From Global, Finite-Time, Linear Computations to Local, Edge-Based Interaction Rules

Zak Costello and Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology
{zak.costello,magnus}@gatech.edu

***Abstract*— A network of locally interacting agents can be thought of as performing a distributed computation. But not all computations can be faithfully distributed. This paper investigates which global, linear transformations can be computed in finite time using local rules with time varying weights, i.e., rules which rely solely on information from adjacent nodes in a network. The main result states that a linear transformation is computable in finite time using local rules if and only if the transformation has positive determinant. An optimal control problem is solved for finding the local interaction rules, and simulations are performed to elucidate how optimal solutions can be obtained.**

I. Introduction

One common theme when designing control and coordination mechanisms for distributed, multi-agent systems is that the information, on which decisions are based, is restricted to be shared among agents that are adjacent in the underlying information-exchange network, e.g., [1], [2], [3], [4]. As a result, local rules are needed for processing the information and coordinating the agents in the network in such a way that some global objective is achieved. Problems that fit this description can be found in a variety of applications, including power systems [5], [6], [7], formation control [8], [9], [10], [11], [12], distributed sensor networks [13], [14], smart textiles [15], and distributed optimization [16], [17]. In this paper we take initial steps towards developing a general theory of local implementability/computability of such global behaviors.

As such, one key aspect of algorithm design is the definition of local interaction rules that produce desired global behaviors. An example of this are consensus algorithms for computing averages in a distributed manner. In fact, consensus plays a role in many different applications, including multi-agent robotics, distributed sensor fusion, and power network control, e.g., [3], [6], [18]. To this end, let the scalar state of each node in a network be $x_i \in \mathbb{R}$, with initial condition $x_i(t_0) = \xi_i, i = 1, \ldots, n$, where n is the number of nodes in the network. By stacking the states together in $x \in \mathbb{R}^n$,average consensus is achieved if

$$\lim_{t\to\infty} x(t) = \frac{1}{n}\begin{bmatrix} 1 & \ldots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \ldots & 1 \end{bmatrix}\xi, \tag{1}$$

where ξ is the vector containing all the initial node values. As such, the network is asymptotically computing the average, which is a global property since it relies on the state of every node.

In this work, we are interested in problems where networks are tasked with computing arbitrary linear transformations of the initial node states. In particular, we answer two fundamental questions: *What global, linear transformations can be computed in finite time using edge-based interaction rules? How do we find the local rules that would compute a given linear transformation?*

Some work has been done in the general area of obtaining global information with local interactions. In [19], a fixed weighting scheme was used to compute linear transformations on networks. That work has a similar aim and takes a different discrete time approach. In a certain sense, the investigation in [20] follows this line of inquiry as well. There, quadratic invariance was used to establish whether or not a convex optimization problem exists whose solution is a decentralized implementation of a centralized feedback controller. [21] further expounds on this idea and provides a practical, graph theoretic method for finding this distributed controller. Additionally, in [22] a method is presented under which consensus is computed in finite time. Our work distinguishes itself from this body of work by using a time varying weighting method, which admits the computation of global, linear transformations in finite time. Specifically, we focus on a continuous time scheme for distributed computation, over finite intervals using time-varying exogenous weight functions.

In fact, in this paper, we consider computations that are to be performed using local rules over a static and undirected information-exchange network. The local rules, once obtained, admits a decentralized implementation, where "decentralized" in this context means that each node in the network only needs to communicate state information among adjacent nodes in the network. In particular, we ask if it is possible to define local interaction laws such that $x(t_f) = T\xi$, given the linear transformation T and the initial conditions $x(t_0) = \xi$. Necessary and sufficient conditions are given for this to be possible, and they state that local interaction rules exist if and only if T has positive determinant.

