Estimation and Registration on Graphs

Stephen D. Howard, Douglas Cochran, William Moran, and Frederick R. Cohen

arXiv:1010.2983v1 [cs.NI] 14 Oct 2010

Abstract—A statistical framework is introduced for a broad class of problems involving synchronization or registration of data across a sensor network in the presence of noise. This framework enables an estimation-theoretic approach to the design and characterization of synchronization algorithms. The Fisher information is expressed in terms of the distribution of the measurement noise and standard mathematical descriptors of the network's graph structure for several important cases. This leads to maximum likelihood and approximate maximum-likelihood registration algorithms and also to distributed iterative algorithms that, when they converge, attain statistically optimal solutions. The relationship between optimal estimation in this setting and Kirchhoff's laws is also elucidated.

I. INTRODUCTION

Registration of data across a network is an ubiquitous problem in distributed sensing. Over more than three decades, much effort has been expended on development of algorithms to provide time synchronization across a distributed network; e.g., [1]–[5]. Synchronization of this kind is important for distributed parallel processing as well as data fusion across a sensor network. It is typically the case that the network is not complete; i.e., each node does not communicate with every other node. A large fraction of algorithms described in the literature produce algorithms to minimize an error or objective function based on least squares, often within power or other resource constraints. Leaving aside the latter issue, the problem in this setting is to assign a clock adjustment to every node based on knowledge of the clock differences, generally noisy, between some pairs of nodes in the network. Even if clock difference measurements are available for every pair of nodes in the network, the presence of noise still raises consistency considerations; e.g., the true offsets must sum to zero around any closed cycle.

The domain of practical network synchronization problems is by no means limited to clock offsets, nor is the natural measurement space restricted to the real line. Individual nodes may possess multiple data to be registered cross the network, and the noise affecting such vector data may be correlated across its components. Further, the natural measurement space is often not a linear space. In phase synchronization, for

example, typical data could be measurements of the phase differences between local oscillators at the nodes. In this case, the natural measurement space is the circle $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$ rather then the real line $\mathbb{R}$. If several local oscillators are involved, measurements might lie on the torus $\mathbb{T}^n$. Another important practical example where the measurement space is a nonlinear multi-dimensional manifold is registration of local coordinate systems, for which the natural setting is the special orthogonal group $SO(3)$. Even in the context of clocks, if both offset and clock speed are adjustable locally the offsets are elements of the affine group $\mathbb{A}$. These examples illustrate that practical problems can entail data on Lie groups that are compact (e.g., $\mathbb{T}$ or $SO(3)$), non-compact ($\mathbb{R}^n$), abelian ($\mathbb{R}^n$, $\mathbb{T}$), or non-abelian ($SO(3)$ or $\mathbb{A}$).

It is common to represent networks by graphical models. In this setting, the network nodes that provide data to be registered or synchronized are represented by vertices labeled with their associated parameters, such as local clock time or local coordinate system. Each pair of vertices corresponding to a pair of nodes that are in direct communication are joined by an edge. Information is shared between vertices along such edges, each of which is labeled by a noisy measurement of the difference between the parameters at the end vertices of the edge. The notion of difference between two parameter values depends on the algebraic structure of the parameter space. In $\mathbb{R}^n$, for example, it is defined by subtraction of vectors. In other spaces, which are assumed to be Lie groups, difference is defined in terms of the group operation. The estimation problem on which this paper focuses arises precisely because the difference values that label the edges are corrupted by noise in a sense that will be made precise later in the paper. As a consequence of this corruption, the edge labels in any graph with cycles will generally be inconsistent; i.e., the edge labels around closed cycles will not sum to zero, even though the true difference values must do so. For a connected graph, the desired synchronization should provide a set of consistent edge labels that can be used to determine a unique offset value to register any pair of vertices in the graph. If one vertex label is known, this is equivalent to assigning labels to all other vertices consistently throughout the graph.

The first goal of this paper is to frame a class of network synchronization problems encompassed by the graphical model just described in terms of statistical estimation theory. The second goal is to derive and characterize maximum-likelihood estimators or approximations thereof for a significant subclass class of these problems, together with local algorithms that realize these estimators. Specifically, these estimation problems entail estimation of the vertex labels (parameters) from the edge labels (noisy data). This paper focuses upon two cases in which the noise models are classical: Gaussian noise on $\mathbb{R}^d$ and von Mises distributed noise on $\mathbb{T}$.

Between them, these cases encompass most of the elements encountered in the situation where the measurements and noise are on abelian Lie groups, which is the most general setting addressed in detail in this paper. The non-abelian situation, which presents additional mathematical challenges, will be addressed in a sequel.

In each form of the estimation problem treated here, the Fisher information and maximum likelihood estimator are derived in a closed forms that depend both on the distribution of the noise and on the structure of the graph in intuitively appealing ways. In each of these cases, it is observed that the maximum likelihood estimator is non-local in the sense that the offsets for a node relative to its neighbors cannot be estimated at that node using only information obtained from its neighbors. Nevertheless, it is possible to find iterative (gradient descent) algorithms that are local and do converge to the ML estimator when they converge, which happens very frequently in empirical tests.

Interesting problems associated with local maxima of the likelihood function that arise in the setting of compact Lie groups are discussed in connection with the circle case treated here. The value of the Fisher information and functions thereof (e.g., its determinant) in designing networks that support accurate registration is also discussed.

Structurally, Section II begins with a synopsis of the essential concepts of graph theory and the associated notation that will be used throughout the paper. Some relevant introductory material on Lie groups is also summarized in this section, and the mathematical framework for discussion of the abelian Lie group is set forth. Section III treats the case of Gaussian noise on $\mathbb{R}^d$ and Section IV subsequently describes and analyzes the performance of local algorithms for this case. Section V covers the case of von Mises noise on $\mathbb{T}$, followed by description and performance analysis for local algorithms for this situation and a discussion of critical points of the likelihood function. The general abelian Lie group case is presented in Section VI, where it is shown to follow essentially the same pattern as the special cases presented earlier. Finally, some preliminary remarks on the important non-abelian Lie group setting appears in Section VII together with some concluding remarks.

II. Graph Theory Preliminaries

The aspects of elementary graph theory needed in the remainder of this paper are covered in many standard references, such as [6]. In what follows, a graph Γ will be assumed to have a finite vertex set $V(\Gamma)$. The collection of edges in Γ will be denoted by $E(\Gamma)$, where the edge $e = (u, v)$ joins vertices u and v. If the graph is directed then edge (u, v) starts at u and ends at v. If the graph is not directed then $(u, v) \equiv (v, u)$. Unless otherwise noted, graphs in this paper will be directed, and the values labeling the edges will correspond to the difference between the label on the vertex at the end of the edge and the label on the vertex at the start of the edge. At several places in the paper, however, the graph orientation is irrelevant.

In the following definitions of spaces of functions on $V(\Gamma)$ and $E(\Gamma)$, the functions are assumed to be real-valued, as in the clock synchronization example. Later these definitions will be extended to cover other possible parameter spaces. The *vertex space* $C_0(\Gamma)$ of a graph Γ is the real vector space of functions $V(\Gamma) \to \mathbb{R}$; elements of $C_0(\Gamma)$ are vectors of real numbers indexed by the vertices. Similarly the *edge space* $C_1(\Gamma)$ is the real vector space of functions $E(\Gamma) \to \mathbb{R}$. The number of vertices in Γ will be denoted by $|V(\Gamma)| = n$ and the number of edges by $|E(\Gamma)| = m$. It will be convenient to fix an ordering on the sets V and E in order to construct bases for $C_0(\Gamma)$ and $C_1(\Gamma)$. Denoting $V(\Gamma) = \{v_1, \ldots, v_n\}$, a basis for $C_0(\Gamma)$ is obtained by defining functions $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ according to

$$\boldsymbol{v}_i(v_j) = \delta_{ij}, \quad i, j = 1, \ldots, n \qquad (1)$$

so that any element of $C_0(\Gamma)$ can be written

$$\boldsymbol{x} = \sum_{i=1}^{n} x_i \boldsymbol{v}_i$$

$C_0(\Gamma)$ is an inner product space with inner product $\langle \cdot, \cdot \rangle_{C_0}$ such that $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle_{C_0} = \delta_{ij}$, so that $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ becomes an orthonormal basis of the real inner product space. The linear map $A : C_0(\Gamma) \to C_0(\Gamma)$ defined by

$$A\boldsymbol{v}_\ell = \sum_{v_j \sim v_\ell} \boldsymbol{v}_j$$

where $v_j \sim v_\ell$ indicates that there is an edge connecting v_j and v_ℓ, is called the (undirected) *adjacency* map. The corresponding matrix in the $\boldsymbol{v}_\ell$ basis is called the adjacency matrix.

The degree d_v of a vertex v is the number of edges either starting or ending at v, and the linear map $N : C_0(\Gamma) \to C_0(\Gamma)$ defined by

$$N\boldsymbol{v}_\ell = d_{v_\ell}\boldsymbol{v}_\ell,$$

is called the degree map. The corresponding matrix in the $\boldsymbol{v}_j$ basis is called the degree matrix and is diagonal with the degrees of the vertices of Γ on its main diagonal. The Kirchhoff map (unnormalized Laplacian) of Γ is defined by

$$L = N - A \ : \ C_0(\Gamma) \to C_1(\Gamma).$$

L is positive semidefinite and the dimension of its zero eigenspace (null space) is the number of connected components of Γ.

Indexing the edges of Γ as $E(\Gamma) = \{e_1, \cdots, e_m\}$, a basis for $C_1(\Gamma)$ can be constructed in similar fashion to the one for $C_0(\Gamma)$. Specifically, define functions $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ by

$$\boldsymbol{e}_i(e_j) = \delta_{ij}, \quad i, j = 1, \ldots, m \qquad (2)$$

Then any element of $C_1(\Gamma)$ can be written

$$\boldsymbol{x} = \sum_{i=1}^{n} x_i \boldsymbol{e}_i$$

Defining an inner product $\langle \cdot, \cdot \rangle_{C_1}$ such that $\langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle_{C_1} = \delta_{ij}$ makes $C_1(\Gamma)$ into an inner product space. Linear *source* and *target* maps, respectively $s, t : C_1(\Gamma) \to C_0(\Gamma)$ are defined by

$$s(\boldsymbol{e}_j) = \boldsymbol{u} \quad \text{and} \quad t(\boldsymbol{e}_j) = \boldsymbol{v}$$

where $e_j = (u, v)$. Finally the (directed) *incidence* (or boundary) map of Γ is $D : C_1(\Gamma) \to C_0(\Gamma)$, where $D = t - s$. Certain concepts will be needed from homology theory, and will be introduced informally as required. The map D is, in homological terms, a *boundary operator*, which applied to any edge $e = (u, v)$ gives

$$D(e) = t(e) - s(e) = v - u.$$

The Kirchhoff map can be written in terms of the incidence map as $L = DD^\mathsf{T}$, where the adjoint map $D^\mathsf{T} : C_0(\Gamma) \to C_1(\Gamma)$ is the *coboundary* operator defined by

$$D^\mathsf{T}(v) = \sum_{e_j : t(e_j) = v} e_j - \sum_{e_j : s(e_j) = v} e_j.$$

The *cycle space* $Z(\Gamma)$ is defined as follows. A *cycle* is a closed path in Γ; that is, a sequence of vertices $\mathfrak{L} = v_1 v_2 v_3 \ldots v_q$ in Γ where v_i is adjacent to v_{i+1} for $i = 1, \ldots q$ and v_q is adjacent to v_1. The corresponding element of $Z(\Gamma)$ is a function $z_\mathfrak{L} \in C_1(\Gamma)$ given by

$$z_\mathfrak{L}(e_j) = \begin{cases} 1 & \text{if } e_j \in \mathfrak{L} \text{ and } e_j \text{ is oriented as } \mathfrak{L} \\ -1 & \text{if } e_j \in \mathfrak{L} \text{ and } e_j \text{ is oriented opposite to } \mathfrak{L} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$Z(\Gamma)$ is the linear subspace of $C_1(\Gamma)$ spanned by the $z_\mathfrak{L}$ as $\mathfrak{L}$ runs over all cycles in Γ. The cycle space is exactly the kernel of D; that is, for all $z \in Z(\Gamma)$,

$$Dz = 0, \tag{4}$$

and every $z \in C_1(\Gamma)$ satisfying (4) is a linear combination of $z_\mathfrak{L}$. This condition implies that, for $z \in Z(\Gamma)$, the oriented sum of the values on the set of edges meeting at any of the vertices is zero; i.e.,

$$\sum_{e_j : t(e_j) = v} z_j = \sum_{e_j : s(e_j) = v} z_j, \tag{5}$$

for all $v \in V(\Gamma)$. This, of course, is a statement of *Kirchhoff's current law*. For a graph with k connected components the dimension of $Z(\Gamma)$ is the first Betti number of the graph; i.e., $\dim Z(\Gamma) = m - n + k$.

