Scuola IMT Alti Studi Lucca

Coarse-graining complex networks for control equivalence

Questa è la versione preprint della seguente opera:

Original
Coarse-graining complex networks for control equivalence / Toller, D., Tribastone, M., Tschaikowski, M., Vandin, A.. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - 70:2(2025), pp. 1169-1175. [10.1109/TAC.2024.3448240]

Availability:
This version is available at: 20.500.11771/41687

Publisher:
IEEE

Published
DOI:10.1109/TAC.2024.3448240

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

09 November 2026

arXiv:2312.07421v1 [eess.SY] 12 Dec 2023

Coarse-graining Complex Networks for Control Equivalence

Daniele Toller[1], Mirco Tribastone[2], Max Tschaikowski[1], and Andrea Vandin[3]

***Abstract*— The ability to control complex networks is of crucial importance across a wide range of applications in natural and engineering sciences. However, issues of both theoretical and numerical nature introduce fundamental limitations to controlling large-scale networks. In this paper, we cope with this problem by introducing a coarse-graining algorithm. It leads to an aggregated network which satisfies control equivalence, i.e., such that the optimal control values for the original network can be exactly recovered from those of the aggregated one. The algorithm is based on a partition refinement method originally devised for systems of ordinary differential equations, here extended and applied to linear dynamics on complex networks. Using a number of benchmarks from the literature we show considerable reductions across a variety of networks from biology, ecology, engineering, and social sciences.**

I. Introduction

A large variety of complex systems of biological, ecological, social, and technological nature can be conveniently modeled as networks. Their dynamics can thus be studied as a (typically high-dimensional) system of differential equations where the evolution of each node depends on how it is connected to its neighbors and how it responds to external stimuli. Network control, i.e., the ability to steer the system toward a desired target behavior, is of crucial importance both theoretically and practically [1]. This involves two main steps [2]: (i) determining through which nodes to input a control action; and (ii) finding the actual values of the inputs.

Network science has made fundamental progress with approaches that identify a minimum subset of nodes to use as drivers in order to achieve control of the system, e.g., [3], [4], [5], [6]. The actual computation of a control law is usually a nontrivial task [7], [8]. For example, using standard results from systems theory [9], the controls that minimize energy dissipation can be computed explicitly in networks with linear dynamics. However, this involves the inversion of the controllability matrix, which ultimately gives a cubic dependency of the computational cost with the cardinality of the network [10]. Furthermore, the realizability of explicit solutions may be hindered in practice by numerical ill-conditioning of the controllability matrix [11], by physical constraints that preclude the use of arbitrarily large input signals [10], [12], or control trajectories that evolve far away in the state space [13], [14]. Overall, these issues have been recognized as fundamental limitations to controlling large networks [11], [1].

Across many branches of science and engineering, coarse graining is a typical strategy to cope with large-scale, intractable problems. Broadly speaking, it can be defined as an approach that maps a model onto a simpler one on which the analysis can be carried out more efficiently. Here we propose a coarse-graining algorithm for complex networks which preserves the control trajectories and the computation of optimal control values.

Although our ultimate goal is the control of networks with nonlinear dynamics, the study of the linear case is a fundamental step in this endeavour. We follow the standard setting [1] whereby the linear dynamical system induced by a network with N nodes and adjacency matrix $A \in \mathbb{R}^{N\times N}$ is given by

$$\partial_t x(t) = Ax(t) + Bu(t), \tag{1}$$

where $x(t) \in \mathbb{R}^N$ is network's state at time t, $B \in \mathbb{R}^{N\times K}$ is the input matrix, and $u(t) \in \mathbb{R}^K$ is the control input. The columns of B are given by K different unit vectors and describe the fact that K nodes in the network are *driver nodes*, each with its own distinct control input [4]; the remaining $N - K$ nodes, instead, are not controlled. We assume that $u(t)$ is bounded, which can thus take into account constraints for physical realizability [15], [10], [12].

In this context, the synthesis of a control law can be phrased as a problem of minimizing a cost function using $u(t)$ as the set of decision variables and (1) as the constraints. We consider a rather general cost function in the form

$$J(x[0], u, T) = F(x(T)) + \int_0^T \big[S(t, x(t)) + Q(t, u(t))\big]dt, \tag{2}$$

where the function F, called the final cost, may be used to steer the system toward a given target state at final time T; the integrand $R = S + Q$ is called the running cost. Note that the above integral is computed along the trajectory $x(t)$ determined by the initial condition $x[0]$, and the control input u. This covers the classical setting where only the energy of the control is to be minimized [1]; the term S allows for objectives where one is interested in driving the system toward a desired trajectory [1], possibly in a targeted fashion where only a subset of the state space, and not full control, is of interest, for instance [6], [16].

Our method partitions the nodes of a network into $n \leq N$ macro-nodes and builds a coarse-grained adjacency matrix $\hat{A} \in \mathbb{R}^{n\times n}$ and input matrix $\hat{B} \in \mathbb{R}^{n\times k}$, where $k \leq K$ is the number of macro-inputs. The state of the coarse-grained network $\hat{x}(t) \in \mathbb{R}^n$ is the solution of the system

$$\partial_t \hat{x}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), \tag{3}$$

and $\hat{u}(t) \in \mathbb{R}^k$ is the control macro-input. The partition is such that each macro-state of the coarse-grained network

[1] D. Toller and M. Tschaikowski are with Aalborg University, Denmark
[2] M. Tribastone is with IMT Lucca, Italy
[3] A. Vandin is with Sant'Anna Pisa, Italy and DTU Compute, Denmark

exactly preserves the sum of the values of the corresponding states in the original network whenever each macro-input is equal to the sum of the corresponding original input values.