II. Problem Definition

To formalize what is meant by local interactions, we first need to discuss the information-exchange network over which the interactions are defined. To this end, let V be a vertex set with cardinality n, and $E \subset V \times V$ be an edge set with cardinality m, where we insist on $(i,i) \in E,\ \forall i \in V$, as well as $(i,j) \in E \Leftrightarrow (j,i) \in E$. Let G be the graph $G = (V,E)$, where the assumptions on E imply that G is undirected and contains self-loops. We moreover assume that G is connected. As the main purpose with G is to encode adjacency information in the information-exchange network, we introduce the operator sparse(G) to capture these adjacencies, and we say that an $n \times n$ matrix $M \in$ sparse(G) if $(i,j) \notin E \Rightarrow M_{ij} = 0$.

There are a number of different ways in which local interactions can be defined. In this paper, we assume that they are given by exogenous time-varying, weights associated with the edges in the network. These weights denoted $w_{ij}(t)$ are in $L_\infty([t_0,t_f])$ where i and j indicate the originating and terminal node of the edge respectively. If $x_i \in \mathbb{R}$ is the scalar state associated with node $i \in V$, we define a local interaction as a continuous-time process

$$\dot{x}_i(t) = \sum_{j|(i,j)\in E} w_{ij}(t)x_j(t). \tag{2}$$

Note that we do not insist on $w_{ij} = w_{ji}$ even though G is undirected.

If we stack the states together in $x = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$, what we mean by *local interactions* is thus

$$\dot{x}(t) = W(t)x(t), \quad W(t) \in \text{sparse}(G), \tag{3}$$

with solution

$$x(t) = \Phi(t,t_0)x(t_0), \tag{4}$$

where Φ is the state transition matrix associated with the system in (3), e.g., [23].

The purpose of the local interactions is to perform a global, linear computation. In other words, given the $n \times n$ matrix T and the initial condition $x(t_0) = \xi$, what we would like to do is find $W(t) \in \text{sparse}(G),\ t \in [t_0,t_f]$, such that

$$x(t_f) = T\xi. \tag{5}$$

But, comparing this expression to (4), this simply means that what we would like is

$$\Phi(t_f,t_0) = T. \tag{6}$$

If this was indeed the case, then the local interactions, as defined through $W(t)$, would indeed compute $T\xi$ over the interval $[t_0,t_f]$ for all possible values of ξ, i.e., one can think of the network as a black box that takes ξ as the input at time t_0 and, at time t_f, returns $T\xi$ as the output.

As a final observation before we can formulate the general problem of performing global, linear computations using local interactions, we note that state transition matrix satisfies the same dynamics as (3), i.e.,

$$\frac{d\Phi(t,t_0)}{dt} = W(t)\Phi(t,t_0), \tag{7}$$

with initial condition $\Phi(t_0,t_0) = I$, where I is the $n \times n$ identity matrix.

Problem 1 [Local Computation]
Given a linear transformation T and a connected graph G, find $W(t) \in$ sparse$(G),\ t \in [t_0,t_f]$, such that

$$\dot{\mathbf{X}}(t) = W(t)\mathbf{X}(t), \tag{8}$$

with boundary conditions $\mathbf{X}(t_0) = I,\ \mathbf{X}(t_f) = T$.

III. On the Existence of Solutions

The main point with this paper is an exploration of what linear transformations T admit a local implementation, i.e., for what T Problem 1 has a solution. In this section, we develop necessary and sufficient conditions for this to be the case.

We start by observing that since $\mathbf{X}(t)$ is really the state transition matrix $\Phi(t,t_0)$, it is always invertible,

$$\mathbf{X}(t)^{-1} = \Phi(t,t_0)^{-1} = \Phi(t_0,t). \tag{9}$$

As a direct consequence of this, T has to be invertible for a solution to Problem 1 to exist, i.e., we need that $\det(T) \neq 0$. But, as $\mathbf{X}(0) = I$, we have that $\det(\mathbf{X}(0)) = 1 > 0$. Moreover, the determinant of a matrix depends continuously on its entries, and therefore the only way for $\det(\mathbf{X}(\tau)) < 0$ for some $\tau \in (t_0,t_f]$, there has to exist a $\tau' \in (t_0,\tau)$ such that $\det(\mathbf{X}(\tau')) = 0$. But this can not happen since $\mathbf{X}$ is always invertible. From this it directly follows that for Problem 1 to have a solution, T has to satisfy $\det(T) > 0$.

