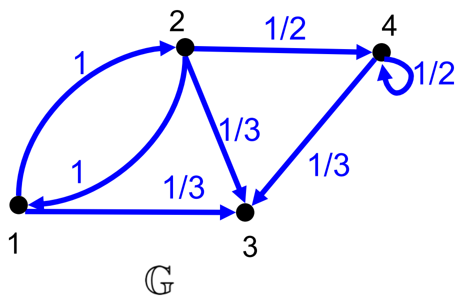
Lecture: Incidence Matrices & Rigidity Matrix

Shaoshuai Mou

Assistant Professor
School of Aeronautics and Astronautics
PURDUE UNIVERSITY

Review



$$\mathbb{G} = \{\mathcal{V}, \mathcal{E}\} \qquad \mathcal{V} = \{1,2,3,4\}$$

$$\mathcal{E} = \{(1,2),(1,3),(2,1),(2,3),(2,4),(4,3),(4,4)\}$$

- **Adjacency Matrix:**

$$A_{\mathbb{G}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \\ 0 & 1/2 & 0 & 1/2 \end{bmatrix} \qquad A_{\mathbb{G}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$A_{\mathbb{G}} = [a_{ij}]_{n\times n} \qquad a_{ij} = \begin{cases} w_{ij}, & j \to i; \\ 0, & \text{otherwise} \end{cases}$$

$$D_{\mathbb{G}} = \mathrm{diag}(A_{\mathbb{G}}\mathbf{1}) \qquad D_{\mathbb{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad D_{\mathbb{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

- **Laplacian Matrix:**

$$L_{\mathbb{G}} = D_{\mathbb{G}} - A_{\mathbb{G}} \qquad L_{\mathbb{G}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1/3 & -1/3 & 1 & -1/3 \\ 0 & -1/2 & 0 & 1/2 \end{bmatrix} \qquad L_{\mathbb{G}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad (A_{\mathbb{G}}x)_k = \sum_{j\in\mathcal{N}_k} w_{kj}x_j \qquad (L_{\mathbb{G}}x)_k = \sum_{j\in\mathcal{N}_k} w_{kj}(x_k - x_j)$$

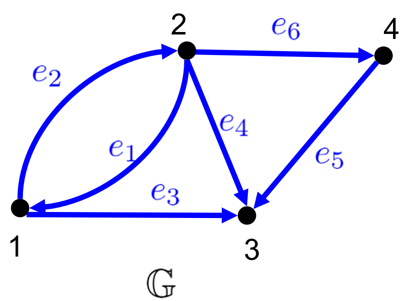
Incidence Matrix

of an *n*-node-*m*-edge directed graph

self-arcs are excluded

$$H = [h_{ik}]_{n\times m} = \begin{cases} 1, & \text{node } i \text{ is the } \textbf{head} \text{ of edge } k; \\ -1, & \text{node } i \text{ is the } \textbf{tail} \text{ of edge } k; \\ 0, & \text{otherwise} \end{cases}$$

Label all edges



How many vertices? $n = 4$

How many edges? $m = 6$

Try by yourself

$$H = \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} & \text{Nodes} \\ \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}_{4\times 6} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

- At each column of H, there is one entry equal to 1, one entry equal to -1, and all other entries are 0s.

$$\mathbf{1}'H = 0$$

- The *k*th column of **H** corresponds to the *k*th edge $i \to j$ with the *i*th entry -1 and *j*th entry 1.

$$(H'x)_k = x_j - x_i$$

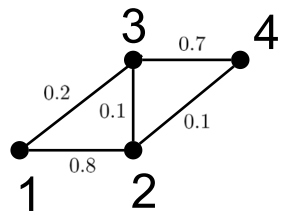
$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$H'x =$$

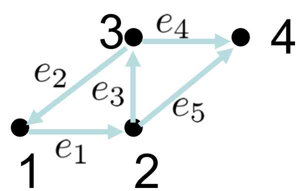
Try by yourself

Incidence Matrix

of an *n*-node-*m*-edge undirected graph



- Assign an arbitrary direction for each edge.



$$H = [h_{ik}]_{n\times m} = \begin{cases} 1, & \text{node } i \text{ is the } \textbf{head} \text{ of edge } k; \\ -1, & \text{node } i \text{ is the } \textbf{tail} \text{ of edge } k; \\ 0, & \text{otherwise} \end{cases}$$

$$H = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}_{4\times 5} \qquad L = \begin{bmatrix} 1 & -0.8 & -0.2 & 0 \\ -0.8 & 1 & -0.1 & -0.1 \\ -0.2 & -0.1 & 1 & -0.7 \\ 0 & -0.1 & -0.7 & 0.8 \end{bmatrix}$$

- Verify in Matlab $L = HDH'$ $\qquad D = \mathrm{diag}\,\{w_1, w_2, \ldots, w_m\}$

- Prove $\mathrm{rank}(H) = \mathrm{rank}(L) = n - c$ $\qquad$ c: the number of connected components

$$\ker(HDH') = \ker(H')$$

$$\ker D^{1/2}H'$$

What is the rank of H for a connected graph?