The preservation of the dynamics up to sums of variables has been already established for linear and nonlinear systems of ordinary differential equations (ODEs) using the notion of forward equivalence [17]. Importantly, here we prove *control equivalence*, a property that guarantees the preservation of optimal costs. For this, we assume that the cost functional in (2) can be equivalently rewritten in terms of a cost functional $\hat{J}(\hat{x}, \hat{u}, T)$ that depends on the coarse-grained model. Under these conditions, the solution of the control problem for minimizing $\hat{J}(\hat{x}, \hat{u}, T)$ with the constraints given from (3) is such that one can recover a mapping from the optimal macro-inputs $\hat{u}^*(t)$ to optimal original inputs $u^*(t)$ that give rise to the optimal trajectory in (1). With this, the analysis of the original control problem can be entirely circumvented.

The crux of our method is that the coarse-graining can be found by computing the forward equivalence for the ODEs system $\partial_t x(t) = Ax(t)$. Importantly, it can be performed using the partition-refinement algorithm defined in [17]. The algorithm provides the reduction as the coarsest partition that refines an initial candidate partition, iteratively splitting its blocks until certain criteria are met [18], [17]. As a straightforward consequence of this fact, from the computational complexity analysis in [17] we obtain that control equivalence can be computed in $O(E \log N)$ time, where E is the number of non-zero entries in A.

The initial partition may be arbitrarily chosen. Similarly to the other applications of partition refinement [19], [20], [21], [22], this freedom permits certain state variables to be kept observable in the coarse-grained model. For example, let us assume that the original cost function $J(x, u, T)$ only concerns a given state component x_i; then, one can ensure by construction that the coarse-grained cost function $\hat{J}(\hat{x}, \hat{u}, T)$ can be written in terms of x_i by placing it in an initial singleton block for the partition-refinement algorithm.

Results: Our main technical contribution is a characterization result. Here, we prove that the reduction provided by our reduction algorithm is maximal in the sense that there exists no coarser refinement of a given initial partition which yields an optimality-preserving reduction. Additionally to the technical result, we conduct a large-scale evaluation of our framework on networks from public repositories. Here, the main finding is that networks that are controlled by a minimal set of driver nodes, as computed through [4], often allow for further substantial optimality-preserving reductions.

Further related work: Aggregation of dynamical systems has been extensively studied in the control theory literature since a long time starting from Aoki [23]; while it can be related to control equivalence, a method for computing is not provided. More closely related is the bisimulation/abstraction of control systems [24], [25], [26] where the largest bisimulation gives rise to the smallest reduced control system which coincides with the original one up to a given observation map. The main similarity with the bisimulation approaches is that control equivalence computes such an observation map, for which, in particular, the reduced system can be shown to be a so-called *consistent implementable abstraction* of the original one [27].

To the best of our knowledge, a (non-unique dimension-based) computation of observation maps has been addressed in the case with unbounded control domains [27]. Being described by systems of linear equations, it relies on matrix transformations and has thus a cubic time complexity in the number of nodes. Control equivalence, instead, is described by structural properties [28], [29] that are used in the computation of graph isomorphisms [30, Definition 5]. Hence, while considering a smaller class of observation maps than bisimulation, control equivalence avoids matrix transformations and enjoys a quasilinear time complexity in the number of edges; moreover, it considers bounded control domains. A less closely related work is that of balanced truncation and its extensions, see for instance [31, Chapter 9] and [32]. This is because the respective reductions are not exact and the control domains are unbounded; in addition, they have cubic time complexity [31]. Bisimulation approaches have been also studied for nonlinear systems [33], [34], [24], [25], are however computationally more demanding than the respective linear counterparts.

Notation: We denote by N and K, respectively, the number of nodes and driver nodes in the original network; similarly, $n \leq N$ and $k \leq K$ have the respective meaning in the reduced network. The respective dynamics of the original and the reduced networks are described by Eq. 1 and Eq. 3, where $x, \hat{x}$ are state vectors, while $u, \hat{u}$ are control vectors. Letters i, j denote node indices, l, l' refer to control inputs, $\mathcal{H}, \mathcal{H}'$ refer to partitions of $\{1, \ldots, N\}$, while H, H' to partition blocks. A partition $\mathcal{H}$ is said to refine a partition $\mathcal{H}'$ if every block $H' \in \mathcal{H}'$ can be expressed in terms of a disjoint union of blocks from $\mathcal{H}$. Note that $\mathcal{H}$ is a trivial refinement of itself. Likewise, $\{\{1, \ldots, N\}\}$ and $\{\{i\} \mid 1 \leq i \leq N\}$ are considered to be trivial partitions because the former aggregates all nodes to a single one, while the latter does not aggregate anything.

II. Control Equivalence

For two vectors $m \in \mathbb{R}^K$ and $M \in \mathbb{R}^K$ with respective l-th entries $m_l \leq M_l$ for every $l = 1, \ldots, K$, consider the real interval $[m_l; M_l] \subseteq \mathbb{R}$ and the K-dimensional cube

$$[m; M] = \prod_{l=1}^{K} [m_l; M_l] \subseteq \mathbb{R}^K.$$

We always consider control inputs $u(\cdot) \in [m; M]$, by which we mean that $u \colon \mathbb{R} \to [m; M]$ is a measurable function of time taking values in $[m; M]$.