To state this fact more compactly, let $\text{GL}_+^n(\mathbb{R})$ denote the set of all $n \times n$, real matrices with positive determinant. We have thus established the following necessary condition for the existence of a solution:

Lemma 1. *A solution to Problem 1 exists only if $T \in GL_+^n(\mathbb{R})$.*

One consequence of Lemma 1 is that it is impossible to use local rules, as understood in this paper, to achieve consensus in finite time. This follows directly from the fact that the consensus computation is given by the linear map

$$T_{cons} = \frac{1}{n}\mathbf{1}^T\mathbf{1}, \tag{10}$$

where $\mathbf{1}$ is a vector of length n, with all entries equal to one. And,

$$\text{rank}(T_{cons}) = 1,$$

i.e., $\det(T_{cons}) = 0$. Note, of course, that asymptotic consensus is possible, e.g. [2], [3], [4], [10].

Now that we have established necessary conditions for Problem 1 to have a solution, we turn our attention to sufficient conditions. And, surprisingly enough, $T \in \text{GL}_+^n(\mathbb{R})$ turns out to be both necessary and sufficient for a solution to exist, which constitutes the main result in this paper:

Theorem 1. *A solution to Problem 1 exists if and only if $T \in GL_+^n(\mathbb{R})$.*

As we have already established sufficiency, what must be shown is that whenever $\det(T) > 0$, there is a $W(t) \in \text{sparse}(G)$ that drives $\mathbf{X}$ from I to T. The remainder of this section is devoted to the establishment of this fact. However, before we can give the proof to Theorem 1, a number of supporting results are needed, involving the controllability of nonlinear, drift-free systems, i.e., systems of the form

$$\dot{x} = \sum_{i=1}^{p} g_i(x)u_i, \tag{11}$$

where $x \in \mathbb{R}^n$ is the state of the system, and $u_1, \ldots, u_p \in \mathbb{R}$ are the control inputs. For the sake of easy reference, we start by recalling Chow's Theorem, as formulated in [24], for such drift-free systems:

Theorem 2 (Chow's Theorem, e.g. [24])**.** *The system in (11) is locally controllable about a point x_0 if and only if*

$$dim(\overline{\Delta}(x_0)) = n, \tag{12}$$

where $\overline{\Delta}$ is the involutive closure of the distribution span$\{g_1, \ldots, g_p\}$.

The system is moreover controllable if it is locally controllable everywhere. And, the proof that $T \in \text{GL}_+^n(\mathbb{R})$ is sufficient for Problem 1 to have a solution will hinge on showing that the dynamics, as defined through the local interaction rules in (3), is indeed controllable everywhere on $\text{GL}_+^n(\mathbb{R})$. To this end, we first must rewrite the dynamics in Problem 1 on the appropriate form. For this, we need the index matrix $\mathbb{I}_{ij} \in \mathbb{R}^{n\times n}$, which has a one at the ith row and jth column, and zeros everywhere else. The index matrix allows us to rewrite

$$\dot{\mathbf{X}} = W\mathbf{X}$$

as

$$\dot{\mathbf{X}} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} W \odot \mathbb{I}_{ij}\right)\mathbf{X}, \tag{13}$$

where the $\odot$ symbol represents element-wise matrix product, i.e.,

$$\dot{\mathbf{X}} = \left(\begin{bmatrix} w_{11} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{bmatrix} + \ldots + \begin{bmatrix} 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & w_{nn} \end{bmatrix}\right)\mathbf{X}, \tag{14}$$

where we have surpressed the explicit dependence on t for the sake of notational ease.

