskovec, "Kronecker Graphs : An Approach to Modeling Networks," *Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.
- [11] E. Parsonage, H. X. Nguyen, R. Bowden, S. Knight, N. Falkner, and M. Roughan, "Generalized graph products for network design and analysis," in *2011 19th IEEE International Conference on Network Protocols*, 2011, pp. 79–88.
- [12] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, "Controllability and observability of networks-of-networks via Cartesian products," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2668–2679, 2014.
- [13] A. Nguyen and M. Mesbahi, "A factorization lemma for the agreement dynamics," in *46th IEEE Conference on Decision and Control*, no. 1, 2007, pp. 288–293.
- [14] A. Chapman and M. Mesbahi, "Kronecker product of networked systems and their approximates," in *21st International Symposium on the Mathematical Theory of Networks and Systems*, 2014, pp. 1426–1431.
- [15] W. Imrich and S. Klavzar, *Product Graphs: Structure and Recognition*. New York: Wiley, 2000.
- [16] R. Xu and D. Wunsch, "Survey of Clustering Algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. New York: Cambridge University Press, 1991.
- [18] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Upper Saddle River, NJ, USA: Prentice Hall, 1990.
- [19] A. D. González, L. Dueñas-Osorio, M. Sánchez-Silva, and A. L. Medaglia, "The Interdependent Network Design Problem for Optimal Infrastructure System Restoration," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 5, pp. 334–350, 2016.