A second subspace of $C_1(\Gamma)$ that arises in the development to follow is the cocycle space (cut set) $U(\Gamma)$, an element of which is defined by fixing a partition $\mathfrak{P}$ of the vertex set $V(\Gamma)$ into two disjoint sets; i.e., $V(\Gamma) = V_1 \cup V_2$. With respect to this partition, define a vector $\omega_\mathfrak{P} \in U(\Gamma)$ to be

$$\omega_\mathfrak{P}(e_j) = \begin{cases} 1 & \text{if } e_j \text{ joins } V_1 \text{ to } V_2 \\ -1 & \text{if } e_j \text{ joins } V_2 \text{ to } V_1 \\ 0 & \text{otherwise} \end{cases}$$

$U(\Gamma)$ is the linear subspace of $C_1(\Gamma)$ spanned by the $\omega_\mathfrak{P}$ as $\mathfrak{P}$ runs over all partitions of $V(\Gamma)$.

The cocycle space is exactly the orthogonal complement of $Z(\Gamma)$ in $C_1(\Gamma)$. Thus, for any $z \in Z(\Gamma)$ and any $\omega \in U(\Gamma)$,

$$\langle z, \omega \rangle_{C_1} = 0 \tag{6}$$

This implies that every $\omega = \sum_j \omega_j e_j \in U(\Gamma)$ satisfies

$$\sum_{e_j \in \mathfrak{L}} (-1)^{\sigma_j} \omega_j = 0,$$

for all cycles $\mathfrak{L}$, where $\sigma_j = 0$ if e_j is oriented as $\mathfrak{L}$ and $\sigma_j = 1$ if e_j is oriented opposite to $\mathfrak{L}$. This is *Kirchhoff's voltage law*. Furthermore, every vector $\omega \in C_1(\Gamma)$ can be uniquely decomposed as

$$x = \omega + z, \quad \omega \in U(\Gamma) \text{ and } z \in Z(\Gamma)$$

In other words, $C_1(\Gamma) = Z(\Gamma) \oplus U(\Gamma)$. It is customary in the mathematical literature (e.g., [7]) not to choose a basis to identify the $C_i(\Gamma)$ ($i = 0, 1$) with their duals and to regard boundary and coboundary maps as being on different spaces. It is convenient here to identify them.

The cocycle space $U(\Gamma)$ is also the image of $C_0(\Gamma)$ under the coboundary operator, $U(\Gamma) = \text{Im}(D^\mathsf{T})$. The kernel of D^T is the space of locally constant functions on $V(\Gamma)$; i.e., functions that are constant on the vertices of connected components. In the development to follow, there is no loss of generality in working on different connected components separately. So from this point Γ will be assumed to be connected, and hence the kernel of D^T is $\text{span}\{\mathbf{1}\}$ where $\mathbf{1}$ denotes the unit constant function on $V(\Gamma)$. With this connectedness assumption, every $x \in C_0(\Gamma)$ can be decomposed uniquely as

$$x = D\omega + \alpha\mathbf{1}, \tag{7}$$

where $\omega \in U(\Gamma)$ and $\alpha \in \mathbb{R}$. Formally, this decomposition is orthogonal since the kernel of D^T is orthogonal to the image of D.

A spanning tree S in Γ is a graph with $V(S) = V(\Gamma)$ such that every pair of vertices is joined by exactly one path in S. Equivalently, S is a maximal subtree of Γ. Kirchhoff's matrix tree theorem implies that the number of spanning trees $t(\Gamma)$ in Γ is equal to the absolute value of any cofactor of $L(\Gamma)$; i.e., $t(\Gamma)$ is the modulus of the product of the $n - 1$ largest eigenvalues of $L(\Gamma)$.

In graph theory, one often encounters maps $W : C_1(\Gamma) \to C_1(\Gamma)$ that describe some weighting on the edges of Γ. In the standard basis, such a map has diagonal matrix, and a weighted Laplacian can be defined by $\tilde{L} = DWD^\mathsf{T}$. The matrix tree theorem can be generalized to state that the absolute value of any cofactor of $\tilde{L}$ is equal to

$$\sum_S \prod_{e \in S} W(e)$$

where the sum extends over all spanning trees of Γ.

III. Gaussian Noise on $\mathbb{R}^d$

This section provides precise formulations of estimation problems that arise in connection with registration on networks in situations where the parameter values at each node are real numbers or vectors in $\mathbb{R}^d$. The measurements of differences between parameter values at communicating nodes are corrupted by zero-mean additive Gaussian noise. The one-dimensional problem with independent noise on each measurement is treated first. The value of explicit expressions

for the Fisher information, its determinant, and maximum-likelihood estimators obtained for this case in network design problems is discussed. This is followed by treatment of the multi-dimensional problem, in which both correlated and independent noise are considered.
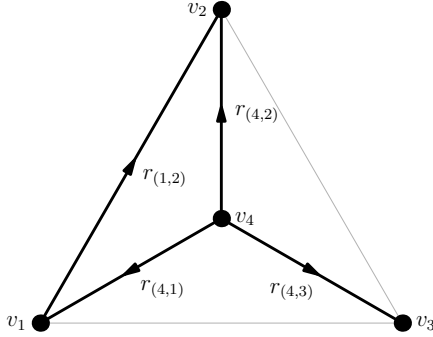

Fig. 1. Each edge e of a directed graph Γ is labeled with a value r_e that is the difference ω_e of labels at its boundary vertices plus a noise value ε_e.

A. One-dimensional Gaussian problem

The situation in which the parameter space is the real line and the differences between communicating nodes are corrupted by zero-mean additive Gaussian noise is developed here in detail because it illustrates the approach that will be used in the more complicated cases that follow. In this setting, the data vector $\boldsymbol{r} \in C_1(\Gamma)$, is a sum of a vector of true difference values and noise; i.e.,

$$\boldsymbol{r} = \boldsymbol{\omega} + \boldsymbol{\varepsilon} \qquad (8)$$

where $\boldsymbol{r}$ and $\boldsymbol{\varepsilon}$ are in $C_1(\Gamma)$. Because the true difference values must sum to zero around any cycle, $\boldsymbol{\omega} \in U(\Gamma)$. For the moment, assume that $\boldsymbol{\varepsilon}$ is jointly normal with covariance matrix $\sigma^2 \mathbb{I}$; i.e., that random variables ε_i are independent and identically distributed (iid) with variance σ^2. With this assumption, the conditional probability density function for $\boldsymbol{r}$ given $\boldsymbol{\omega}$ is then

$$
\begin{aligned}
p(\boldsymbol{r}|\boldsymbol{\omega}) &= \prod_{e_j \in E(\Gamma)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(r_j - \omega_j)^2\right) \\
&= (2\pi\sigma^2)^{-|E(\Gamma)|/2} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{r} - \boldsymbol{\omega}\|_{C_1}^2\right)
\end{aligned}
$$

and the log-likelihood function is thus

$$\ell(\boldsymbol{r}|\boldsymbol{\omega}) = -\frac{1}{2\sigma^2}\|\boldsymbol{r} - \boldsymbol{\omega}\|_{C_1}^2 + \text{constant}$$

The ML estimator minimizes $\|\boldsymbol{r} - \boldsymbol{\omega}\|_{C_1}$, and is hence given by

$$\hat{\boldsymbol{\omega}} = \Pi_U \, \boldsymbol{r} \qquad (9)$$

where Π_U denotes orthogonal projection into $U(\Gamma)$ with respect to the inner product on C_1. The residual $\boldsymbol{r} - \hat{\boldsymbol{\omega}}$ then resides in the cycle space $Z(\Gamma)$. This result provides a useful characterization of the ML estimator: the estimate satisfies Kirchhoff's voltage law and the residual satisfies Kirchhoff's current law.

To explicitly compute (9), it is desirable to parametrize the space $U(\Gamma)$, whose definition (6) does not well elucidate the nature of its elements. Two basic parameterizations are useful. In the first, a particular vertex is chosen as a reference. Then (7) implies that the value of $\boldsymbol{\omega} \in U(\Gamma)$ is determined by the relative offsets of the other $|V(\Gamma)| - 1 = n - 1$ vertices, denoted by W. The second parameterization of $U(\Gamma)$ is obtained by choosing spanning tree $S \in E(\Gamma)$ of Γ and noting that if $\boldsymbol{\omega}$ is known on S, then all $n - 1$ offsets relative to a reference vertex can be determined by following the tree.

In the first of these cases, the basis given above for $\text{span}(W)$ is chosen and $\boldsymbol{x} \in \text{span}(W)$ may be expressed as

$$\boldsymbol{x} = \sum_{j=1}^{n-1} x_j \boldsymbol{v}_j.$$

Alternatively, for the fixed spanning tree S, one may write $\boldsymbol{\nu} \in \mathcal{S} = \text{span}(S)$ as

$$\boldsymbol{\nu} = \sum_{j=1}^{n-1} \nu_j \boldsymbol{e}_{\rho(j)}$$

where the $\rho(j)$ label the $n - 1$ edges comprising the spanning tree S.

These representations enable the definition of the $(n-1) \times m$ matrix D_W and the $(n-1) \times (n-1)$ matrix D_{WS} with entries

$$
\begin{aligned}
[D_W]_{ij} &= \langle \boldsymbol{v}_i, D\boldsymbol{e}_j \rangle_{C_0} \\
[D_{WS}]_{ij} &= \langle \boldsymbol{v}_i, D\boldsymbol{e}_{\rho(j)} \rangle_{C_0},
\end{aligned}
$$

$[D_{WS}]$ is the matrix of the restriction of D to $\text{span}(S)$. With these definitions,

$$\boldsymbol{\omega} = D_W^\mathsf{T} \boldsymbol{x} \qquad (10)$$

and

$$\boldsymbol{\nu} = D_{WS}^\mathsf{T} \boldsymbol{x}. \qquad (11)$$

D_{WS} is invertible; in fact [8, p. 101],

$$\det D_{WS} = \pm 1. \qquad (12)$$

Taking inverses in (11) yields

$$\boldsymbol{x} = \left(D_{WS}^\mathsf{T}\right)^{-1} \boldsymbol{\nu} = {D_{WS}^\mathsf{T}}^{-1} P_S \boldsymbol{\omega}$$

where $[P_S]_{ij} = \delta_{\rho(i),j}$ is the matrix of the orthogonal projection onto $\mathcal{S}$. The matrix

$$L_W = D_W D_W^\mathsf{T}$$

is the Laplacian (Kirchhoff) matrix with the row and column corresponding to v_n removed.