Rearranging the terms and letting

$$g_{ij}(\mathbf{X}) = \mathbb{I}_{ij}\mathbf{X}, \tag{15}$$

we get the drift-free matrix formulation

$$\dot{\mathbf{X}} = \sum_{i=1}^{n}\sum_{j|(i,j)\in E} g_{ij}(\mathbf{X})w_{ij}. \tag{16}$$

To clarify, $g_{ij}(\mathbf{X})$ is a matrix whose ith row contains the jth row of $\mathbf{X}$, with the rest of the elements in the matrix equal to 0,

$$g_{ij}(\mathbf{X}) = \begin{array}{c} 1 \\ \vdots \\ i-1 \\ i \\ i+1 \\ \vdots \\ n \end{array}\begin{bmatrix} 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \\ \mathbf{X}_{j1} & \ldots & \mathbf{X}_{jn} \\ 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{bmatrix}. \tag{17}$$

As a final step towards a formulation that is amenable to Chow's Theorem, let the vectorized version of g_{ij} be given by $\vec{g}_{ij} = \text{vec}(g_{ij})$, resulting in the vectorized version of (16),

$$\text{vec}(\dot{\mathbf{X}}) = \sum_{i=1}^{n}\sum_{j|(i,j)\in E} \vec{g}_{ij}(\mathbf{X})w_{ij}. \tag{18}$$

The first order of business towards establishing controllability of this system is the derivation of the Lie brackets for the system in Equation 18.

Lemma 2.

$$[\vec{g}_{ij}(\mathbf{X}), \vec{g}_{kl}(\mathbf{X})] = \begin{cases} -\vec{g}_{il}(\mathbf{X}) & \text{if } j = k,\ i \neq l \\ \vec{g}_{kj}(\mathbf{X}) & \text{if } i = l,\ j \neq k \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{19}$$

The proof to this can be found in [25], which we omit here for brevity. Now that Lie brackets can be computed in general for this problem, we must determine if the involutive closure of the distribution associated with the system in (18) contains enough independent vector fields for local controllability. To help with this determination, we provide the following lemma.

Lemma 3. *If node i is path-connected to node j, then $\vec{g}_{ij}(\mathbf{X})$ is in the distribution $\overline{\Delta}(\mathbf{X})$.*

Proof. That node i is path-connected to node j means that there is a path through adjacent nodes in the graph G that starts at node i and ends at node j. Assume that the path goes through the nodes $N_1, \ldots, N_q$, i.e., N_1 is adjacent to N_2, N_2 is adjacent to N_3, and so forth, while $N_1 = i$ and $N_q = j$. Since these nodes are adjacent, we, by definition, have that $\vec{g}_{N_1N_2}, \vec{g}_{N_2N_3}, \ldots, \vec{g}_{N_{q-1}N_q} \in \Delta(\mathbf{X})$.

The involutive closure contains every possible Lie bracket that can be recursively created from elements $\Delta(\mathbf{X})$, which implies that the problem is to create $\vec{g}_{ij}$ from some combination of Lie brackets from elements in $\Delta(\mathbf{X})$. And, from Lemma 2, we know that $[\vec{g}_{N_1N_2}, \vec{g}_{N_2N_3}]$ is equal to $-\vec{g}_{N_1N_3}$. Applying Lemma 2 again gives $[-\vec{g}_{N_1N_3}, \vec{g}_{N_3N_4}] = \vec{g}_{N_1N_4}$. This procedure can be repeated until we arrive at one of two possible cases. If q is even, the result is $[-\vec{g}_{N_1N_{q-1}}, \vec{g}_{N_{q-1}N_q}] = \vec{g}_{N_1N_q}$. If q is odd we get $[\vec{g}_{N_1N_{q-1}}, \vec{g}_{N_{q-1}N_q}] = -\vec{g}_{N_1N_q}$. In either case, we are able to construct $\vec{g}_{N_1N_q}$ from previous Lie brackets, as shown in Figure 1. And, as $N_1 = i$ and $N_q = j$, we have $\vec{g}_{ij} \in \overline{\Delta}(\mathbf{X})$. □

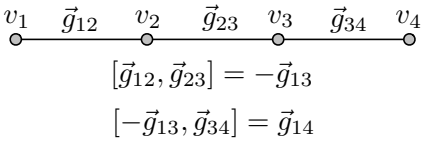

Fig. 1. An example of the construction in the proof of Lemma 3 with node i and j being represented by v_1 and v_4, respectively.

Additionally the linear independence of vector fields is needed in order to establish controllability. To this end the following lemma is presented.