Definition 1 (Optimal costs)**.** *Given a matrix $A \in \mathbb{R}^{N \times N}$ and a measurable control input $u(\cdot) \in [m; M]$, we denote the value functional at time $T \in \mathbb{R}$ when starting at $x[0] \in \mathbb{R}^N$ by*

$$J_u(x[0], T) = F\big(x^u(T)\big) + \int_0^T R\big(t, x^u(t), u(t)\big) dt,$$

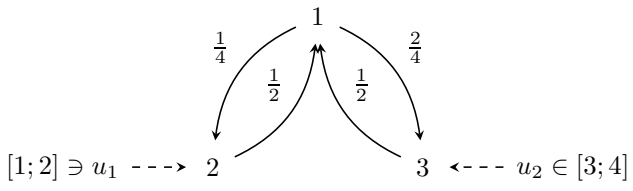

Fig. 1: A simple controlled network which will be used as a running example. Solid lines describe entries of the adjacency matrix, while dashed lines signify control inputs.

where x^u is the solution of (1)*, with initial condition $x[0]$, while F is the final cost function, and $R = S + Q$ is the running cost function. With this, the optimal costs at time T when starting at $x[0] \in \mathbb{R}^N$ are given by the value functions:*

$$V^{\inf}(x[0], T) = \inf\{J_u(x[0], T) \mid u : \mathbb{R} \to [m; M]\},$$
$$V^{\sup}(x[0], T) = \sup\{J_u(x[0], T) \mid u : \mathbb{R} \to [m; M]\},$$

where the control inputs $u(\cdot)$ are measurable.

In the following, we shall use the simple network provided in Fig. 1 to explain the main concepts.

Example 1. *Eq. 1 of the network from Fig. 1 is given by*

$$\begin{pmatrix} \partial_t x_1 \\ \partial_t x_2 \\ \partial_t x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}}_{A=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}_{B=} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

As next, we introduce control equivalence. For a set H, we denote its cardinality by $|H|$.

Definition 2 (Control Equivalence)**.** *Consider the system of ODEs in* (1)*. A control equivalence of this system is a partition $\mathcal{H}$ whose aggregation matrix $L = L(\mathcal{H})$ and the corresponding right inverse $\bar{L} = \bar{L}(\mathcal{H})$ satisfy $LA = LA\bar{L}L$. Specifically, $L \in \mathbb{R}^{n \times N}$ and $\bar{L} \in \mathbb{R}^{N \times n}$ are such that*

- *The i-th row of L encodes the i-th block of $\mathcal{H}$ by setting $L_{i,j} = 1$ if $j \in H_i$ and zero otherwise;*
- *The i-th column of $\bar{L}$ encodes the i-th block of $\mathcal{H}$ by setting $\bar{L}_{j,i} = 1/|H_i|$ if $j \in H_i$ and zero otherwise.*

When the partition $\mathcal{H}$ is clear from the context, we omit the dependence on $\mathcal{H}$ in $L(\mathcal{H})$ and $\bar{L}(\mathcal{H})$.

Example 2. *In the case of the network from Fig. 1, it is easy to see that the trivial partition $\{\{1, 2, 3\}\}$ is not a control equivalence, which corresponds to the impossibility of aggregating all nodes together.*

On the other hand, the partition $\mathcal{H} = \{H_1, H_2\} = \{\{2, 3\}, \{1\}\}$ is a control equivalence (intuitively: it is possible to aggregate the nodes 2 and 3). This can be seen by checking that the matrices

$$L = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \bar{L} = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & 0 \\ \frac{1}{2} & 0 \end{pmatrix}$$

satisfy $LA = LA\bar{L}L$.

Given a partition $\mathcal{H} = \{H_1, \dots, H_n\}$ of nodes $\{1, \dots, N\}$, we always assume without loss of generality that its first k blocks $H_1, \dots, H_k$ contain all at least one driver node, while the remaining blocks $H_{k+1}, \dots, H_n$ have no driver nodes.

Definition 3 (Reduced System)**.** *Consider the system of ODEs in* (1)*. Then, the reduced system underlying a partition $\mathcal{H}$ is given by* (3) *where*

- *$\hat{A} = LA\bar{L}$;*
- *For any $l \leq k$, the l-th column of $\hat{B} \in \mathbb{R}^{n \times k}$ is the l-th unit vector in $\mathbb{R}^n$;*
- *For any $l \leq k$, the l-th control satisfies $\hat{u}_l(\cdot) \in [\hat{m}_l; \hat{M}_l]$ with*

$$\hat{m}_l = \sum_{l' \in \mathcal{K}(H_l)} m_{l'} \quad \text{and} \quad \hat{M}_l = \sum_{l' \in \mathcal{K}(H_l)} M_{l'},$$

where $\mathcal{K}(H_l) \subseteq \{1, \dots, K\}$ are the control indices that steer the driver nodes contained in block H_l.

Note that the sets $\mathcal{K}(H_1), \mathcal{K}(H_2), \dots$ form a partition of $\{1, \dots, K\}$ (i.e., of the driver nodes). We used this fact to well-define the bounds $\hat{m}$ and $\hat{M}$ for the macro-inputs.

Example 3. *Continuing Example 2, the reduced system is*

$$\begin{pmatrix} \partial_t \hat{x}_1 \\ \partial_t \hat{x}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \frac{3}{4} \\ \frac{1}{2} & 0 \end{pmatrix}}_{\hat{A}=} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\hat{B}=} (\hat{u}_1),$$

where $\hat{u}_1(\cdot) = u_1(\cdot) + u_2(\cdot) \in [4; 6]$, while $\hat{x}_1 = x_2 + x_3$ and $\hat{x}_2 = x_1$. We note, in particular, that $\hat{x}_2 = x_1$ is not controlled, i.e., is not a driver node.