With this notation, the maximum-likelihood estimates for $\boldsymbol{x}$ and $\boldsymbol{\nu}$ are then

$$\hat{\boldsymbol{x}} = L_W^{-1} D_W \boldsymbol{r} \qquad (13)$$

$$\hat{\boldsymbol{\nu}} = D_{WS}^\mathsf{T} L_W^{-1} D_W \boldsymbol{r}$$

and the estimate of $\boldsymbol{\omega}$ is

$$\hat{\boldsymbol{\omega}} = D_W^\mathsf{T} L_W^{-1} D_W \boldsymbol{r}$$

The Fisher information matrix for estimation of the offsets x_j is

$$F^W = -\mathsf{E}\left\{\boldsymbol{\nabla}_{\boldsymbol{x}}^2 \log p(\boldsymbol{r}|\boldsymbol{\omega})\right\}$$
$$= \frac{1}{\sigma^2}\left(\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega}\right)^{\mathsf{T}}\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega}.$$

Equation (10) implies that

$$\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega} = \left[\frac{\partial \omega_i}{\partial x_j}\right] = D_W^{\mathsf{T}}, \qquad (14)$$

and so

$$F^W = \frac{1}{\sigma^2}D_W D_W^{\mathsf{T}} = \frac{1}{\sigma^2}L_W$$

Hence F^W is proportional to the matrix of the Kirchhoff map in the standard basis, but with the row and column corresponding to the reference vertex removed. In a similar way the Fisher information matrix for estimation of the ν_j is

$$F^S = \frac{1}{\sigma^2}D_{WS}^{\mathsf{T}}L_W D_{WS}.$$

Note that $\det L_W$ is a minor of the Kirchhoff matrix L, and so by the Kirchhoff matrix tree theorem it is equal to the number of spanning trees $t(\Gamma)$ of Γ. By (12), the determinant of the Fisher information is thus

$$\det F^W = \det F^S = \sigma^{-2(n-1)}\det L_W$$
$$= \sigma^{-2(n-1)}t(\Gamma) \qquad (15)$$

The best possible situation occurs when the pairwise difference between all nodes is measured. In this case, Γ is a complete graph and the number of spanning trees is known to be $t(\Gamma) = n^{(n-2)}$. In this situation, the "average" Fisher information per node is

$$(\det F^W)^{1/(n-1)} = \frac{1}{\sigma^2}n^{(n-2)/(n-1)} \sim \frac{n}{\sigma^2} \quad \text{as } n \to \infty$$

Both of the ML estimators $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\omega}}$ are unbiased, and as a result, since the covariance matrix of $\boldsymbol{r}$ is $\sigma^2\mathbb{I}$,

$$C_{\hat{\boldsymbol{x}}} = \mathsf{E}\{(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^{\mathsf{T}}\} = \frac{1}{\sigma^2}L_W,$$

and its determinant is

$$\det C_{\hat{\boldsymbol{x}}} = \frac{\sigma^{2(n-1)}}{t(\Gamma)}, \qquad (16)$$

as anticipated from (15). The covariance matrix of the estimator $\hat{\omega}$ is

$$C_{\hat{\omega}} = \mathsf{E}\{(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}})(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}})^{\mathsf{T}}\}$$
$$= D_W^{\mathsf{T}}(F^W)^{-1}D_W$$
$$= \sigma^2 D_W^{\mathsf{T}}L_W^{-1}D_W$$
$$= \sigma^2 P_U$$

since $P_U = D_W^{\mathsf{T}}L_W^{-1}D_W$ is the orthogonal projection onto $U(\Gamma)$. An interesting consequence of this observation is that

$$\mathrm{Tr}\, C_{\hat{\omega}} = \sigma^2 \dim U(\Gamma)$$

B. Network design for independent errors on edges

Before going on to other measurement models, it is instructive to consider briefly the consequences of the above results in the design of a synchronization or registration scheme for a network. Since the ML estimator (13) is unbiased for any graph Γ, the role of the number of spanning trees $t(\Gamma)$ in the determinant of the estimator covariance matrix (16), or equivalently in the determinant of the Fisher information matrix (15), shows that a large number of spanning trees is desirable for good estimator performance.
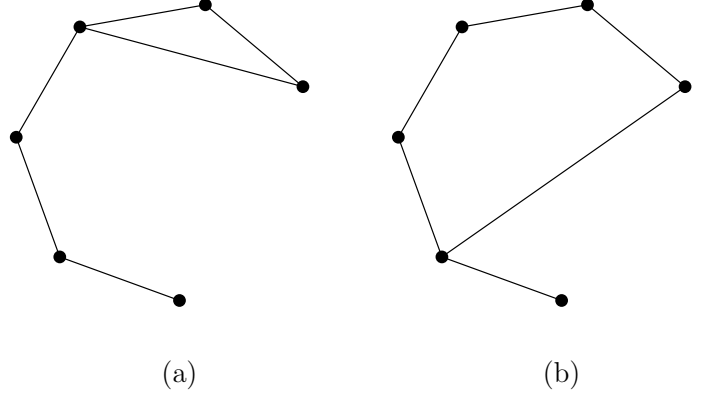

(a)   (b)

Fig. 2. Two networks with the same number of nodes and links. Network (a) has two spanning trees while network (b) has four spanning trees and is hence superior in the estimation context of this paper.

The network depicted in Figure 2(a) has the same number of nodes and links as the one in Figure 2(b). But the number of spanning trees in the former is two while the number of spanning trees in the latter is four. So, under the model assumed in this paper, estimation fidelity will be better for the network of Figure 2(b). From the perspective of design, if one has the opportunity to add one link to the acyclic network shown in Figure 3, the best choice in the context of this paper is to create a ring network (five spanning trees) and the worst is to create the network of Figure 2(a).
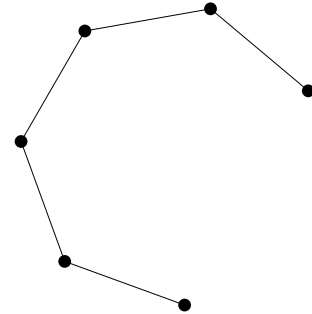

Fig. 3. An acyclic network. If one link can be added, the best choice is to create a ring network. The worst choice is to create the network shown in Figure 2(a).

C. Estimation with correlated measurements

This section further examines the situation where $G = \mathbb{R}$ and the measurement model is given by (8), but now with the measurement errors, in the standard basis for $C_1(\Gamma)$, being

jointly Gaussian with covariance matrix R. In this setting, the probability density of the measurements is

$$p(\boldsymbol{r}|\boldsymbol{\omega}) = \frac{1}{\sqrt{(2\pi)^m \det R}} \exp\left(-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\omega})R^{-1}(\boldsymbol{r}-\boldsymbol{\omega})^\mathsf{T}\right)$$

The maximum-likelihood estimate of $\boldsymbol{\omega} \in U(\Gamma)$ is obtained by splitting the data $\boldsymbol{r}$ as

$$\boldsymbol{r} = \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\varepsilon}}, \tag{17}$$

where $\hat{\boldsymbol{\omega}} \in U(\Gamma)$ and residual $\hat{\boldsymbol{\varepsilon}}$ satisfies $R^{-1}\hat{\boldsymbol{\varepsilon}} \in Z(\Gamma)$. That is,

$$\hat{\boldsymbol{\omega}} = Q_U \boldsymbol{r}$$

where Q_U is now an oblique projection with range $U(\Gamma)$ and null space $R^{-1}(Z(\Gamma))$. This situation is illustrated in Figure 4
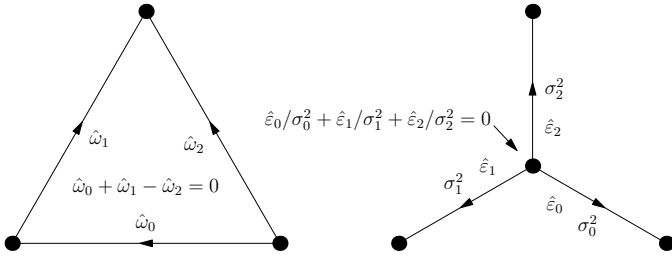

Fig. 4. The residual $\boldsymbol{\varepsilon}$ of the ML estimate has the $R^{-1}\hat{\boldsymbol{\varepsilon}}$ satisfies the Kirchhoff current law.

for a diagonal covariance matrix with entries σ_j^2. The estimate $\hat{\boldsymbol{\omega}}$ satisfies the Kirchhoff voltage law; i.e., the oriented sum of $\hat{\boldsymbol{\omega}}$ around any cycle as intended. It is interesting to observe that it is now $R^{-1}\hat{\boldsymbol{\varepsilon}}$ that satisfies the Kirchhoff current law; i.e., the oriented sum at any vertex is zero, with the covariance R playing the role of resistance in a way akin to Ohm's law.

The columns of D_W^T are a basis for $U(\Gamma)$ and (4) implies that the columns of $R^{-1}D_W^\mathsf{T}$ are a basis for $(R^{-1}(Z(\Gamma)))^\perp$. Thus,

$$\hat{\boldsymbol{\omega}} = D_W^\mathsf{T}\left(D_W R^{-1} D_W^\mathsf{T}\right)^{-1} D_W R^{-1} \boldsymbol{r}$$

The corresponding ML estimate for the vertex offsets $\boldsymbol{x}$ is

$$\hat{\boldsymbol{x}} = \left(D_W R^{-1} D_W^\mathsf{T}\right)^{-1} D_W R^{-1} \boldsymbol{r}$$

Motivated by these expressions, it is convenient to define the weighted Laplacian

$$\tilde{L} = D R^{-1} D^\mathsf{T}$$

and similarly

$$\tilde{L}_W = D_W R^{-1} D_W^\mathsf{T}$$

By (14), the Fisher information matrix for the vertex parametrization is given by

$$F^W = -\mathsf{E}\left\{\boldsymbol{\nabla}_{\boldsymbol{x}}^2 \log p(\boldsymbol{r}|\boldsymbol{\omega})\right\}$$
$$= D_W R^{-1} D_W^\mathsf{T} = \tilde{L}_W$$

The Cauchy-Binet formula [9] allows the determinant of the Fisher information to be written as

$$\det F^W = \det(D_W R^{-1} D_W^\mathsf{T})$$
$$= \sum_T \sum_{T'} \det(D_{WT}) \det([R^{-1}]_{TT'}) \det(D_{WT'})$$

where T and T' denote subsets of columns (edges) which are retained. The sum extends over all subsets of $E(\Gamma)$ of order $n-1$. The matrices D_{WT} have the property [8, p. 101]

$$\det D_{WT} = \begin{cases} \pm 1, & \text{if } T \text{ is a spanning tree,} \\ 0, & \text{otherwise.} \end{cases}$$

The only terms in the above sum over T which are non-zero are those corresponding to spanning trees and so

$$\det F^W = \sum_{S,S} \alpha_{SS'} \det([R^{-1}]_{SS'}) \tag{18}$$

The quantity

$$\alpha_{SS'} = \det(D_{WS} D_{WS'}^\mathsf{T}) \tag{19}$$

takes values ± 1 depending on the pair of spanning trees S and S' and independently of the choice of W.

With the assumption $R = \mathrm{diag}(\sigma_1^2, ..., \sigma_m^2)$, (18) simplifies to

$$\det F^W = \sum_S \prod_{e_j \in S} \frac{1}{\sigma_j^2},$$

which reduces to further to (15) when $\sigma_j^2 = \sigma^2$ for all $e_j \in E(\Gamma)$.