Lemma 4. *$\{\vec{g}_{uv}(\mathbf{X}) \forall u, v \in V\}$ is a set of linearly independent vectors.*

Proof. By definition, $\vec{g}_{uv}(\mathbf{X}) = vec(\mathbb{I}_{uv}\mathbf{X})$. This definition can be expanded to $vec(\mathbb{I}_{uv}\mathbf{X}) = (\mathbf{X}^T \otimes I_n) vec(\mathbb{I}_{uv})$. Concatenating all possible vectors resulting from combinations of vertices,

$$(\mathbf{X}^T \otimes I_n) \begin{bmatrix} vec(\mathbb{I}_{11}) & vec(\mathbb{I}_{21}) & \dots & vec(\mathbb{I}_{nn}) \end{bmatrix} \quad (20)$$

which can be further simplified to

$$(\mathbf{X}^T \otimes I_n) I_{n^2} \quad (21)$$

Taking the determinant of this expression yields

$$det(\mathbf{X}^T \otimes I_n) = det(\mathbf{X}^T)^n \quad (22)$$

Because $\mathbf{X} \in GL_n^+(\mathbb{R})$ the determinant of X is always positive and therefore we can write

$$det(\mathbf{X}^T)^n \neq 0 \quad (23)$$

This implies that the set of vectors $\{\vec{g}_{uv}(\mathbf{X}) \forall u, v \in V\}$ is linearly independent. □

To establish that the system is controllable on $\mathrm{GL}_+^n(\mathbb{R})$, $\overline{\Delta}(\mathbf{X})$ must have rank n^2 everywhere on this set, which is the topic of the next lemma.

Lemma 5. *If G is connected then $\overline{\Delta}(\mathbf{X})$ has dimension n^2 if and only if rank$(\mathbf{X}) = n$.*

Proof. To prove this lemma, we need to show that the implication goes both ways. Assume first that $\dim(\overline{\Delta}(\mathbf{X})) = n^2$. If G is connected then, by Lemma 3 and Lemma 4 the set $\{g_{ij}(\mathbf{X}) | i, j \in V\}$ is in $\bar{\Delta}(\mathbf{X})$ and is linearly independent. Therefore,

$$\overline{\Delta}(\mathbf{X}) = \mathrm{span}\{\vec{g}_{ij},\ \forall (i,j) \in V \times V\}. \quad (24)$$

For the purpose of the proof, it is convenient to go back to the matrix formulation, and we recall that $\vec{g}_{ij} = vec(g_{ij})$. As such, we will use the matrix form g_{ij} to construct $\mathbf{X}$. And, since the goal is to form a matrix with rank n, only n linearly independent matrices are needed. So, we arbitrarily choose to form $\mathbf{X}$ from the "diagonal" set $\{g_{11}, g_{22}, \dots, g_{nn}\}$. Using the fact that $g_{ij} = \mathbb{I}_{ij}\mathbf{X}$, we can write,

$$\sum_{i=1}^{n} g_{ii} = \sum_{i=1}^{n} \mathbb{I}_{ii}\mathbf{X},$$

which simplifies to

$$\sum_{i=1}^{n} g_{ii} = \mathbf{X}. \quad (25)$$

g_{ii} is a matrix with one nonzero row at row i. The nonzero rows of each g_{ii} are linearly independent. And, since $\mathbf{X}$ is composed of n linearly independent rows, $\mathrm{rank}(\mathbf{X}) = n$, and the first implication follows. Next, we must show that

$$\mathrm{rank}(\mathbf{X}) = n \Rightarrow \dim(\overline{\Delta}(\mathbf{X})) = n^2, \quad (26)$$

which we do by contradiction. Using the expression $g_{ij} = \mathbb{I}_{ij}\mathbf{X}$, n^2 matrices can be formed from $\mathbf{X}$. Let us assume that they are not linearly independent. This implies that there exists a set of coefficients α_{ij} such that, for some (k, l),

$$\sum_{(i,j) \neq (k,l)} g_{ij}\alpha_{ij} = g_{kl}. \quad (27)$$

Since $\mathbf{X}$ has full rank, $\mathbf{X}$ can be removed from (27) based on the fact that $g_{ij} = \mathbb{I}_{ij}\mathbf{X}$, yielding

$$\sum_{(i,j) \neq (k,l)} \mathbb{I}_{ij}\alpha_{ij} = \mathbb{I}_{kl}. \quad (28)$$