In order to associate a cost to a reduced system, the final and running costs have to satisfy an additional property. More specifically, for a partition of nodes $\mathcal{H}$, we call the final cost F and running cost R *constant on* $\mathcal{H}$ if $F(x) = F(y)$ and $R(t, x, u) = R(t, y, v)$ for all $x, y \in \mathbb{R}^N$, and $u, v \in \mathbb{R}^K$ that satisfy $Lx = Ly$ and $LBu = LBv$, for $L = L(\mathcal{H})$. In such a case, we study the well-defined reduced costs $\hat{R}$ and $\hat{F}$ for the reduced system defined for $\hat{x} \in \mathbb{R}^n$ and $\hat{u} \in [\hat{m}; \hat{M}]$ via

$$\hat{R}(t, \hat{x}, \hat{u}) := R(t, x, u) \qquad \text{and} \qquad \hat{F}(\hat{x}) := F(x),$$

for arbitrary x and u such that $\hat{x} = Lx$ and $\hat{B}\hat{u} = LBu$ (equivalently, $\hat{x}_i = \sum_{j \in H_i} x_j$ and $\hat{u}_l = \sum_{l' \in \mathcal{K}(H_l)} u_{l'}$ for all $1 \leq i \leq n$ and $1 \leq l \leq k$).

Example 4. *We continue Example 3 by picking as running cost and final cost, respectively, $R = 0$ and $F(x) = x_2 + x_3$. Then, R and F are constant on $\mathcal{H} = \{\{2, 3\}, \{1\}\}$ and we obtain $\hat{R}(\hat{x}) = 0$ and $\hat{F}(\hat{x}) = \hat{x}_1$. With these cost functions, the value functionals have the form*

$$J_u(x[0], T) = F(x^u(T)) = x_2^u(T) + x_3^u(T).$$

Definition 4 (Optimal cost of Reduced System)**.** *Let $\mathcal{H}$ be an arbitrary partition, and assume that F and R are, respectively, final and running costs that are constant on $\mathcal{H}$.*

Then, for a control input $\hat{u}(\cdot) \in [\hat{m}; \hat{M}]$, the value functional at time $T \in \mathbb{R}$ when starting at $\hat{x}[0] \in \mathbb{R}^n$ is given by

$$\hat{J}_{\hat{u}}(\hat{x}[0], T) = \hat{F}\big(\hat{x}^{\hat{u}}(T)\big) + \int_0^T \hat{R}\big(t, \hat{x}^{\hat{u}}(t), \hat{u}(t)\big)dt,$$

where $\hat{x}^{\hat{u}}$ is the solution of (3)*, with initial condition $\hat{x}[0]$. With this, the optimal costs of the reduced system* (3) *at time T when starting at $\hat{x}[0] \in \mathbb{R}^n$ are given by the reduced value functions:*

$$\hat{V}^{\inf}(\hat{x}[0], T) = \inf\{\hat{J}_{\hat{u}}(\hat{x}[0], T) \mid \hat{u} : \mathbb{R} \to [\hat{m}; \hat{M}]\},$$
$$\hat{V}^{\sup}(\hat{x}[0], T) = \sup\{\hat{J}_{\hat{u}}(\hat{x}[0], T) \mid \hat{u} : \mathbb{R} \to [\hat{m}; \hat{M}]\},$$

where the control inputs $\hat{u}(\cdot)$ are measurable.

We are now in a position to state our first result which is key in proving that the original and the reduced control system admit identical optimal costs if and only if $\mathcal{H}$ is a control equivalence.

Theorem 1 (Optimality-preservation)**.** *Let $\mathcal{H}$ be a control equivalence of the system* (1) *with final and running costs that are constant on $\mathcal{H}$.*

1) *For any control of the original system $u : \mathbb{R} \to [m; M]$, there exists a control of the reduced system $\hat{u} : \mathbb{R} \to [\hat{m}; \hat{M}]$ that satisfies*

$$J_u(x[0], T) = \hat{J}_{\hat{u}}(\hat{x}[0], T)$$

for every $T \in \mathbb{R}$ and every $x[0] \in \mathbb{R}^N$, where $\hat{x}[0] \in \mathbb{R}^n$ is given by $\hat{x}[0] = Lx[0]$.

2) *Conversely, for any control of the reduced system $\hat{u} : \mathbb{R} \to [\hat{m}; \hat{M}]$, there exists a control of the original system $u : \mathbb{R} \to [m; M]$ that satisfies*

$$J_u(x[0], T) = \hat{J}_{\hat{u}}(\hat{x}[0], T)$$

for every $T \in \mathbb{R}$ and every $x[0] \in \mathbb{R}^N$, where $\hat{x}[0] \in \mathbb{R}^n$ is given by $\hat{x}[0] = Lx[0]$. Additionally, the control u can be chosen as

$$u_{l'}(\cdot) = \begin{cases} m_{l'} + \dfrac{M_{l'} - m_{l'}}{\hat{M}_l - \hat{m}_l}(\hat{u}_l(\cdot) - \hat{m}_l) & , \text{ if } \hat{m}_l < \hat{M}_l \\ m_{l'} = M_{l'} & , \text{ otherwise} \end{cases}$$

for every $l \in \{1, \dots, k\}$ and $l' \in \mathcal{K}(H_l)$.

Armed with Theorem 1, we infer our main technical result.