What is the rank of H for a tree graph?

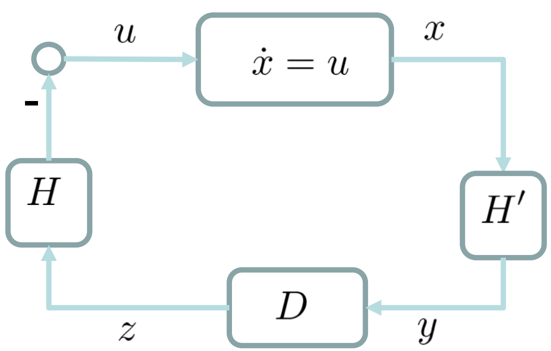
Representation of the Laplacian flow

$$\boxed{\dot{x} = -Lx} \xleftarrow{L = HDH'} \dot{x} = -HDH'x$$

- The plant: $\dot{x}_i = u_i \qquad i = 1, 2, \ldots, n$
- Measurements: $y_{ij} = x_i - x_j \qquad i \to j$
- Control gains: $z_{ij} = a_k y_{ij} \qquad i \to j$
- Feedback controls: $u_i = -\sum_{(i,j)\in\mathbb{G}} z_{ij} \qquad i = 1, 2, \ldots, n$

$$\boxed{\begin{aligned} \dot{x} &= u \\ y &= H'x \\ z &= Dy \\ u &= -Hz \end{aligned}}$$

$D = \mathrm{diag}\ \{a_1, a_2, \ldots, a_m\}$



Nodes H Edges

States **Relative** States

Pseudo-inverse of a matrix

- The inverse of a full rank matrix $M \in \mathbb{R}^{n\times m}$ is defined as

$M^{-1}M = I_m$ left inverse

may not exist L

$MM^{-1} = I_n$ right inverse

- For $M \in \mathbb{R}^{n\times m}$, its **pseudo**-inverse $M^\dagger$

  is the unique $m \times n$ matrix such that
$$\begin{cases} MM^\dagger M = M \\ M^\dagger M M^\dagger = M^\dagger \\ MM^\dagger, \quad M^\dagger M \text{ are both symmetric} \end{cases}$$

  Matrix inverse is always a pseudo-inverse!

❖ ***For an undirected connected graph, what is the pseudo-inverse of its Laplacian?***

L is symmetric $L = U\mathrm{diag}\{0, \lambda_2, ..., \lambda_n\}U'$ Columns of U are **orthonormal eigenvectors** of L.

$U'U = I$

$$\boxed{L^\dagger = U\mathrm{diag}\{0, \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_n}\}U'}$$

$$\boxed{L^\dagger \mathbf{1} = 0 \qquad LL^\dagger = L^\dagger L = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n}$$

$$= U(I - e_1 e_1')U'$$

$$= I - (Ue_1)(Ue_1)'$$

$$= I - \frac{\mathbf{1}}{\sqrt{n}}\frac{\mathbf{1}'}{\sqrt{n}}$$

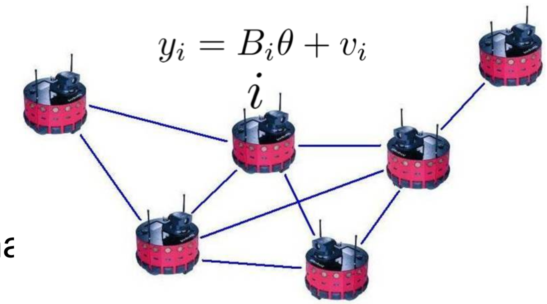
Verify the three conditions

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Ue_1 = \frac{1}{\sqrt{n}}\mathbf{1}$$

Application of Distributed Averaging in **Distributed Estimations**

- Utilize a multi-agent network to achieve an important parameter vector θ , which is not directly observable/available.