The ML estimate of the vertex offsets $\boldsymbol{x}$ is unbiased and its covariance matrix is

$$C_{\hat{\boldsymbol{x}}} = \mathsf{E}\{(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^\mathsf{T}\} = \tilde{L}_W^{-1}$$

and has determinant

$$\det C_{\hat{\boldsymbol{x}}} = \frac{1}{\sum_{S,S} \alpha_{SS'} \det([R^{-1}]_{SS'})}$$

from (18). The ML estimate of $\boldsymbol{\omega}$ is also unbiased and has covariance

$$C_{\hat{\boldsymbol{\omega}}} = \mathsf{E}\{(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}})(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}})^\mathsf{T}\} = D_W^\mathsf{T} \tilde{L}_W^{-1} D_W$$
$$= Q_U R$$

D. Multi-dimensional Gaussian problem

This section further generalizes the setting to $G = \mathbb{R}^d$, where the state of each vertex in the network is a vector in $\mathbb{R}^d$. In this situation, it is necessary to consider more general functions of the graph Γ than were treated in Section II. The vertex space now consists of functions $V(\Gamma) \to \mathbb{R}^d$ and is correspondingly denoted by $C_0(\Gamma, \mathbb{R}^d)$. In fact,

$$C_0(\Gamma, \mathbb{R}^d) = \mathbb{R}^d \otimes C_0(\Gamma, \mathbb{R})$$

In terms of the standard basis $\boldsymbol{q}_j$, $j = 1, \cdots, d$ for $\mathbb{R}^d$, any element of $C_0(\Gamma, \mathbb{R}^d)$ can be expressed as

$$\boldsymbol{x} = \sum_{i=1}^d \sum_{j=1}^n x_{ij} \boldsymbol{q}_i \otimes \boldsymbol{v}_j = \sum_{j=1}^n \boldsymbol{x}_j \otimes \boldsymbol{v}_j$$

Similarly, the vector space of functions $E(\Gamma) \to \mathbb{R}^d$ is

$$C_1(\Gamma, \mathbb{R}^d) = \mathbb{R}^d \otimes C_1(\Gamma, \mathbb{R})$$

The boundary map on $C_1(\Gamma, \mathbb{R}^d)$ is $\mathbb{I} \otimes D$ where $\mathbb{I}$ is the identity map on $\mathbb{R}^d$ and D is the boundary map on $C_1(\Gamma)$. The coboundary map on $C_0(\Gamma, \mathbb{R}^d)$ is $\mathbb{I} \otimes D^\mathsf{T}$.

As in the one-dimensional case, the cycle space $Z(\Gamma, \mathbb{R}^d)$ is defined to be the kernel of the boundary map; i.e., the set of $z \in C_1(\Gamma, \mathbb{R}^d)$ such that

$$(\mathbb{I} \otimes D)z = 0.$$

Writing

$$z = \sum_{j=1}^{n} z_j \otimes e_j$$

the cycles satisfy a vector form of (5).

$$\sum_{e_j : t(e_j) = v} z_j = \sum_{e_j : s(e_j) = v} z_j,$$

i.e., at any vertex, the oriented vector sum of the values of z on the set of edges meeting at the vertex is zero.

The cocycle space $U(\Gamma, \mathbb{R}^d)$ is the image of the coboundary map $\mathbb{I} \otimes D^\mathsf{T}$. For any

$$\omega = \sum_{j=1}^{n} \omega_j \otimes e_j \in U(\Gamma, \mathbb{R}^d)$$

the equation

$$\sum_{e_j \in \mathfrak{L}} (-1)^{\sigma_j} \omega_j = 0,$$

must hold for for all cycles $\mathfrak{L}$. In this expression, $\sigma_j = 0$ if e_j is oriented as $\mathfrak{L}$ and $\sigma_j = 1$ if e_j is oriented opposite to $\mathfrak{L}$.

An inner product can be defined on $C_1(\Gamma, \mathbb{R}^d)$, based on the inner product defined earlier on $C_1(\Gamma)$ and the standard inner product on $\mathbb{R}^d$, namely, for $s_1, s_2 \in \mathbb{R}^d$ and $x_1, x_2 \in C_1(\Gamma)$,

$$\langle s_1 \otimes x_1, s_1 \otimes x_1 \rangle = \langle s_1, s_2 \rangle_{\mathbb{R}^d} \langle x_1, x_2 \rangle_{C_1(\Gamma)}$$

For all $z \in Z(\Gamma, \mathbb{R}^d)$ and $\omega \in U(\Gamma, \mathbb{R}^d)$

$$\langle z, \omega \rangle = 0,$$

i.e., $Z(\Gamma, \mathbb{R}^d)$ is the orthogonal complement of $U(\Gamma, \mathbb{R}^d)$.

1) Independent identically distributed edge measurements: Recall that the measurement model is $r = \omega + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, R)$, r and ε in $C_1(\Gamma, \mathbb{R}^d)$, and $\omega \in U(\Gamma, \mathbb{R}^d)$. The first case of interest is when the errors $\varepsilon_j \in \mathbb{R}^d$ are independent. Their individual probability densities are

$$p(\varepsilon_j) = \frac{1}{\sqrt{(2\pi)^d \det R}} \exp\left(-\frac{1}{2} \langle \varepsilon_j, R^{-1} \varepsilon_j \rangle_{\mathbb{R}^d} \right)$$

for $j = 1, \cdots, m$. By independence of the ε_j, the joint probability density for the data $r \in C_1(\Gamma, \mathbb{R}^d)$ is thus

$$p(r|\omega) = \frac{1}{((2\pi)^d \det R)^{m/2}} \exp -\frac{1}{2} \langle r - \omega, (R^{-1} \otimes I)(r - \omega) \rangle$$

The maximum likelihood estimate of ω is obtained by splitting the data r as

$$r = \hat{\omega} + \hat{\varepsilon},$$

where $\hat{\omega} \in U(\Gamma, \mathbb{R}^d)$ and $\hat{\varepsilon} \in Z(\Gamma, \mathbb{R}^d)$. Explicitly, $\omega \in U(\Gamma, \mathbb{R}^d)$ is parameterized in terms of the offsets of $n-1$ vertices with respect to a reference vertex

$$\omega = (\mathbb{I} \otimes D_W^\mathsf{T})x$$

or alternatively, in terms of a spanning tree S, as

$$\omega = (\mathbb{I} \otimes D_W^\mathsf{T} D_{WS}^{-1})\nu \qquad (20)$$

where x and ν are related by

$$x = (\mathbb{I} \otimes D_{WS}^{-1})\nu$$

In terms of these parameterizations, the maximum-likelihood estimate is

$$\hat{x} = (R^{-1} \otimes L_W^{-1})(R^{-1} \otimes D_W)r$$
$$= (\mathbb{I} \otimes L_W^{-1} D_W)r,$$

and

$$\hat{\nu} = (\mathbb{I} \otimes D_{WS}^{-1} L_W^{-1} D_W)r$$

So $\hat{\omega} = (\mathbb{I} \otimes D_W^\mathsf{T} L_W^{-1} D_W)r$.

The Fisher information matrix for x is given by

$$F^W = -\mathsf{E}\left\{ \nabla_x^2 \log p(r|\omega) \right\}$$
$$= (\nabla_x \omega)^\mathsf{T} (\mathbb{I} \otimes R^{-1}) \nabla_x \omega$$

Equation (20) implies that

$$\nabla_x \omega = \mathbb{I} \otimes D_W^\mathsf{T},$$

so that

$$F^W = R^{-1} \otimes D_W D_W^\mathsf{T}$$
$$= R^{-1} \otimes L_W$$

In a similar way, the Fisher information matrix for estimation of ν is

$$F^S = R^{-1} \otimes D_{WS}^\mathsf{T} L_W D_{WS}.$$

The determinant of the Fisher information in both cases is

$$\det F^W = \det F^S = \frac{(\det L_W)^d}{(\det R)^{n-1}} = \frac{t(\Gamma)^d}{(\det R)^{n-1}},$$

which is obtained by using the tensor product identity

$$\det A \otimes B = (\det A)^{\dim B} (\det B)^{\dim A}.$$

Finally, the error covariance of the unbiased ML estimate $\hat{\omega}$ is

$$C_{\hat{\omega}} = \mathsf{E}\{(\omega - \hat{\omega})(\omega - \hat{\omega})^\mathsf{T}\}$$
$$= R \otimes D_W^\mathsf{T} L_W^{-1} D_W = R \otimes P_U,$$

where P_U is the orthogonal projection onto $U(\Gamma)$. Taking the partial trace over $C_1(\Gamma)$ yields

$$\mathrm{Tr}_{C_1(\Gamma)} C_{\hat{\omega}} = (\dim U(\Gamma))R.$$

2) Correlated edge measurements: Now assume the errors on the edge measurements $\varepsilon_j \in \mathbb{R}^d$ have joint probability density

$$p(\varepsilon) = \frac{1}{\sqrt{(2\pi)^{md} \det R}} \exp\left(-\frac{1}{2} \langle \varepsilon, R^{-1} \varepsilon \rangle_{C_1(\Gamma, \mathbb{R}^d)} \right)$$

where R now denotes the $md \times md$ covariance matrix of the m d-dimensional edge measurements. The probability density for the data $r \in C_1(\Gamma, \mathbb{R}^d)$ is

$$p(r|\omega) = \frac{1}{\sqrt{(2\pi)^{md} \det R}} \exp\left(-\frac{1}{2} \langle r - \omega, R^{-1} r - \omega \rangle \right)$$

where the inner product in the exponent is in $C_1(\Gamma, \mathbb{R}^d)$. The ML estimate involves a decomposition of the data as

$$\boldsymbol{r} = \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\varepsilon}},$$

where $\hat{\boldsymbol{\omega}} \in U(\Gamma, \mathbb{R}^d)$ and $\hat{\boldsymbol{\varepsilon}} \in R^{-1}(Z(\Gamma, \mathbb{R}^d))$. Explicitly,

$$\hat{\boldsymbol{\omega}} = Q_U \boldsymbol{r},$$

where Q_U is the oblique projection with range $U(\Gamma, \mathbb{R}^d)$ and null space $R^{-1}(Z(\Gamma, \mathbb{R}^d))$; i.e.,

$$Q_U = (\mathbb{I} \otimes D_W^\mathsf{T}) \left((\mathbb{I} \otimes D_W) R^{-1} (\mathbb{I} \otimes D_W^\mathsf{T}) \right)^{-1} (\mathbb{I} \otimes D_W) R^{-1}$$

The ML estimate of the vertex offsets is

$$\hat{\boldsymbol{x}} = \left((\mathbb{I} \otimes D_W) R^{-1} (\mathbb{I} \otimes D_W^\mathsf{T}) \right)^{-1} (\mathbb{I} \otimes D_W) R^{-1} \boldsymbol{r}.$$

The Fisher information for $\boldsymbol{x}$ is

$$F^W = (\mathbb{I} \otimes D_W) R^{-1} (\mathbb{I} \otimes D_W^\mathsf{T})$$

In order to calculate the determinant of F^W, it is helpful to first find

$$\det(\mathbb{I} \otimes D_W)_{\mathcal{T}}$$

where $\mathcal{T}$ denotes a subset of $(n-1)d$ of the columns. Now $(\mathbb{I} \otimes D_W)$ has the form

$$\begin{pmatrix} D_W & 0 & 0 & 0 \\ 0 & D_W & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & D_W \end{pmatrix}$$

Write $\mathcal{T} = (T_1, T_2, \cdots, T_d)$, where T_k denotes the set of columns selected from the kth block. Since Γ is assumed to be connected, D_W has rank $n-1$ thus $\det(\mathbb{I} \otimes D_W)_{\mathcal{T}} = 0$ unless exactly $n-1$ columns are chosen from each block, in which case

$$\det(\mathbb{I} \otimes D_W)_{\mathcal{T}} = \prod_{j=1}^d \det(D_{WT_j})$$
$$= \begin{cases} \pm 1, & \text{if each } T_j \text{ is a spanning tree} \\ 0, & \text{otherwise,} \end{cases}$$

$\mathcal{S} = (S_1, S_2, \ldots, S_d)$ will be called a multi-spanning tree if it consists of one spanning tree for each dimension of the parameter space R^d. Applying the Cauchy-Binet formula gives

$$\det F^W = \sum_{\mathcal{T}\mathcal{T}'} \det(\mathbb{I} \otimes D_W)_{\mathcal{T}} \det([R^{-1}]_{\mathcal{T}\mathcal{T}'}) \det(\mathbb{I} \otimes D_W^\mathsf{T})_{\mathcal{T}}$$
$$= \sum_{\mathcal{S}\mathcal{S}'} \alpha_{\mathcal{S}\mathcal{S}'} \det([R^{-1}]_{\mathcal{S}\mathcal{S}'}) \tag{21}$$

where the sum is over multi-spanning trees. In this expression,

$$\alpha_{\mathcal{S}\mathcal{S}'} = \prod_{j=1}^d \alpha_{S_j S_j'}$$

with $\alpha_{\mathcal{S}\mathcal{S}'}$ given by (19), which takes values ± 1. If the edge measurement errors are independent then (21) reduces to

$$\det F^W = \sum_{\mathcal{S}} \det([R^{-1}]_{\mathcal{S}\mathcal{S}}). \tag{22}$$