By definition of the index matrix, (28) cannot be true, since every matrix in the sum on the left has a value of zero where $\mathbb{I}_{kl}$ has value of 1. Therefore, we have reached a contradiction and can conclude that $\dim(\overline{\Delta}(\mathbf{X})) = n^2$. □

Since $\mathbf{X}$ is really a state transition matrix, i.e., it is indeed invertible (with $\mathrm{rank}(\mathbf{X}) = n$), the system in (16) is locally controllable everywhere on $\mathrm{GL}_+^n(\mathbb{R})$ as long as the underlying graph G is connected:

Theorem 3. *The system*

$$\dot{\mathbf{X}} = W\mathbf{X},\ W \in \mathit{sparse}(G)$$

is locally controllable everywhere on $GL_+^n(\mathbb{R})$ if G is connected.

Theorem 3 and Lemma 1 give us all the ammunition needed to prove the main result in this paper, namely Theorem 1:

Proof of Theorem 1. Lemma 1 tells us that a solution only exists if $T \in \mathrm{GL}_+^n(\mathbb{R})$, so what remains is to establish that this is indeed sufficient. Hence, assume that $T \in \mathrm{GL}_+^n(\mathbb{R})$. Since $I \in \mathrm{GL}_+^n(\mathbb{R})$, and $\mathrm{GL}_+^n(\mathbb{R})$ is connected [26], there is a continuous curve of matrices in $\mathrm{GL}_+^n(\mathbb{R})$ that connects I and T. And, by Theorem 3, every point along the path connecting I and T is locally controllable. The system being drift-free moreover implies that it can flow along this curve, e.g., [27]. Therefore, a solution to Problem 1 exists if $T \in \mathrm{GL}_+^n(\mathbb{R})$. □

If we return to the consensus problem, we have already established that T_{cons} in (10) is not computable in finite time

using local rules. However, consider instead the transformation

$$T_{cons2} = \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \quad (29)$$

We have

$$\det(T_{cons2}) = \frac{1}{n} \quad (30)$$

and, as such, it is computable using local rules. In this case, the network average is only computed by a single node (node 1 in this case), while the remaining nodes return to their initial values at the end of the computation. This can in fact be generalized to any scalar, non-zero, linear map $\ell : \mathbb{R}^n \to \mathbb{R}$ through

$$T_\ell \xi = \begin{bmatrix} \ell(\xi) \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix},$$

where we have assumed that $\ell(\xi)$ depends on ξ_1.[1] The point with this is that *it is possible to compute any scalar, non-zero, linear map as long as the computation only has to take place at a single node.*

IV. A Numerical Example

Just because we know that a computation $T\xi$ can be done using local rules it does not follow that we can (easily) find these rules, encoded through $W(t) \in \text{sparse}(G)$, such that $\dot{\mathbf{X}} = W\mathbf{X},\ \mathbf{X}(t_0) = I, \mathbf{X}(t_f) = T$. There are many possible ways in which weight functions can be found. In this section, we address this problem in the context of optimal control. The optimal control method detailed below was chosen simply to illustrate that solutions can be found in support of Theorem 1 and not for the purpose of providing an efficient or scalable numerical method. In fact, we consider the cost

$$J(W) = \int_0^{t_f} \frac{1}{2} \|W(t)\|_F^2 dt, \quad (31)$$

where $\|\cdot\|_F$ is the Frobenius norm. The resulting constrained minimization problem becomes

Problem 2 [Optimal Local Interactions]

$$\min_W J(W) = \int_0^{t_f} \frac{1}{2} \|W(t)\|_F^2 dt \quad (32)$$

such that

$$\begin{aligned} &\dot{\mathbf{X}} = W\mathbf{X} \\ &W(t) \in \textit{sparse}(G),\ \forall t \in [t_0, t_f] \\ &\mathbf{X}(t_0) = I,\ \mathbf{X}(t_f) = T. \end{aligned} \quad (33)$$

[1]If not, simply pick another node in the network that ξ does depend on, as the node where the computation takes place.