Theorem 2 (Characterization of Optimality-preservation)**.** *Consider the system in* (1)*, and let $\mathcal{H}$ be a partition of the nodes. Then, $\mathcal{H}$ is a control equivalence if and only if*

$$V^{\inf}(x[0], T) = \hat{V}^{\inf}(\hat{x}[0], T),$$
$$V^{\sup}(x[0], T) = \hat{V}^{\sup}(\hat{x}[0], T)$$

for all final and running costs that are constant on $\mathcal{H}$, for every $T \in \mathbb{R}$, and every $x[0] \in \mathbb{R}^N$, where $\hat{x}[0] \in \mathbb{R}^n$ is determined by $\hat{x}[0] = Lx[0]$.

Example 5. *In the case of our running example, and with the cost functions as in Example 4, we obtain:*

- *For any $u_1(\cdot) \in [1; 2]$ and $u_2(\cdot) \in [3; 4]$, there exists $\hat{u}(\cdot) \in [4; 6]$ such that for any $x[0] \in \mathbb{R}^3$ and $\hat{x}[0] = Lx[0]$, for every T we have that*

$$x_2^u(T) + x_3^u(T) = \hat{x}_1^{\hat{u}}(T).$$

- *For any $\hat{u}_1(\cdot) \in [4; 6]$, there exist $u_1(\cdot) \in [1; 2]$ and $u_2(\cdot) \in [3; 4]$ such that for any $\hat{x}[0] \in \mathbb{R}^2$ and $x[0] \in \mathbb{R}^3$ with $\hat{x}[0] = Lx[0]$, we have that*

$$x_2^u(T) + x_3^u(T) = \hat{x}_1^{\hat{u}}(T).$$

Specifically, we can pick

$$u_1(t) = 1 + \tfrac{1}{2}(\hat{u}_1(t) - 4), \quad u_2(t) = 3 + \tfrac{1}{2}(\hat{u}_1(t) - 4).$$

We end the section by stating that control equivalence can be computed in quasilinear time in the number of edges. As the trivial partition made of singletons is a control equivalence, the refining procedure in the next theorem is guaranteed to terminate.

Theorem 3 (Computation of Control Equivalence)**.** *Consider system* (1) *and let $\mathcal{H}'$ be a partition of the nodes. Then the coarsest control equivalence refining $\mathcal{H}'$, denoted by $\mathcal{H}$, and the underlying reduced system $\partial_t \hat{x}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t)$, can be computed in at most $O(E \log N)$ steps, where E is the number of non-zero entries of $A \in \mathbb{R}^{N \times N}$.*

Example 6. *If we apply the partition refinement from [17] on the partition $\mathcal{H}' = \{\{1, 2, 3\}\}$, the algorithm will refine the block $\{1, 2, 3\}$ to the blocks $\{1\}, \{2, 3\}$, thus providing us with the previously seen partition $\mathcal{H} = \{\{1\}, \{2, 3\}\}$.*

We remark that the partition $\mathcal{H}'$ in Theorem 3 can be used to steer the search for control equivalences. Indeed, if one is interested in finding a reduction which does not aggregate node 1 with any other node, one could pick $\mathcal{H}' = \{\{1\}, \{2, 3\}\}$. With this choice, the partition refinement algorithm either splits the block $\{2, 3\}$ into $\{2\}, \{3\}$ or not. From the foregoing discussion, we know that the latter will be the case because $\{\{1\}, \{2, 3\}\}$ is a control equivalence. On the contrary, if one would use $\mathcal{H}' = \{\{1, 2\}, \{3\}\}$ (say, because one wants to obtain a reduced model which does not aggregate node x_3 with any other node), then the algorithm would split the block $\{1, 2\}$ and return the (trivial) control equivalence $\{\{1\}, \{2\}, \{3\}\}$ which does not aggregate any node.

III. Evaluation

We applied control equivalence to benchmark real-world directed networks studied in the literature [36], [4], [35]. They are taken from several domains including biology, ecology, engineering, online social networks, and the web with sizes ranging from 47 to 82169 nodes. The chosen networks show a wide spectrum of controllability ranging from 0.91% to 99.59% of driver nodes in the network as provided by the algorithm from [4] which yields a minimal set of driver nodes for network controllability. We considered initial partitions that separated the set of driver nodes from the rest of the network. In this way one can identify the