The error covariance of the ML estimate of the vertex offsets $\boldsymbol{x}$ is

$$\mathsf{E}\{(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^\mathsf{T}\} = \left((\mathbb{I} \otimes D_W) R^{-1} (\mathbb{I} \otimes D_W^\mathsf{T}) \right)^{-1}$$

So,

$$\det \mathsf{E}\{(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^\mathsf{T}\} = \frac{1}{\sum_{\mathcal{S}\mathcal{S}'} \alpha_{\mathcal{S}\mathcal{S}'} \det([R^{-1}]_{\mathcal{S}\mathcal{S}'})}$$

and

$$\mathsf{E}\{(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})(\boldsymbol{\omega} - \hat{\boldsymbol{\omega}})^\mathsf{T}\}$$
$$= (\mathbb{I} \otimes D_W^\mathsf{T}) \left((\mathbb{I} \otimes D_W) R^{-1} (\mathbb{I} \otimes D_W^\mathsf{T}) \right)^{-1} (\mathbb{I} \otimes D_W^\mathsf{T})$$
$$= Q_U R$$

IV. LOCAL ESTIMATORS FOR GAUSSIAN NOISE ON $\mathbb{R}$

The estimators presented thus far have either implicitly or explicitly assumed a fixed reference vertex or a particular spanning tree. This is intrinsically incompatible with the development of local estimators; i.e., estimators that can be implemented in ways that each vertex follows some procedure that uses only information accessible from its nearest neighbors. Consider again the situation involving Gaussian noise on $\mathbb{R}$ with the noise on edge measurements being independent, treated in Section III-A, and recall that the ML estimate for $\boldsymbol{\omega}$ is such that the residual $(\boldsymbol{r} - \hat{\boldsymbol{\omega}}) \in Z(\Gamma)$; i.e., it obeys Kirchhoff's current law. Thus, $\hat{\boldsymbol{\omega}}$ is the solution of

$$D\hat{\boldsymbol{\omega}} = D\boldsymbol{r}.$$

Indeed, if $\boldsymbol{x}$ is any element of $C_0(\Gamma)$ satisfying

$$L\boldsymbol{x} = D\boldsymbol{r}$$

then $\hat{\boldsymbol{\omega}} = D^\mathsf{T}\boldsymbol{x}$ will be the ML estimate of $\boldsymbol{\omega}$. The quantity $\boldsymbol{x}$ may be interpreted as the collection of offsets each vertex needs to apply to its own coordinate in order for the entire network to be aligned in a statistically optimal way. It is interesting to note that addition of a positive multiple of the orthogonal projection onto the subspace of constant functions on $V(\Gamma)$ to the Laplacian matrix creates an invertible matrix. Hence a solution to (IV) is obtained from

$$\boldsymbol{x} = (L + \frac{\mu}{n} \mathbf{1}\mathbf{1}^\mathsf{T})^{-1} D\boldsymbol{r}$$

where $\mathbf{1}$ denotes a vector of ones. Since $\mathbf{1}^\mathsf{T} D = 0$, this is the same solution for all $\mu > 0$. In fact, it is the solution satisfying $\mathbf{1}^\mathsf{T}\boldsymbol{x} = 0$; i.e., the set of offsets obtained do not change the mean of the vertex states across the network.

If the linear system (IV) is solved using Jacobi's method [10], the structure of the Laplacian matrix L ensures this algorithm is local. Jacobi's method involves writing $L = N - A$ in terms of the diagonal degree matrix N and the adjacency matrix A and applying the recursion

$$\boldsymbol{x}^{(t+1)} = N^{-1} \left(D\boldsymbol{r} + A\boldsymbol{x}^{(t)} \right) \tag{23}$$

A fixed point of this recursion satisfies (40). Thus if the method converges, it gives the ML estimate. Jacobi's method is known to converge if the matrix L is diagonally dominant [11]; i.e.,

$$|L_{ii}| > \sum_{j \neq i} |L_{ij}|, \tag{24}$$

although this is not a necessary condition. The Laplacian matrix of a graph satisfies

$$|L_{ii}| = \sum_{j \neq i} |L_{ij}|.$$

Another sufficient condition for convergence of Jacobi's method is the so-called *walk-summability* condition [12]. This specifies that the spectral radius of $N^{-1}A$ is less than one; i.e.,

$$\rho(N^{-1}A) < 1. \qquad (25)$$

The Gershgorin circle theorem [9] implies that $\rho(N^{-1}A) \leq 1$.

The Jacobi algorithm has been applied in MAP estimation using Gaussian belief propagation in Bayesian belief networks [13]. In this work, it is noted that when neither (24) and (25) are satisfied and Jacobi's method fails to converge, convergence can be forced by using a double-loop iterative method. In the estimation problem of interest in this section, the possible violations of the sufficient conditions for convergence are as mild as can be found in practice. In experiments run to date, Jacobi's method has never failed to converge.

Once the recursion (23) is written out in detail, the update for the kth vertex becomes

$$x_k^{(t+1)} = \frac{1}{n_k} \sum_{v_\ell \sim v_k} \left(x_\ell^{(t)} + r_{(\ell,k)} \right),$$

where $r_{(\ell,k)}$ is r_e if $e = (v_\ell, v_k) \in E(\Gamma)$ or $-r_e$ if $e = (v_k, v_\ell) \in E(\Gamma)$. Thus the recursion is indeed local. Note that $x_\ell^{(t)} + r_{(\ell,k)}$ is the current prediction at the neighboring vertex ℓ of the value at the kth should be. At each vertex, a single iteration of the algorithm can be summarized as "become the mean of what your neighbors say your value should be."

An alternate way to state the recursion, which will be important subsequent application to estimation in Lie groups other that $\mathbb{R}^d$, is as follows. Denote an action of $\mathbb{R}$ on itself by

$$T_r x = x + r$$

Then (23) can be written as

$$\boldsymbol{x}^{(t+1)} = Q\boldsymbol{x}^{(t)}$$

where

$$Q_{k,\ell} = \begin{cases} \frac{1}{n_k} T_{r_{(k,\ell)}}, & \text{if } (v_\ell, v_k) \in E(\Gamma), \\ \frac{1}{n_k} T_{r_{(k,\ell)}}^{-1}, & \text{if } (v_k, v_\ell) \in E(\Gamma), \\ 0, & \text{otherwise.} \end{cases}$$

Although it may be possible to develop local algorithms for some forms of correlation between measurement noise on different edges of the graph, this possibility is not explored here. For $G = \mathbb{R}^d$, any $\boldsymbol{x} \in C_1(\Gamma, \mathbb{R}^d)$ satisfying

$$(\mathbb{I} \otimes L)\boldsymbol{x} = (\mathbb{I} \otimes D)\boldsymbol{r} \qquad (26)$$

then $\hat{\boldsymbol{\omega}} = (\mathbb{I} \otimes D^{\mathsf{T}})\boldsymbol{x}$ will be the ML estimate of $\boldsymbol{\omega}$. Jacobi's method in this case give the recursion

$$\boldsymbol{x}^{(t+1)} = (\mathbb{I} \otimes N)^{-1} \left((\mathbb{I} \otimes D)\boldsymbol{r} + (\mathbb{I} \otimes A)\boldsymbol{x}^{(t)} \right) \qquad (27)$$

The fixed points of (27) satisfy (26) and thus

$$|(\mathbb{I} \otimes L)_{ii}| = \sum_{j \neq i} |(\mathbb{I} \otimes L)_{ij}|.$$

and by the Gershgorin circle theorem

$$\rho((\mathbb{I} \otimes N)^{-1}(\mathbb{I} \otimes A)) \leq 1.$$

In terms of the action of $\mathbb{R}^d$ on itself

$$T_{\boldsymbol{r}} = \boldsymbol{x} + \boldsymbol{r},$$

(27) becomes

$$\boldsymbol{x}^{(t+1)} = Q\boldsymbol{x}^{(t)}$$

where the operator Q has a "block" form with the $(k,\ell)^{\text{th}}$ block being

$$Q_{k,\ell} = \begin{cases} \frac{1}{n_k} T_{\boldsymbol{r}_{(k,\ell)}}, & \text{if } (v_\ell, v_k) \in E(\Gamma), \\ \frac{1}{n_k} T_{\boldsymbol{r}_{(k,\ell)}}^{-1}, & \text{if } (v_k, v_\ell) \in E(\Gamma), \\ 0, & \text{otherwise.} \end{cases}$$

V. PHASE ALIGNMENT

This section addresses the case of phase estimation in a network. For this problem, the natural parameter space at the vertices is the group of real numbers modulo 1, or equivalently the circle group comprised of complex numbers with absolute value one; i.e., $\mathbb{T} = \{e^{2\pi i \theta} : \theta \in [0,1)\}$. In what follows, it is convenient to use the latter description.

In the preceding sections, the offsets between measurements at vertices naturally form elements of a vector space. In the current situation this will no longer be true, resulting in significant differences in both theory and algorithms. These issues are elucidated here in the context of measurements on the circle $\mathbb{T}$ and in slightly more generality in Section VI, where the data reside in a connected abelian Lie group. The setting of a compact (not necessarily abelian) Lie group, which includes the important practical situation $G = SO(3)$, will be addressed in a sequel to this paper.

A. Cycles and Cocycles

Following the treatment in Section II, denote by $C_0(\Gamma, \mathbb{C})$ the collection of functions from the vertices $V(\Gamma)$ to $\mathbb{C}$. Similarly, denote by $C_1(\Gamma, \mathbb{C})$ the vector space of complex-valued functions on the edges $E(\Gamma)$. Any element of $C_0(\Gamma, \mathbb{C})$ can be written in terms of the basis functions (1) as

$$\boldsymbol{x} = \sum_{j=1}^{n} x_j \boldsymbol{v}_j$$

where the x_j are now complex numbers. Similarly, the basis functions (2) can be used to write any element of $C_1(\Gamma, \mathbb{C})$ as

$$\boldsymbol{x} = \sum_{j=1}^{n} x_j \boldsymbol{e}_j$$

with $x_j \in \mathbb{C}$. The definitions of the incidence map, etc., then generalize in a straightforward way to this situation.

In this setting, the appropriate space for the measurement data is the collection of functions from the edges $E(\Gamma)$ to $\mathbb{T}$. Motivated by analogy with the preceding cases, this will be

denoted by $C_1(\Gamma, \mathbb{T})$. Also as earlier, $C_0(\Gamma, \mathbb{T})$ will denote the collection of functions from $V(\Gamma)$ to $\mathbb{T}$. Both $C_1(\Gamma, \mathbb{T})$ and $C_0(\Gamma, \mathbb{T})$ are groups under pointwise multiplication of complex numbers. However, although $C_0(\Gamma, \mathbb{T}) \subset C_0(\Gamma, \mathbb{C})$ and $C_1(\Gamma, \mathbb{T}) \subset C_1(\Gamma, \mathbb{C})$, neither $C_0(\Gamma, \mathbb{T})$ or $C_1(\Gamma, \mathbb{T})$ are vector spaces. In the absence of a vector space structure, in particular the abilities to use bases and to use inner products to identify spaces with their duals, it is necessary to adopt a modified approach to the definition of the boundary and coboundary operators. This begins by considering the "spaces" of functions $C_0(\Gamma, \mathbb{Z}) \subset C_0(\Gamma, \mathbb{C})$ from the set of vertices of Γ to the integers $\mathbb{Z}$ and $C_1(\Gamma, \mathbb{Z}) \subset C_1(\Gamma, \mathbb{C})$ from $E(\Gamma)$ to $\mathbb{Z}$. An element of $C_0(\Gamma, \mathbb{Z})$ can be expressed in terms of the basis functions for $C_0(\Gamma, \mathbb{C})$, which are themselves elements of $C_0(\Gamma, \mathbb{Z})$, as

$$z = \sum_{j=1}^{n} m_j \boldsymbol{v}_j, \quad m_j \in \mathbb{Z}$$

It is straightforward to verify that set $C_0(\Gamma, \mathbb{Z})$ is an abelian group. Similarly, an element of $C_1(\Gamma, \mathbb{Z})$ can be expanded in terms of the basis elements $\boldsymbol{e}_j$ of $C_1(\Gamma, \mathbb{C})$ with integer coefficients.