- **Local Measurement**: Each agent *i* observes/measures a linear combina

$$y_i = \underbrace{B_i}_{\text{known to } i}\theta + \underbrace{v_i}_{\text{noise}}$$

$v_1, v_2, \ldots, v_m$ are independent jointly-Gaussia

$E[v_i] = 0 \qquad E[v_i v_i'] = \Lambda_i = \Lambda_i'$

- **Global Goal**: Achieve a *nice estimate* to θ

$\hat{\theta}^*$ minimizes the following objective function

convex

$$F(\hat{\theta}) = \sum_{i=1}^{n} \underbrace{(y_i - B_i\hat{\theta})'}_{\text{estimation error}} \Lambda_i^{-1} (y_i - B_i\hat{\theta})$$

quadratic form $x'Ax$

each agent's estimation error is weighted by Λ_i^{-1} Accurate (inaccurate) measurements are with high (low) weights.

$$\frac{\partial F}{\partial \theta}\Big|_{\hat{\theta}=\hat{\theta}^*} = 0$$

$$\sum_{i=1}^{n} B_i'\Lambda_i^{-1}(y_i - B_i\hat{\theta}^*) = 0$$

$$\sum_{i=1}^{n} B_i'\Lambda_i^{-1}B_i\hat{\theta}^* = \sum_{i=1}^{n} B_i'\Lambda_i^{-1}y_i$$

$$\hat{\theta}^* = \left(\frac{1}{n}\sum_{i=1}^{n} B_i'\Lambda_i^{-1}B_i\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} B_i'\Lambda_i^{-1}y_i\right)$$

❖ **Conclusion:** Note each agent i knows B_i, Λ_i, y_i

Thus, these two terms could be achieved by distributed averaging.

Estimation from Relative Measurements

difference between corresponding variables
(relative positions of robots; clock synchronizations)

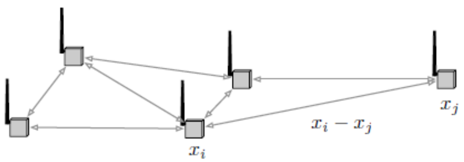
- Relative Measurements:

For the *k*th edge $j \longrightarrow i$, let

$$y_k = x_i - x_j + v_k$$
$$= (H'x)_k + v_k$$

Independent jointly-Gaussian

$0 \quad \sigma_k^2$



The optimal estimation based on available measurements for x is

$$x^* = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} |H'x - y|^2_{\Sigma^{-1}} \qquad \Sigma = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, ..., \sigma_m^2\}$$

Since no absolute information is available, we add one additional constraint (zero mean)

$$x^* = \arg\min_{\mathbf{1}'x=0} \frac{1}{2} |H'x - y|^2_{\Sigma^{-1}}$$

Centralized Computation:

$$\ker(\begin{bmatrix} L \\ \mathbf{1}' \end{bmatrix}) = 0$$

$$\frac{\partial \frac{1}{2} |H'x^* - y|^2_{\Sigma^{-1}}}{\partial x} = 0$$

$$H\Sigma^{-1}(H'x^* - y) = 0$$

$$\underbrace{H\Sigma^{-1}H'}_{L}x^* = H\Sigma^{-1}y$$

$$Lx^* = H\Sigma^{-1}y$$
$$\mathbf{1}'x^* = 0$$

$$x^* = L^\dagger H\Sigma^{-1}y$$

Unique solution

$$LL^\dagger = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n \qquad \mathbf{1}'L^\dagger = 0$$

Distributed Estimation to achieve $x^* = L^\dagger H \Sigma^{-1} y$

$$x_i(k+1) = x_k(k) - \gamma \sum_{j \in \mathcal{N}_i} \frac{1}{\sigma_{ij}^2} (x_i(k) - x_j(k) - y_{ij})$$

$x_i(0) = 0$ $\quad$ γ is sufficiently small

$$x(k+1) = (I - \gamma L)x(k) + \gamma H \Sigma^{-1} y$$

$x(0) = 0$ $\quad$ $L = H \Sigma^{-1} H'$

$x(k) \to x^*$ exponentially fast

$$Lx^* = H\Sigma^{-1} y$$

$$\mathbf{1}' x^* = 0$$

$$\epsilon(k) = x(k) - x^*$$

$$\begin{aligned}
\epsilon(k+1) &= (I - \gamma L)x(k) + \gamma H \Sigma^{-1} y - (I - \gamma L + \gamma L)x^* \\
&= (I - \gamma L)\epsilon(k) + \gamma(H\Sigma^{-1} y - Lx^*) \\
&= (I - \gamma L)\epsilon(k)
\end{aligned}$$

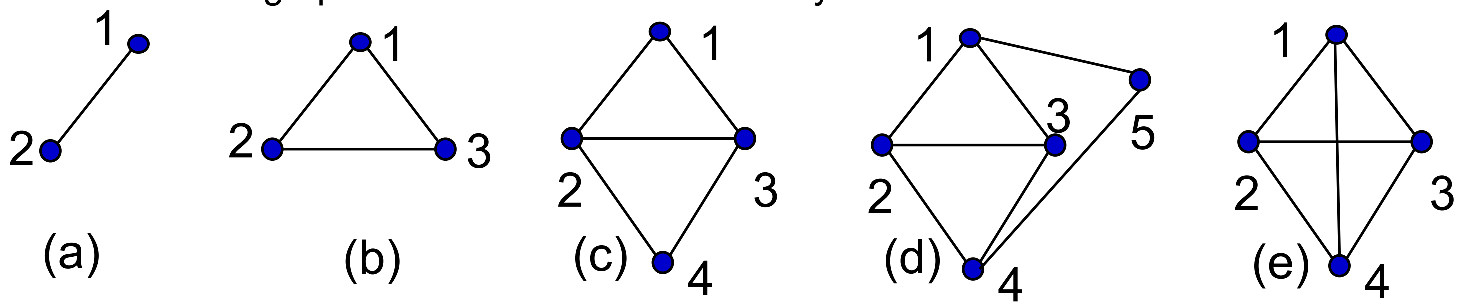
Non-negative; doubly stochastic; symmetric.