Using the maximum principle, the local solution is found by solving the two point boundary problem numerically

$$\begin{aligned} \dot{\mathbf{X}}_{ij} &= -\sum_{k|(i,k)\in E} \mathbf{X}_{kj} \sum_{l=1}^{n} \lambda_{il} \mathbf{X}_{kl} \\ \mathbf{X}(t_0) &= I,\ \mathbf{X}(t_f) = T \\ \dot{\lambda}_{ij} &= \sum_{k|(i,k)\in E} \lambda_{kj} \sum_{l=1}^{n} \lambda_{kl} \mathbf{X}_{il}, \end{aligned} \quad (34)$$

where $\lambda_{ij}\ i, j = 1, \ldots, n$ are costates. Numerical solutions for the weight functions were found in both examples by solving (34) using test shooting. A reference which explains this method in detail is [28]. Using optimal control to find weight functions was simply a convenient method for illustrating the feasibility of finding solutions.

A. Swapping Node Values

Consider the situation when the linear transformation T represents a reordering (or swapping) of states. We examine the 4 node case where the underlying graph topology is given by nodes $2, 3, 4$ forming a clique (fully connected subgraph) and node 1 is connected to node 2. In this example agents 1 and 2 and agents 3 and 4 are to "swap" state values, the transformation matrix becomes

$$T_{swap} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (35)$$

However, the linear interpolation between I and T_{swap} contains a singular matrix, which makes the two-point boundary problem numerically ill-conditioned when using shooting methods, e.g., [28]. There are many such choices of transformations where this ill-conditioning is a concern, as discussed in [29]. A way around this problem is to avoid this singular matrix by solving two sequential two-point boundary problems.

As an example, in the first iteration, we let the boundary conditions be $\mathbf{X}(t_0) = I,\ \mathbf{X}((t_f - t_0)/2) = T_1$. For the second iteration, they are $\mathbf{X}((t_f - t_0)/2) = T_1,\ \mathbf{X}(t_f) = T_{swap}$, where

$$T_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (36)$$

This sequential approach avoids the numerical ill-conditioning, and the solution is shown in Figures 2 - 3.

V. Conclusions

In this paper, a step was taken towards computing arbitrary global functions on networks with local interaction rules. In particular, it presented a method which allows a networked system to compute global, linear transformations using only local rules.

We derived necessary and sufficient conditions under which it is possible to use a distributed, time-varying weighting scheme to compute the transformation T for undirected,

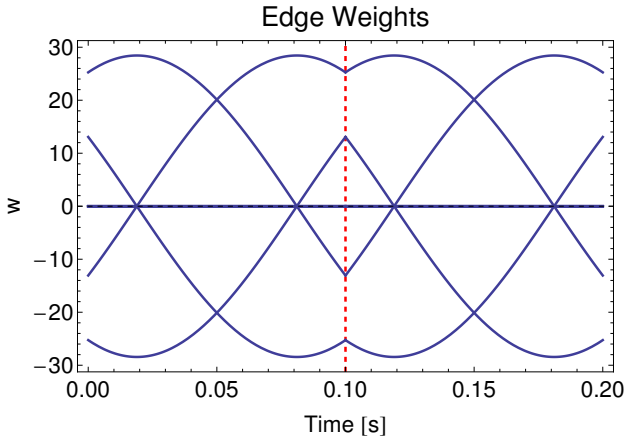

Fig. 2. The weight functions define the local interactions needed to achieve the swap in the 4-node case. The first and second subproblems are solved over the time intervals $[0, 0.1)$ and $[0.1, 0.2]$ respectively.