In place of the inner product are "pairings" between $C_i(\Gamma, \mathbb{Z})$ and $C_i(\Gamma, \mathbb{T})$ given by

$$\langle \boldsymbol{z}, \boldsymbol{\tau} \rangle_{\mathbb{T},1} = \prod_{e \in E(\Gamma)} \tau_e^{z_e} \quad (\boldsymbol{z} \in C_1(\Gamma, \mathbb{Z}),\ \boldsymbol{\tau} \in C_1(\Gamma, \mathbb{T})),$$

$$\langle \boldsymbol{z}, \boldsymbol{\tau} \rangle_{\mathbb{T},0} = \prod_{v \in V(\Gamma)} \tau_v^{z_v} \quad (\boldsymbol{z} \in C_0(\Gamma, \mathbb{Z}),\ \boldsymbol{\tau} \in C_0(\Gamma, \mathbb{T})),$$

$$(28)$$

where, in each case, $\tau^{-1} = \overline{\tau}$ is the complex conjugate of $\tau \in \mathbb{T}$.

The boundary operator $D : C_1(\Gamma, \mathbb{C}) \to C_0(\Gamma, \mathbb{C})$ is defined, as in Section II on the "basis" elements by

$$D(\boldsymbol{e}) = t(e) - s(e) \quad (\boldsymbol{e} \in C_1(\Gamma, \mathbb{C})), \qquad (29)$$

with the convention that a vertex is identified with the corresponding member of the basis of $C_0(\Gamma, \mathbb{C})$. When restricted to $C_1(\Gamma, \mathbb{Z})$, the operator D maps $C_1(\Gamma, \mathbb{Z})$ into $C_0(\Gamma, \mathbb{Z})$. Similarly, D^{T} maps $C_0(\Gamma, \mathbb{Z})$ into $C_1(\Gamma, \mathbb{Z})$ and the Laplacian $L = DD^{\mathsf{T}}$, adjacency, and degree maps all map $C_0(\Gamma, \mathbb{Z})$ into $C_0(\Gamma, \mathbb{Z})$.

The functions $\boldsymbol{z}_{\mathfrak{L}}$ defined in (3), where $\mathfrak{L}$ is taken over all cycles, are functions in $C_1(\Gamma, \mathbb{Z})$. The cycle space $Z(\Gamma, \mathbb{Z}) \subset C_1(\Gamma, \mathbb{Z})$ is now regarded as a set of integer combinations of the $\boldsymbol{z}_{\mathfrak{L}}$; i.e., it consists of integer sums $\sum_k n_k \boldsymbol{z}_{\mathfrak{L}_k}$ where $\mathfrak{L}_k$ are cycles. The cycle space is the kernel of the boundary operator (29) acting on $C_1(\Gamma, \mathbb{Z})$, so $D\boldsymbol{z} = 0$ for all $\boldsymbol{z} \in C_1(\Gamma, \mathbb{Z})$. Note that $Z(\Gamma, \mathbb{Z}) \subset Z(\Gamma, \mathbb{C})$.

With $\boldsymbol{x} : V(G) \to \mathbb{T} \in C_0(\Gamma, \mathbb{T})$, the $\mathbb{T}$-coboundary operator can be defined as

$$D_*(\boldsymbol{\tau})(e) = \overline{\tau}_{t(e)} \tau_{s(e)}.$$

This is the dual operator of D using the pairings $\langle \cdot, \cdot \rangle_{\mathbb{T},i}$ defined in (28); i.e.,

$$\langle \boldsymbol{z}, D_*\boldsymbol{\tau} \rangle_{\mathbb{T},0} = \langle D\boldsymbol{z}, \boldsymbol{\tau} \rangle_{\mathbb{T},1},$$

for all $\boldsymbol{z} \in C_0(\Gamma, \mathbb{Z})$ and $\boldsymbol{\tau} \in C_1(\Gamma, \mathbb{T})$. The $\mathbb{T}$-cocycle space is defined to be the image of the operator D_*:

$$U(\Gamma, \mathbb{T}) = D_*(C_0(\Gamma, \mathbb{T})) \subset C_1(\Gamma, \mathbb{T}).$$

Members $\boldsymbol{\omega}$ of $U(\Gamma, \mathbb{T})$ share a property corresponding to (6); i.e.,

$$\langle \boldsymbol{z}, \boldsymbol{\omega} \rangle_{\mathbb{T},1} = \prod_{e \in E(\Gamma)} \omega_e^{z_e} = 1$$

for all $\boldsymbol{z} \in Z(\Gamma, \mathbb{Z})$. In other words, they are "orthogonal" to the cycle space $Z(\Gamma, \mathbb{Z})$ in the sense of the pairing defined in (28). This is the analogue of Kirchhoff's voltage law in this setting: the oriented product of the elements of $\boldsymbol{\omega}$ around any cycle is the identity element of $\mathbb{T}$ (i.e., 1).

B. Distributions

A crucial issue in the analysis of the phase alignment problem is specification of an appropriate model for noise on $\mathbb{T}$. Two distributions are typically considered for circular statistics [14]: the wrapped normal and the von Mises distributions. While the wrapped normal distribution is in wide use, using it here leads to an analysis that is essentially the same as given in Section III-A for Gaussian noise on $\mathbb{R}$. It is appropriate and interesting then to consider the effects of using the von Mises distributed in what follows.

Given a set of N points $z_j \in \mathbb{T}$, the usual definition of circular mean $\hat{z} \in \mathbb{T}$ is given by

$$A\hat{z} = \frac{1}{N} \sum_{j=1}^{N} z_j$$

The quantity A is a measure of concentration and is related to the circular variance ρ of the sample by $\rho = 1 - A^2$.

The von Mises distribution is the maximal entropy distribution $p(e^{i\theta})$ subject to the constraint

$$A\mu = \int_{\mathbb{T}} e^{i\theta} p(e^{i\theta})\, d\theta,$$

where μ is the circular mean and $1 - A^2$ is the circular variance. As such, it is the least biased distribution under this constraint. A random variable Z with this distribution is of the form $e^{i\theta_0} Z_0$ where Z_0 has circular mean one and the same circular variance as Z. The von Mises density function with circular mean μ and circular variance $1 - A^2$ is

$$p(z = e^{i\theta} | \mu, A) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\theta - \mu)}$$
$$= \frac{1}{2\pi I_0(\kappa)} e^{\frac{\kappa}{2}(z_0^* z + z_0 z^*)}, \qquad (30)$$

where

$$A = \frac{I_1(\kappa)}{I_0(\kappa)},$$

I_0, I_1 are respectively the first and second modified Bessel functions of the first kind, and $\boldsymbol{z}_0 = e^{i\mu}$. The value of A determines κ via this equation.

For each edge $e \in E(\Gamma)$, a measurement on e has the form

$$r_e = \omega_e \varepsilon_e$$

where $\boldsymbol{\omega} \in U_*(\Gamma, \mathbb{T})$. Thus, for some

$$\boldsymbol{x} = (x_v)_{v \in V(\Gamma)} \in C_1(\Gamma, \mathbb{T})$$

it is possible to write $\omega_e = x_{t(e)} x_{s(e)}^{-1}$. Hence

$$r_e = x_{t(e)} \varepsilon_e x_{s(e)}^{-1} \tag{31}$$

or, more succinctly,

$$\boldsymbol{r} = \boldsymbol{\varepsilon}\boldsymbol{\omega} = \boldsymbol{\varepsilon} D_* \boldsymbol{x}.$$

C. The estimation problem

The joint distribution of the noise $\boldsymbol{\varepsilon} \in C_1(\Gamma, \mathbb{T})$ is taken to be

$$p(\boldsymbol{\varepsilon}) = \prod_{e \in E(\Gamma)} \frac{1}{2\pi I_0(\kappa_e)} \exp \frac{\kappa_e}{2}(\varepsilon_e + \overline{\varepsilon_e}).$$

Consequently, the density of $\boldsymbol{r}$ conditioned on $\boldsymbol{\omega}$ is

$$p(\boldsymbol{r}|\boldsymbol{\omega}) = \prod_{e \in E(\Gamma)} \left(\frac{1}{2\pi I_0(\kappa_e)} \right) \exp \frac{\kappa_e}{2} \sum_{e \in E(\Gamma)} (\overline{\omega_e} r_e + \omega_e \overline{r_e}).$$

The log-likelihood is then

$$\ell(\boldsymbol{r}|\boldsymbol{\omega}) = \sum_{e \in E(\Gamma)} \frac{\kappa_e}{2}(\overline{\omega_e} r_e + \omega_e \overline{r_e}) + \sum_{e \in E(\Gamma)} \log \frac{1}{2\pi I_0(\kappa_e)}. \tag{32}$$

The Fisher information for this estimation problem can be found by first differentiating $\ell(\boldsymbol{r}|\boldsymbol{\omega})$ with respect to θ_v (cf. (30)):

$$
\begin{aligned}
\frac{\partial}{\partial \theta_v} \ell(\boldsymbol{r}|\boldsymbol{\omega}) &= \sum_{e:s(e)=v} \frac{\kappa_e}{2} i(\overline{\omega_e} r_e - \omega_e \overline{r_e}) \\
&\quad - \sum_{e:t(e)=v} \frac{\kappa_e}{2} i(\overline{\omega_e} r_e - \omega_e \overline{r_e}) \\
&= \operatorname{Im}\Big(\sum_{e:t(e)=v} \kappa_e \overline{\omega_e} r_e - \sum_{e:s(e)=v} \kappa_e \overline{\omega_e} r_e \Big) \\
&= \operatorname{Im} \sum_{e \in E(\Gamma)} D_{ve} \kappa_e \overline{\omega_e} r_e
\end{aligned}
$$

Differentiating again with respect to θ_u yields

$$\frac{\partial^2 \ell(\boldsymbol{r}|\boldsymbol{\omega})}{\partial \theta_u \partial \theta_v} = -\operatorname{Re} \sum_{e \in E(\Gamma)} \kappa_e D_{ue} D_{ve} \overline{\omega_e} r_e$$

The Fisher information is thus

$$
\begin{aligned}
F_{uv} &= -\mathsf{E}\left\{ \frac{\partial^2 \ell(\boldsymbol{r}|\boldsymbol{\omega})}{\partial \theta_u \partial \theta_v} \right\} \\
&= \operatorname{Re} \sum_{e \in E(\Gamma)} \kappa_e D_{ue} D_{ve} \overline{\omega_e} \mathsf{E}\{r_e\}
\end{aligned}
$$

Note that

$$\mathsf{E}\{r_e\} = \frac{I_1(\kappa_e)}{I_0(\kappa_e)} \omega_e,$$

so that this reduces to

$$F = D_W \mathcal{F} D_W^\mathsf{T}$$

where $\mathcal{F}$ is the diagonal matrix

$$\mathcal{F} = \operatorname{diag}\left(\frac{\kappa_{e_1} I_1(\kappa_{e_1})}{I_0(\kappa_{e_1})}, \cdots, \frac{\kappa_{e_m} I_1(\kappa_{e_m})}{I_0(\kappa_{e_m})} \right).$$

The quantity $\kappa_e I_1(\kappa_e)/I_0(\kappa_e)$ associated with an edge e may be interpreted as follows. Consider the problem of estimating ω_e given the datum r_e. As above, the conditional density is

$$p(r_e|\omega_e) = \frac{1}{2\pi I_0(\kappa_e)} \exp \frac{\kappa_e}{2}(\overline{\omega_e} r_e + \omega_e \overline{r_e}).$$