Name	*Type*	N	n	n/N	K	k	k/K	K/N
Kohonen	CITATION	3773	2652	70.29	2114	1123	53.12	56.03
subelj_cora	CITATION	23167	9254	39.94	10210	281	2.75	44.07
CitHepTh	CITATION	27771	19764	71.17	5994	650	10.84	21.58
CitHepPh	CITATION	34547	24936	72.18	8030	802	9.99	23.24
s208st	ELECTRONIC CIRCUITS	123	105	85.37	29	15	51.72	23.58
s420st	ELECTRONIC CIRCUITS	253	214	84.58	59	29	49.15	23.32
s838st	ELECTRONIC CIRCUITS	513	438	85.38	119	60	50.42	23.20
seagrass	FOOD WEB	50	43	86.00	13	8	61.54	26.00
ythan	FOOD WEB	136	89	65.44	69	24	34.78	50.74
grassland	FOOD WEB	89	31	34.83	46	10	21.74	51.69
littlerock	FOOD WEB	184	54	29.35	99	4	4.04	53.80
maayan-faa	INFRASTRUCTURE	1227	1027	83.70	363	215	59.23	29.58
opsahl-open-flights	INFRASTRUCTURE	2940	2422	82.38	872	469	53.78	29.66
US-powergrid-4941	INFRASTRUCTURE	4942	4688	94.86	575	445	77.39	11.63
tntp-Chicago-Regional	INFRASTRUCTURE	12980	12969	99.92	215	215	100.00	1.66
p2p-Gnutella08	INTERNET	6302	5273	83.67	4106	3116	75.89	65.15
p2p-Gnutella06	INTERNET	8718	7737	88.75	5033	4085	81.16	57.73
p2p-Gnutella05	INTERNET	8847	7800	88.17	5111	4115	80.51	57.77
p2p-Gnutella04	INTERNET	10877	9872	90.76	6004	5024	83.68	55.20
p2p-Gnutella-24A	INTERNET	26519	20273	76.45	18965	12875	67.89	71.51
Consulting	INTRA-ORGANIZATIONAL	47	47	100.00	2	2	100.00	4.26
Manufacturing	INTRA-ORGANIZATIONAL	78	74	94.87	1	1	100.00	1.28
rhesusbrain1	BRAIN CONNECTIONS	243	229	94.24	242	228	94.21	99.59
celegans-neuronal	NEURONAL	298	264	88.59	49	17	34.69	16.44
coliInter-NoAutoReg	METABOLIC	420	40	9.52	314	2	0.64	74.76
coliInter-full	METABOLIC	425	49	11.53	309	2	0.65	72.71
TRNYeast2-Costanzo	REGULATORY	689	92	13.35	565	43	7.61	82.00
maayan-Stelzl	PROTEIN-PROTEIN	1707	1276	74.75	765	420	54.90	44.82
TRNYeast1-Balaji	REGULATORY	4441	1983	44.65	4282	1829	42.71	96.42
yeast2017-full	REGULATORY	6854	6645	96.95	6648	6439	96.86	96.99
yeast2019-full	REGULATORY	6887	6705	97.36	6670	6488	97.27	96.85
ownership	REGULATORY	7254	360	4.96	5950	152	2.55	82.02
moreno-innovation	SOCIAL	242	214	88.43	29	7	24.14	11.98
CSphdA	SOCIAL	1883	241	12.80	1176	91	7.74	62.45
ego-twitter	SOCIAL	23371	2692	11.52	22432	2202	9.82	95.98
ego-gplus	SOCIAL	23629	2438	10.32	23497	2339	9.95	99.44
Epinions	SOCIAL	75880	37651	49.62	41627	7450	17.90	54.86
Slashdot	SOCIAL	82169	65436	79.64	3737	3680	98.47	4.55
linux	SOURCE CODE	30838	3767	12.22	20049	260	1.30	65.01
libre-film-trust	TRUST	875	425	48.57	359	93	25.91	41.03
WikiVote	TRUST	7116	2321	32.62	4736	3	0.06	66.55
librec-ciao-dvd	TRUST	4659	3059	65.66	3247	1748	53.83	69.69
moreno_blogs	WWW	1225	889	72.57	436	123	28.21	35.59
dimacs10_polblogs	WWW	1225	1177	96.08	126	81	64.29	10.29
wikipedia_link_gag	WWW	2930	1097	37.44	1185	52	4.39	40.44
EPAA	WWW	4272	680	15.92	3285	161	4.90	76.90
wikipedia_link_csb	WWW	5562	2274	40.88	2971	233	7.84	53.42
CaliforniaA	WWW	6176	1045	16.92	4489	96	2.14	72.68
wikipedia_link_mi	WWW	7997	2185	27.32	4825	153	3.17	60.34
wbcsstanfordA	WWW	9436	5326	56.44	3653	794	21.74	38.71
cfinder_google	WWW	15764	8678	55.05	5313	1487	27.99	33.70
wikipedia_link_bat_smg	WWW	21901	6845	31.25	14047	891	6.34	64.14

TABLE I: Reductions of published networks taken from [36], [4], [35]. N and n is, respectively, the size of the original and reduced network. K is the minimal number of driver nodes of the original network, computed using [4], while k is the number of driver nodes of the reduced network obtained via Theorem 3 and Definition 3. All reductions are optimality-preserving by Theorem 1 and 2.

macro-nodes in the coarse-grained network that represent original driver nodes.

Overall, these benchmark networks show a full range of reducibility by control equivalence measured as $\rho = n/N$, i.e., the ratio between the number n of macro-nodes and the number N of original nodes. This ranges from 0.10 in the case of metabolic networks to 0.999 in a road network (Table I). There is a high correlation (0.86) between the overall reduction and the reduction of driver nodes, indicated by $\rho_D = k/K$, i.e., the percentage ratio between the number k of macro driver nodes and the number K of original driver nodes. The former is obtained via Definition 3 and by initializing the reduction algorithm from Theorem 3 with an initial partition containing two blocks, one containing all driver nodes and its complement (as mentioned earlier, the set of driver nodes is computed using [4] and is minimal in size). With this in place, we note that in some networks, very small values of k were found (as $k < 5$), where K is between one and three orders of magnitude larger. This suggests a considerably more effective computation of the control law on the coarse-grained network, owing to the much smaller size of decision variables involved. Interestingly, the best reduction in the number of driver nodes, i.e., $\rho_D < 0.10$, occur in networks where the original density of driver nodes K/N is considerable, i.e., over 0.40.