Strongly connected; self-arcs

$$\begin{aligned}
\epsilon(k) &\to (\frac{1}{n}\mathbf{1}'\epsilon(0))\mathbf{1} \\
&= \frac{1}{n}\mathbf{1}'(x(0) - x^*) = 0
\end{aligned}$$

Rigid Graph

A graph that can not be deformed by continuous motions.



- A **minimally rigid** graph is a rigid graph and deletion of any edge will violate the rigidity.

a,b,c,d are minimally rigid; e is not.

- A **rigid** graph is graph which contains a minimally rigid graph as a *spanning subgraph*.

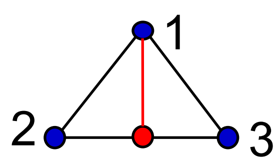
(Same vertex set;Subset of edges)

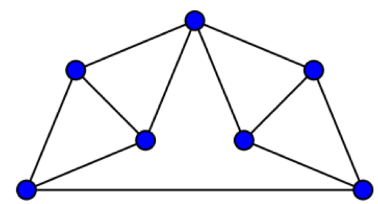
c is a spanning subgraph of e

❖ How to produce a minimally rigid graph in 2D?

- Vertex Addition: Add a new vertex by connecting it to two other vertices by two new edges. a,b,c,d

Henneberg Operations





The Moser spindle

- Edge Splitting: Insert a new vertex into one edge to split it into two and also connect it to another node.

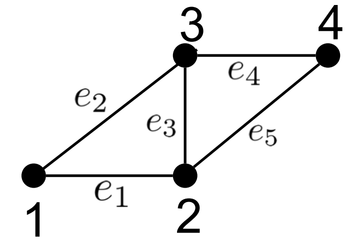
How many edges are there for a minimally rigid graph in 2D? $2n-3$

Henneberg Operations provide a **geometric way** to determine whether a graph is rigid.

Is there any algebraic way? Since computers usually do not understand geometric shapes but matrices.

Rigidity Matrix

$x_i \in \mathbb{R}^2$

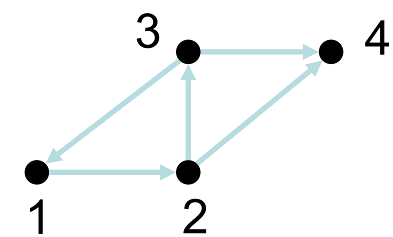


$$R(x) = \begin{array}{c} e_1 \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} x_1' - x_2' & x_2' - x_1' & 0 & 0 \\ x_1' - x_3' & 0 & x_3' - x_1' & 0 \\ 0 & x_2' - x_3' & x_3' - x_2' & 0 \\ 0 & 0 & x_3' - x_4' & x_4' - x_3' \\ 0 & x_2' - x_4' & 0 & x_4' - x_2' \end{bmatrix}_{m \times 2n}$$

(infinitesimally) rigid $\quad \text{rank } R = 2n - 3$

(infinitesimally) minimally rigid: full row rank

- Connection to incidence matrix
- Assign an arbitrary direction fro each edge;



$$H = [h_{ik}]_{n \times m}$$

$$= \begin{cases} 1, & \text{node } i \text{ is the } \textbf{head} \text{ of edge } k; \\ -1, & \text{node } i \text{ is the } \textbf{tail} \text{ of edge } k; \\ 0, & \text{otherwise} \end{cases}$$

$$H = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 \\ 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}_{4 \times 5}$$

For the kth edge from i to j, one define $z_k = x_j - x_i$ $\qquad Z = \text{diag}\{z_1, z_2, \ldots, z_m\}$

$$R = Z'_{2m \times m}(H'_{n \times m} \otimes I_2)$$

- Kronecker Product $\otimes$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix},$$

$$H \otimes I_2 = \begin{bmatrix} I_2 & -I_2 & 0 & 0 & 0 \\ -I_2 & 0 & I_2 & 0 & I_2 \\ 0 & I_2 & -I_2 & I_2 & 0 \\ 0 & 0 & 0 & -I_2 & -I_2 \end{bmatrix}_{8 \times 10}$$