The parametrization, $\omega_e = \exp(i\phi_e)$, yields the Fisher information for estimation of ϕ_e from r_e as

$$\mathcal{F}_e = -\mathsf{E}\left\{ \frac{\partial^2}{\partial \phi_e^2} \log p(r_e|\omega_e) \right\} = \frac{\kappa_e I_1(\kappa_e)}{I_0(\kappa_e)}$$

Thus when the $\mathcal{F}_e$ are regarded as edge weights, the graph Fisher information is the weighted Laplacian with these weights. The determinant of the Fisher information is, by Kirchhoff's matrix tree theorem,

$$\det F = \sum_S \prod_{e \in S} \mathcal{F}_e.$$

If the edges of Γ share a common κ, this reduces to

$$\det F = \frac{\kappa I_1(\kappa)}{I_0(\kappa)} t(\Gamma).$$

D. Maximum-likelihood estimator

This section considers the problem of determining the maximum-likelihood estimator $\hat{\boldsymbol{x}}$ for the vertex offsets from data $\boldsymbol{r}$. Differentiation of the likelihood (32) reveals that the critical points must satisfy

$$\operatorname{Im} \sum_{e \in E(\Gamma)} D_{ve} \kappa_e \overline{\hat{\omega}_e} r_e = 0$$

at every vertex $v \in V(\Gamma)$. In this expression, $\hat{\omega}$ denotes the ML estimate of $\boldsymbol{\omega}$. If the map $K : C_1(\Gamma, \mathbb{T}) \to C_1(\Gamma, \mathbb{C})$ is defined by

$$K(\boldsymbol{\varepsilon}) = \sum_{e \in E(\Gamma)} \kappa_e \varepsilon_e$$

and the residual is written as $\hat{\varepsilon}_e = \hat{\omega}_e r_e$, the critical points of the likelihood satisfy

$$\hat{\omega} \in U(\Gamma, \mathbb{T})$$
$$\operatorname{Im}(K(\hat{\boldsymbol{\varepsilon}})) \in Z(\Gamma, \mathbb{C}) \tag{33}$$

and $\operatorname{Re}(K(\hat{\boldsymbol{\varepsilon}}))$ is the corresponding value of the log-likelihood (up to a constant). These correspond to (17) for the Gaussian case. This condition can be rearranged as

$$\sum_{e \in E(v)} \kappa_e \left(\overline{\hat{\omega}_e} r_e \right)^{D_{ve}} = \rho_v \in \mathbb{R}, \tag{34}$$

for all $v \in V(\Gamma)$. In other words, the weighted sum of the directed residuals at any vertex must be real. This condition may be seen as a generalization of the Kirchhoff current law applicable to the group $\mathbb{T}$ and circular statistics. The ML estimate for $\boldsymbol{x}$ will be among the solutions of (34). In contrast to the situation for $\mathbb{R}^d$, there are now a number of critical points. In consequence, to obtain a reliable fast estimator, it is necessary to distinguish the global maxima from the other critical points.
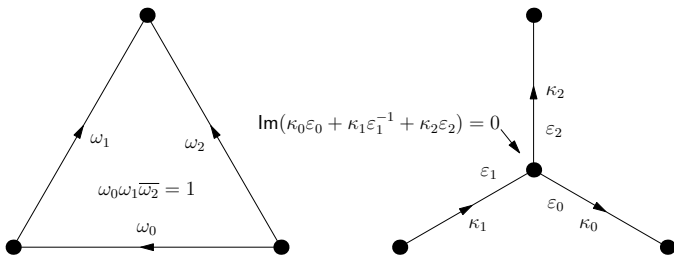
Fig. 5. Schematic of Kirchhoff laws for the group $\mathbb{T}$. At a maximum of the likelihood (32) the residual ε satisfies Kirchhoff's current law in the form (33).

Writing

$$\tilde{\kappa}_v = \frac{1}{d_v} \sum_{e \in E(v)} \kappa_e,$$

so that $\tilde{\kappa}_v d_v$ measures the strength of the connection of the vertex v to the rest of the network, reveals that what distinguishes the ML estimate from the other critical points of the likelihood function is that, for moderate noise values,

$$\frac{1}{d_v} \sum_{e \in E(v)} \frac{\kappa_e}{\tilde{\kappa}_v} (\overline{\omega}_e r_e)^{D_{ve}} \approx 1 \qquad (35)$$

In terms of the augmented adjacency matrix

$$A_{vv'} = \begin{cases} \kappa_{(v,v')} r_{(v,v')} & \text{if } (v, v') \in E(\Gamma) \\ \kappa_{(v,v')} r_{(v,v')}^{-1} & \text{if } (v', v) \in E(\Gamma) , \\ 0 & \text{otherwise} \end{cases} \qquad (36)$$

equation (34) can be reformulated with the aid of (31) as

$$Q\boldsymbol{x} = N^{-1} A \boldsymbol{x} = G\boldsymbol{x}, \qquad (37)$$

where $N_{v,v'} = \tilde{\kappa}_v d_v \delta_{v,v'}$ and G is a real diagonal matrix. The approximation (35) corresponds to $\tilde{\kappa}_v d_v \approx \rho_v$ or $G \approx \mathbb{I}$. When $\kappa_e = \kappa$ for all $e \in E(\Gamma)$, the matrix N is κ times the degree matrix.

A fast estimator for $\boldsymbol{x}$ may be obtained as follows. First, find the eigenvector $\boldsymbol{y} \in \mathbb{C}^n$ corresponding to the largest eigenvalue of Q,

$$Q\boldsymbol{y} = \lambda \boldsymbol{y} \qquad (38)$$

Then $\hat{x}_v = y_v/|y_v|$ for all $v \in V(\Gamma)$. The structure of this estimator is elucidated by consideration of its local implementation, which can be obtained by applying the power method [9] to (38). This involves application of the matrix Q repeatedly to a random initial vector $\boldsymbol{y}^{(0)}$. The method results in convergence to an eigenvector of Q corresponding to the eigenvalue of largest magnitude provided this is the only eigenvalue of largest magnitude and the initial $\boldsymbol{y}^{(0)}$ is not orthogonal to a left eigenvector of Q. In practice, to avoid numerical overflow, the resultant vector is re-normalized after each application of Q. This is a critical point in turning (38), or any other eigenvector based estimator, into an algorithm that can be run locally on the network. The normalization of $\boldsymbol{y}$ is a global operation since it involves each vertex knowing the values of $\boldsymbol{y}$ across the entire network. Without the ability to normalize vector, it is necessary to ensure that the eigenvalue of largest magnitude is close enough to unity that the iteration

converges well before an overflow or underflow condition is reached. The algorithm proposed here indeed has the property that the largest magnitude eigenvalue λ is approximately one. Further, as demonstrated in the simulations below, it operates reliably to align the network without renormalization.

To examine the local operation of this algorithm more closely, write $y_v = a_v x_v$ where a_v is a real amplitude and $x_v \in \mathbb{T}$ for each $v \in V(\Gamma)$. In addition to its phase $x_v \in \mathbb{T}$, each vertex keeps a circular variance $0 \le a_v \le 1$ which measures the degree to which it agrees with its neighbors. The local update rule for the case $\kappa_e = \kappa$ for all $e \in E(\Gamma)$ is

$$a_v^{(m+1)} x_v^{(m+1)} = \frac{1}{d_v} \sum_{u \sim v} a_u^{(m)} (x_u^{(m)} r_{(u,v)})^{D_{v,(u,v)}} \qquad (39)$$

Thus, at each update, the vertex v resets to the weighted circular mean of what its nearest neighbors predict its phase should be (compare this to Jacobi's method for alignment in $\mathbb{R}^d$). The weighting is based on how well each of the neighbors are aligned with their own neighbors. In this way, the contributions of vertices that have not yet converged are discounted relative to the ones from vertices that have converged to alignment with their neighbors. More generally, the update takes the form

$$a_v^{(m+1)} x_v^{(m+1)} = \frac{1}{d_v} \sum_{u \sim v} \frac{\kappa_{(u,v)}}{\tilde{\kappa}_v} a_u^{(m)} (x_u^{(m)} r_{(u,v)})^{D_{v,(u,v)}}$$

The convergence rate of the power method algorithm depends on the distance between the largest and next-to-largest magnitude eigenvalues. The Gershgorin circle theorem [9] implies that

$$-1 \le \lambda \le 1$$

The impediment to convergence comes from eigenvalues of Q near -1. Regularization can alleviate this to some extent. Consider the matrix

$$Q_\beta = (N + \beta I)^{-1} (A + \beta \mathbb{I})$$

where $\beta \ge 0$ is a regularization parameter. If $\boldsymbol{y}$ satisfies

$$Q_\beta \boldsymbol{y} = \lambda \boldsymbol{y}$$

then

$$Q\boldsymbol{y} = \lambda \boldsymbol{y} + \beta(\lambda - 1) N^{-1} \boldsymbol{y} \approx \lambda \boldsymbol{y}$$

Thus, if the maximum magnitude eigenvalue is close to one, the regularization leaves it close to 1. For the regularized eigenvalue problem, Gershgorin's theorem implies

$$-1 + \frac{2\beta}{\tilde{\kappa}_{\max} + \beta} \le \lambda \le 1$$

where $\tilde{\kappa}_{\max}$ is the largest element of the matrix N. Thus the parameter β can be used to bound the smallest eigenvalue away from -1, the effectiveness being mitigated by any vertex having a large value of $\tilde{\kappa}_v$.

This estimator exhibits remarkably good performance. In cases tested, the estimator gives a mean circular error performance indistinguishable from that of actual maximum likelihood, even when $\kappa \approx 1$. Although it seems to have no real bearing on practical estimation performance, it is possible to construct estimators that give values closer to

the maximum-likelihood estimate. One example of such an estimator, the hybrid maximum-likelihood algorithm, starts by running a suitable number of iterations of the local algorithm just described. Then it switches to the following iteration

$$a_v^{(m+1)} x_v^{(m+1)} = \frac{1}{d_v} \left(\sum_{u \sim v} \frac{\kappa_{(u,v)}}{\tilde{\kappa}_v^{(m)}} a_u^{(m)} (x_u^{(m)} r_e)^{D_{ve}} \right), \quad (40)$$

where

$$\tilde{\kappa}_v^{(m)} = \frac{1}{d_v} \operatorname{Re} \left((A\boldsymbol{x}^{(m)})_v / x_v^{(m)} \right).$$

At a particular vertex, the algorithm stops when the size of the imaginary component of $(A\boldsymbol{x}^{(m)})_v / x_v^{(m)}$ drops below some threshold. At this point, $\boldsymbol{x}$ satisfies the condition (37) with tolerance corresponding to the threshold, and thus a maximum of the likelihood function to this tolerance is obtained.

The algorithms described above were tested on two simulated networks, one with five nodes and the other with 31 nodes. The errors on each of the edges were identically distributed with concentration parameter κ. Two global algorithms were tested the where the eigenvectors corresponding to the largest eigenvalues of the matrices Q and A defined in (36) and (37) were computed. These were compared with the two local algorithms; i.e., the algorithm (39) for arriving at an the eigenvector of Q corresponding to the largest eigenvalue and the hybrid maximum-likelihood algorithm described in (40). These local algorithms were implemented on a network simulation with only nearest neighbor communication and with purely local stopping criteria. The results are shown in Figures 6 and 7, where the network structure is shown in the bottom left hand corner of each graph. The figures show the mean circular error across the network; i.e.,

$$CE(\hat{\boldsymbol{x}}) = 1 - \left| \frac{1}{|V(\Gamma) - 1|} \sum_{v \in V(\Gamma)/v_0} \hat{x}_v \overline{x}_v \right|^2$$

where x_v is the true value of the node offset. In each case, these results are compared with the the trace of the inverse of the Fisher information,

$$\operatorname{Tr} F^{-1} = \frac{I_0(\kappa)}{\kappa I_1(\kappa)} \operatorname{Tr} \tilde{L}.$$

These results indicate that the local Q eigenvector estimator works as well as the global methods, gives essentially maximum-likelihood performance, and lies very close to the trace of the inverse Fisher information.