Metabolic and regulatory networks have been previously highlighted to be difficult to control because of the large ratios K/N, as occurs in the models of yeast and E. coli [4]. Interestingly, here we find that for some of such models the coarse-grained networks turned out to be considerably more controllable. In particular the E. coli coarse-grained networks only require two driver macro-nodes instead of more than 300 nodes in the original networks. Ego networks reveal similar characteristics. Although they are extracted from different platforms, they are all difficult to control with $K/N > 0.95$. The corresponding coarse-grained networks, instead, show controllability ratios lower than 0.10 in all cases. WWW networks exhibit K/N values between 0.40 and 0.76 but the corresponding coarse-grained networks have similarly low values k/n between 0.02 and 0.04.

IV. Conclusion

In this work, we have introduced control equivalence, an optimality-preserving model reduction technique for linear control systems which can be computed in quasilinear time complexity in the number of nonzero matrix entries. We have conducted a large-scale evaluation on networks from public repositories, showing in particular that networks that are controlled by a minimal set of driver nodes often allow for further substantial optimality-preserving reductions. Future work will consider the optimality-preserving reduction of nonlinear control systems.

References

[1] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Rev. Mod. Phys.*, vol. 88, p. 035006, Sep 2016.

[2] L. Xiang, F. Chen, W. Ren, and G. Chen, "Advances in network controllability," *IEEE Circuits and Systems Magazine*, vol. 19, no. 2, pp. 8–32, 2019.

[3] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[4] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[5] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Communications*, vol. 4, no. 1, p. 2447, 2013.

[6] J. Gao, Y.-Y. Liu, R. M. D'Souza, and A.-L. Barabási, "Target control of complex networks," *Nature Communications*, vol. 5, no. 1, p. 5415, 2014. [Online]. Available: https://doi.org/10.1038/ncomms6415

[7] G. Li, W. Hu, G. Xiao, L. Deng, P. Tang, J. Pei, and L. Shi, "Minimum-cost control of complex networks," *New J. Phys.*, vol. 18, no. 1, p. 013012, dec 2015.

[8] A. E. Motter, "Networkcontrology," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 9, p. 097621, 2015.

[9] T. Kailath and P. Hall, *Linear Systems*, ser. Information and System Sciences Series. Prentice-Hall, 1980.

[10] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, 2014.

[11] J. Sun and A. E. Motter, "Controllability transition and nonlocality in network control," *Phys. Rev. Lett.*, vol. 110, p. 208701, May 2013.

[12] L.-Z. Wang, Y.-Z. Chen, W.-X. Wang, and Y.-C. Lai, "Physical controllability of complex networks," *Scientific Reports*, vol. 7, no. 1, p. 40198, 2017. [Online]. Available: https://doi.org/10.1038/srep40198

[13] G. Yan, J. Ren, Y.-C. Lai, C.-H. Lai, and B. Li, "Controlling complex networks: How much energy is needed?" *Phys. Rev. Lett.*, vol. 108, p. 218703, May 2012.

[14] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang, and A.-L. Barabási, "The fundamental advantages of temporal networks," *Science*, vol. 358, no. 6366, pp. 1042–1046, 2017.

[15] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[16] I. Klickstein, A. Shirin, and F. Sorrentino, "Energy scaling of targeted optimal control of complex networks," *Nature Communications*, vol. 8, no. 1, p. 15145, 2017.

[17] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Maximal aggregation of polynomial dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 114, no. 38, pp. 10 029–10 034, 2017. [Online]. Available: http://www.pnas.org/content/114/38/10029

[18] ——, "Symbolic computation of differential equivalences," in *POPL*, R. Bodík and R. Majumdar, Eds. ACM, 2016, pp. 137–150.

[19] R. Paige and R. Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.

[20] P. C. Kanellakis and S. A. Smolka, "Ccs expressions, finite state processes, and three problems of equivalence," in *Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '83, 1983, pp. 228–240.

[21] S. Derisavi, H. Hermanns, and W. H. Sanders, "Optimal state-space lumping in Markov chains," *Information Processing Letters*, vol. 87, no. 6, pp. 309 – 315, 2003.

[22] A. Valmari and G. Franceschinis, "Simple $O(m \log n)$ time markov chain lumping," in *16th International Conference Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2010, pp. 38–52.

[23] M. Aoki, "Control of large-scale dynamic systems by aggregation," *IEEE Transactions on Automatic Control*, vol. 13, no. 3, pp. 246–253, 1968.

[24] A. J. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Transactions on Automatic Control*, vol. 49, 2004.

[25] G. J. Pappas and S. Simic, "Consistent abstractions of affine control systems," *IEEE TAC*, vol. 47, no. 5, pp. 745–756, 2002.

[26] P. Tabuada and G. J. Pappas, "Abstractions of Hamiltonian control systems," *Automatica*, vol. 39, no. 12, pp. 2025–2033, 2003.

[27] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE TAC*, vol. 45, no. 6, pp. 1144–1160, 2000.

[28] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Efficient Syntax-Driven Lumping of Differential Equations," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2016, pp. 93–111.

[29] S. Tognazzi, M. Tribastone, M. Tschaikowski, and A. Vandin, "Differential equivalence yields network centrality," in *International Symposium on Leveraging Applications of Formal Methods*. Springer, 2018, pp. 186–201.
[30] S. G. Hartke and A. Radcliffe, "Mckay's canonical graph labeling algorithm," *Communicating mathematics*, vol. 479, pp. 99–111, 2009.
[31] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.
[32] G. Casadei, C. Canudas-de Wit, and S. Zampieri, "Model reduction based approximation of the output controllability gramian in large-scale networks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1778–1788, 2020.
[33] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Symbolic computation of differential equivalences," *Theor. Comput. Sci.*, vol. 777, pp. 132–154, 2019.
[34] ——, "Guaranteed error bounds on approximate model abstractions through reachability analysis," in *QEST*, vol. 11024, 2018, pp. 104–121.
[35] T. A. Davis and Y. Hu, "The university of florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, 2011.
[36] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, 2013, pp. 1343–1350. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488173
[37] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton University Press, 2011.