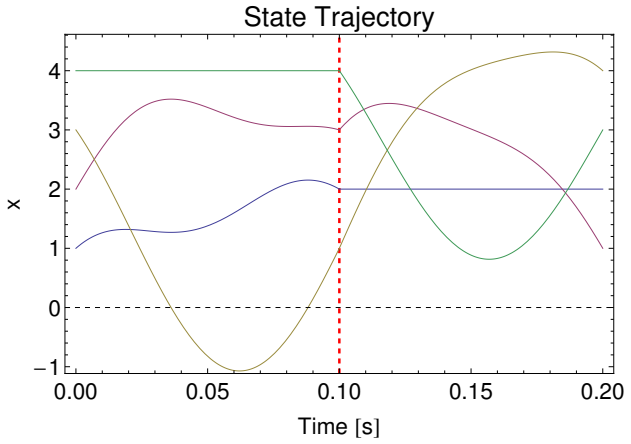

Fig. 3. The evolution of the node states for the swap problem. The initial state is $x(t_0) = [1, 2, 3, 4]^T$ and the final state is $x(t_f) = [2, 1, 4, 3]^T$, i.e., the first and second states swapped values and the third and fourth state swapped values.

connected networks with fixed topology. Specifically, we showed that the necessary and sufficient condition for T to be locally computable is that it has positive determinant, i.e., $T \in \mathrm{GL}^n_+(\mathbb{R})$.

Acknowledgment

This work was sponsored in part by a grant from the US Air Force Office for Sponsored Research. The authors would like to thank Professor Mark Costello at the Georgia Institute of Technology for his advice regarding this work.

References

[1] F. Bullo, J. Cortes, and S. Martnez, *Distributed Control of Robotic Networks. A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[2] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer-Verlag, 2008.

[5] A. L. Dimeas and N. D. Hatziargyriou, "Operation of a multiagent system for microgrid control," *Power Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 1447–1455, 2005.

[6] T. Ramachandran, Z. Costello, P. Kingston, S. Grijalva, and M. Egerstedt, "Distributed power allocation in prosumer networks," in *IFAC Necsys*, 2012.

[7] S. Grijalva, M. Costley, and N. Ainsworth, "Prosumer-based control architecture for the future electricity grid," in *IEEE Multi-Conference on Systems and Control*, 2011.

[8] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[9] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[11] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part II : Dynamic topology," in *Proc. 42nd IEEE Conf. Decision Control*, 2003.

[12] N. Michael and V. Kumar, "Controlling shapes of ensembles of robots of finite size with nonholonomic constraints," in *RSS*, 2008.

[13] K. Romer and F. Mattern, "The design space of wireless sensor networks," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 54–61, 2004.

[14] F. Zhang and N. Leonard, "Coordinated patterns of unit speed particles on a closed curve," *Systems and Control Letters*, vol. 56, no. 6, pp. 397–407, 2007.

[15] D. Marculescu, R. Marculescu, N. H. Zamora, P. Stanley-Marbell, P. K. Khosla, S. Park, S. Jayaraman, S. Jung, C. Lauterbach, W. Weber, *et al.*, "Electronic textiles: A platform for pervasive computing," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 1995–2018, 2003.

[16] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.

[17] A. Nedic, A. Ozdaglar, and A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[18] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 6698–6703.

[19] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 650–660, 2008.

[20] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 274–286, 2006.

[21] J. Swigart and S. Lall, "A graph-theoretic approach to distributed control over networks," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 5409–5414.

[22] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen, "Graph diameter, eigenvalues, and minimum-time consensus," *Automatica*, 2013.

[23] R. Brockett, *Finite Dimensional Linear Systems*. John Wiley & Sons, Inc., 1970.

[24] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer New York, 1999, vol. 10.

[25] Z. Costello and M. Egerstedt, "From global, finite-time, linear computations to local, edge-based interaction rules," *ArXiv.org*, 2014, a preprint is available at arXiv.

[26] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1993.

[27] R. W. Brockett, "System theory on group manifolds and coset spaces," *SIAM Journal on Control*, vol. 10, no. 2, pp. 265–284, 1972.

[28] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.

[29] A. Bhaya, "Real matrices with positive determinant are homotopic to the identity," *SIAM review*, vol. 40, no. 2, pp. 335–340, 1998.