VI. ABELIAN LIE GROUPS

This section briefly describes the situation in which the measurements reside in a general connected abelian Lie group. Such groups are just products of groups considered in the previous sections; i.e., G is of the form $\mathbb{R}^d \times \mathbb{T}^q$, and its elements may thus be written as $\boldsymbol{g} = (\boldsymbol{x}, \boldsymbol{z})$ where $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{z} = (e^{2\pi i \theta_k})_{k=1}^q$. Its dual group, which is the group of homomorphisms from G to $\mathbb{T}$, is identifiable with $\mathbb{R}^d \times \mathbb{Z}^q$, with elements $\boldsymbol{\tau} = (\boldsymbol{y}, \boldsymbol{n})$, via the pairing

$$\langle \boldsymbol{g}, \boldsymbol{\tau} \rangle = \langle (\boldsymbol{x}, \boldsymbol{z}), (\boldsymbol{y}, \boldsymbol{n}) \rangle = \exp i \left(\boldsymbol{x}.\boldsymbol{y} + 2\pi \sum_{k=1}^q n_k \theta_k \right).$$
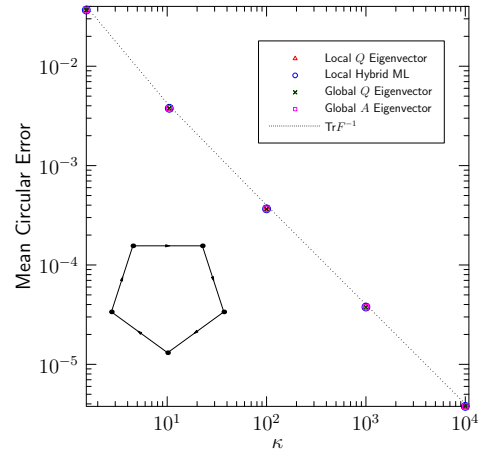

Fig. 6. Mean circular error versus concentration parameter κ for the four estimators operating on the five-node network shown in the inset.
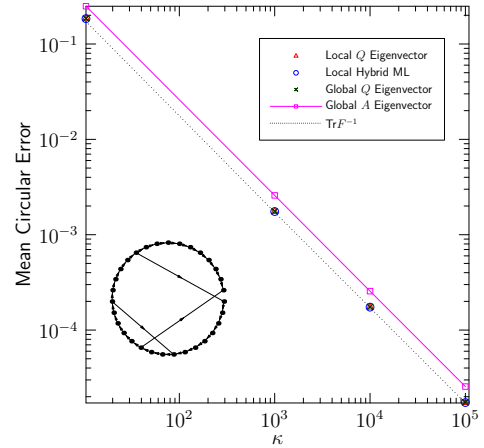

Fig. 7. Mean circular error versus concentration parameter κ for the four estimators operating on the 31-node network shown in the inset.

As in the formulations treated previously, the edges of the graph Γ are labeled with offsets that are elements of G. Rather than treating this case in complete detail, the focus of what follows in on synopsis of the aspects of the problem where significant modification of the approach described for earlier cases is needed to treat this more general situation.

A key issue in this setting is the choice of a suitable model for the distribution of the noise corrupting the edge labels. Little work has been done on probability distributions on abelian Lie groups. Even the situation of a q-torus, which is simply a product of circles, for $q > 2$ is little explored, although the case $q = 2$ is addressed in work of Singh *et al.* [15], in which a version of the von Mises distribution is used in connection with an application to estimation of torsion angles in complex molecules. Specifically, they use

$$\begin{aligned} p(\theta_1, \theta_2) = C \exp \big(&\kappa_1 \cos(\theta_1 - \mu_1) \\ &+ \kappa_2 \cos(\theta_2 - \mu_2) \\ &+ \lambda \sin(\theta_1 - \mu_1) \sin(\theta_2 - \mu_2) \big). \end{aligned} \quad (41)$$

The authors have studied this topic and intend to address it in a later paper, where a detailed analysis of appropriate error

distributions on such groups will be presented. For the present purpose, however, it suffices to assume sufficiently smooth densities $p_e(g) = p_e(\boldsymbol{x}, \boldsymbol{z})$ for $g = (\boldsymbol{x}, \boldsymbol{z}) \in G = \mathbb{R}^d \times \mathbb{T}^q$ and $e \in E(\Gamma)$ to allow the Fisher information to be calculated. Also, for this discussion, attention is restricted to the case where noise on distinct edges is independent.

The methods adopted in the previous examples can be mimicked and generalized as follows, where now the group operation is written additively (so that the circle group $\mathbb{T}$ is represented as the real numbers modulo 2π). The data vector $\boldsymbol{r}$ is, therefore, given by

$$\boldsymbol{r}_e = \boldsymbol{\omega}_e + \boldsymbol{\varepsilon}_e$$
$$= \boldsymbol{x}_{t(e)} - \boldsymbol{x}_{s(e)} + \boldsymbol{\varepsilon}_e$$

for each $e \in E(\Gamma)$. In this expression, $\boldsymbol{\omega}$ is parameterized in terms of the vertex offsets $\boldsymbol{x}$. The probability density for $\boldsymbol{\varepsilon}$ is

$$f(\boldsymbol{\varepsilon}) = \prod_{e \in E(\Gamma)} f_e(\boldsymbol{\varepsilon}_e).$$

The density for the noisy measurements is

$$f(\boldsymbol{r}|\boldsymbol{x}) = \prod_{e \in E(\Gamma)} f_e(\boldsymbol{r}_e - \boldsymbol{x}_{t(e)} + \boldsymbol{x}_{s(e)})$$

and, for given data $\boldsymbol{r}$, the log-likelihood function is

$$\ell(\boldsymbol{r}|\boldsymbol{x}) = \sum_{e \in E(G)} \log f_e(\boldsymbol{r}_e - \boldsymbol{x}_{t(e)} + \boldsymbol{x}_{s(e)})$$

where $\boldsymbol{r}_e$ and $\boldsymbol{x}_e$ are elements of G. It remains to define an "edge Fisher information" $\mathcal{F}_e$ as

$$\mathcal{F}_e = -\mathsf{E}\left\{\boldsymbol{\nabla}^2_{\boldsymbol{\omega}_e} \log p_e(\boldsymbol{r}_e - \boldsymbol{\omega}_e)\right\}$$

where $\boldsymbol{\nabla}^2$ involves partial derivatives in each of the real and circular coordinates of G. Observe that $\mathcal{F}_e$ is a $(d+q) \times (d+q)$ matrix indexed by the coordinates of G.

By analogy to the case of $\mathbb{R}^d$, the Fisher information matrix for $\boldsymbol{x}$ is given by

$$F^W = -\mathsf{E}\left\{\boldsymbol{\nabla}^2_{\boldsymbol{x}} \log p(\boldsymbol{r}|\boldsymbol{\omega})\right\}$$
$$= (\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega})^\mathsf{T} \mathcal{F} \boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega}$$

where $\mathcal{F}$ is a $m(d+q) \times m(d+q)$ matrix (m is the number of edges in the graph) with components

$$\mathcal{F}_{(e,s),(e',t)} = \delta_{e,e'}[\mathcal{F}_e]_{s,t}$$

for $s, t = 1, \cdots, d+q$. Using

$$\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{\omega} = \mathbb{I} \otimes D_W^\mathsf{T},$$

gives

$$F^W = (\mathbb{I} \otimes D_W)\mathcal{F}(\mathbb{I} \otimes D_W^\mathsf{T}).$$

In a way that closely parallels the multi-dimensional Gaussian case treated in Section III-D (cf. equation (22)), the determinant of the Fisher information in this case is

$$\det F^W = \sum_{\mathcal{S}} \det(\mathcal{F}_{\mathcal{SS}}),$$

where the sum is over all multi-spanning trees $\mathcal{S} = (S_1, S_2, \cdots, S_{d+q})$ consisting of one spanning tree for each dimension of G.

VII. Discussion and Conclusions

The preceding sections have introduced a statistical framework for registration and synchronization of data collected at the nodes of a network. The approach, which formulates the registration problem as one of optimally estimating true offsets between data at communicating nodes from noisy measurements of these values. Explicit estimators were derived and analyzed for the case in which the data reside in $\mathbb{R}^d$ and the noise corrupting the measurements is Gaussian. These solutions were pointed out to have a homological character leading to conditions akin to Kirchhoff's laws that optimal estimates and their residual errors must satisfy. They were shown to provide insight about how network topology can be designed and adapted to promote accurate synchronization across the network.

The phase alignment problem in which the data are on the circle $\mathbb{T}$ has also be treated explicitly and has been pointed out to manifest the critical properties of the more general case in which the data belong to a connected abelian Lie group. Further, iterative local algorithms, in which nodes only make use of information from their nearest neighbors in each iteration, have been described and empirically demonstrated.

Important practical cases, including alignment of local coordinate systems, involve non-abelian Lie groups. Application of the approach set forth here to this class of problems will be treated in a sequel to this paper.

References

[1] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, 1978.

[2] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 138–149.

[3] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networks*, vol. 13, no. 2, pp. 384–397, 2005.

[4] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, December 2006, pp. 4915–4920.

[5] K.-L. Noh, Y.-C. Wu, K. Qaraqe, and B. W. Suter, "Extension of pairwise broadcast clock synchronization for multicluster sensor networks," *EURASIP Journal on Advances in Signal Processing*, pp. 1–10, 2008.

[6] B. Bollobás, *Modern Graph Theory*, ser. Graduate Texts in Mathematics 184. Springer-Verlag, 1998.

[7] S. Smale, "On the mathematical foundations of electrical circuit theory," *Journal of Differential Geometry*, vol. 7, pp. 193–210, 1972.

[8] D. Jungnickel, *Graphs, Networks and Algorithms*, 3rd ed., ser. Algorithms and Computation in Mathematics. Springer-Verlag, 2008.

[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1993.

[10] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[11] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[12] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, October 2006.

[13] J. K. Johnson, D. Bickson, and D. Dolev, "Fixing the convergence of the Gaussian belief propagation algorithm," in *IEEE International Symposium on Information Theory*, 2009.

[14] K. V. Mardia and P. Jupp, *Directional Statistics*, 2nd ed. John Wiley and Sons, 2000.

[15] H. Singh, V. Hnizdo, and E. Demchuk, "Probabilistic model for two dependent circular variables," *Biometrika*, vol. 89, no. 3, pp. 719–723, 2002.

[16] D. Blatt and A. O. Hero III, "Distributed maximum likelihood estimation for sensor networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004.

[17] D. Bump, *Lie Groups*, ser. Graduate Texts in Mathematics 225. New York: Springer-Verlag, 2004.

[18] C. Chevalley, *Theory of Lie Groups*, ser. Princeton Landmarks in Mathematics. Princeton University Press, 1946.

[19] F. R. K. Chung, *Spectral Graph Theory*, ser. CBMS Number 92. American Mathematical Society, 1991.

[20] R. Diestel, *Graph Theory*, 3rd ed., ser. Graduate Texts in Mathematics 173. Heidelberg: Springer-Verlag, 2005.

[21] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, ser. Graduate Texts in Mathematics 222. New York: Springer-Verlag, 2003.

[22] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff, *Combinatorics and Graph Theory*, 2nd ed., ser. Undergraduate Texts in Mathematics. Springer, 2008.

[23] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum interpretation and analysis of Gaussian belief propagation," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. MIT Press, 2006, pp. 579–586.

[24] E. B. Sudderth, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," Master's thesis, MIT, 2002.

[25] M. J. Wainwright, E. B. Sudderth, and A. S. Willsky, "Tree-based modeling and estimation of Gaussian processes on graphs with cycles," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and V. Tresp, Eds. Cambridge MA: MIT Press, May 2001.