APPENDIX

PROOFS

Theorem 1. Let $L = L(\mathcal{H})$ be the aggregation matrix of the control equivalence $\mathcal{H}$. As in Definition 3, we assume without loss of generality that $H_1, \dots, H_k$ contain all at least one driver node, while $H_{k+1}, \dots, H_n$ have no driver nodes.

For the proof of 1), pick any $u(\cdot) \in [m; M]$ and note that

$$\begin{aligned}\partial_t Lx(t) &= L\big(Ax(t) + Bu(t)\big) = LA\bar{L}Lx(t) + LBu(t)\\ &= \hat{A}Lx(t) + LBu(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t)\\ &= \partial_t \hat{x}(t),\end{aligned}$$

where the second identity holds because $\mathcal{H}$ is a control equivalence, while the third and the last identity follow from the definition of the reduced model. Since the costs F and R are constant on $\mathcal{H}$, the above calculation yields 1).

Part 2) follows by picking any $\hat{u}(\cdot) \in [\hat{m}; \hat{M}]$, define $u(\cdot)$ using the formula in the part 2) and by reading the above calculation backwards. Specifically, the construction of $u(\cdot)$ from $\hat{u}(t)$ ensures that $\hat{B}\hat{u}(t) = LBu(t)$. □

Theorem 2. In the case $\mathcal{H}$ is a control equivalence, Theorem 1 readily implies that the reduction is optimality-preserving. To show the converse, let us assume that the reduced model underlying a given partition $\mathcal{H}$ is optimality-preserving for any final and running cost which are constant on $\mathcal{H}$. Assume without loss of generality that $H_1, \dots, H_k$ contain all at least one driver node, while $H_{k+1}, \dots, H_n$ have no driver nodes. We then fix arbitrary $T \geq 0$ and $u(\cdot) \in [m; M]$. Write $\|\cdot\|$ for the Euclidean norm and fix the final cost $F = 0$ that is obviously constant on $\mathcal{H}$ and yields $\hat{F} = 0$. Denoting by x^u the solution of the original system induced by u, we consider the running cost

$$R(t, x, v) = \|Lx - Lx^u(t)\| + \|LBv - LBu(t)\|$$

for any $v \in [m; M]$. It is immediate to see that R is constant on $\mathcal{H}$. As $R\big(t, x^u(t), u(t)\big) = 0$ for every $t \geq 0$, we have that

$$0 = \int_0^T R\big(t, x^u(t), u(t)\big)dt = J_u(x[0], T).$$

Moreover, $J_{u'}(x[0], T) \geq 0$ for any $u'(\cdot) \in [m; M]$ as R is non-negative, and so

$$0 \leq V^{\inf}(x[0], T) \leq J_u(x[0], T) = 0.$$

By assumption, $\hat{V}^{\inf}(\hat{x}[0], T) = V^{\inf}(x[0], T) = 0$. Since the optimal control is measurable by Filippov's theorem [37, Section 4.5], there thus exists $\hat{u}(\cdot) \in [\hat{m}; \hat{M}]$ such that

$$\hat{J}_{\hat{u}}(\hat{x}[0], T) = \hat{V}^{\inf}(\hat{x}[0], T) = 0.$$

Writing explicitly the reduced value functional, the above equation gives

$$\begin{aligned}0 &= \hat{J}_{\hat{u}}(\hat{x}[0], T)\\ &= \int_0^T \big(\|\hat{x}^{\hat{u}}(t) - Lx^u(t)\| + \|\hat{B}\hat{u}(t) - LBu(t)\|\big)dt,\end{aligned}$$

where $\hat{x}^{\hat{u}}(t)$ solves $\partial_t \hat{x}^{\hat{u}}(t) = \hat{A}\hat{x}^{\hat{u}}(t) + \hat{B}\hat{u}(t)$. Hence, $\hat{x}^{\hat{u}}(t) - Lx^u(t)$ and $\hat{B}\hat{u}(t) - LBu(t)$ are zero almost everywhere on $[0; T]$. This and the fundamental theorem of calculus then imply that $\hat{x}^{\hat{u}} = Lx^u$ everywhere. Moreover, there exists a sequence of points $(t_\nu)_\nu$ converging to zero that satisfies for all $\nu \geq 0$:

$$\begin{aligned}\hat{A}\hat{x}^{\hat{u}}(t_\nu) + \hat{B}\hat{u}(t_\nu) &= \partial_t \hat{x}^{\hat{u}}(t_\nu)\\ &= \partial_t Lx^u(t_\nu)\\ &= LAx^u(t_\nu) + LBu(t_\nu),\end{aligned}$$

yielding specifically $\hat{A}\hat{x}^{\hat{u}}(t_\nu) = LAx^u(t_\nu)$ for all $\nu \geq 0$. Since this yields $\hat{A}\hat{x}[0] = LAx[0]$ and $x[0]$ was chosen arbitrarily, we infer that $\mathcal{H}$ is a control equivalence. □

Theorem 3. Control equivalence can be seen as the extension of forward equivalence to control systems. Thanks to the fact that the algebraic description of control equivalence (i.e., $LA = LA\bar{L}L$) and the formula for $\hat{A}$ are as in the case of forward equivalence (and $\hat{B}$ is known), the statement readily follows from [17]. □