Connectivity, Rigidity and Online Decentralized Maintenance Methods

Antonio Franchi

CNRS, LAAS, France, Europe

2015 IROS Workshop on ‘On-line decision-making in multi-robot coordination’ (DEMUR’15)
Hamburg, Germany
12th October, 2015
1. Graphs, Matrices, and Eigenvalues

2. Connectivity vs Infinitesimal Rigidity

3. Maintenance Problems and Methods

4. Handling Multiple Objectives in Maintenance Problems

5. Applications
Partial list:


If you want to know more about what follows:


Graphs, Matrices, and Eigenvalues
A Graph models an Adjacency Structure

\[ [(i, j)] \in E \iff \text{vertexes } i \text{ and } j \text{ are neighbors or adjacent} \]

- \((i, j), i < j\) representative element of the equivalence class \([(i, j)]\)
  \[ [\mathcal{V} \times \mathcal{V}] = \{(1, 2), (1, 3), \ldots, (1, N), \ldots, (N - 1, N)\} \]
  \[ = \{e_1, e_2, \ldots, e_{N-1}, \ldots, e_{N(N-1)/2}\} \]
- \([(i, i)] \notin E, \forall i \in \mathcal{V} \) (no self-loops)
- \(\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in E\} \) set of neighbors of \(i\)

\(\mathcal{G} = (\mathcal{V}, \mathcal{E})\) is an \textbf{undirected graph} or simply \textbf{graph}

- \(\mathcal{V} = \{1, \ldots, N\} \) vertex set
- \(\mathcal{E} \subset (\mathcal{V} \times \mathcal{V})/\sim \) edge set
- \(\sim\) equivalence relation identifying \((i, j)\) and \((j, i)\)
Incidence Matrix

\[ E \in \mathbb{R}^{N \times N(N-1)/2} \] is the (full) **incidence matrix** of \( G \)

\[ \forall e_k = (i, j) \in [\mathcal{V} \times \mathcal{V}]: \]

- \( E_{ik} = -1 \) and \( E_{jk} = 1 \), if \( e_k \in \mathcal{E} \)
- \( E_{ik} = 0 \) and \( E_{jk} = 0 \), otherwise

**Matricial representation** of a graph

\[ E = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 & 1 \\
0 & 0 & -1 & 0 & 0 & -1 \\
e_1 & e_2 & e_3 & e_4 & e_5 & e_6
\end{pmatrix} \]

remember:
\[ \{e_1, e_2, \ldots, e_{N-1}, \ldots, e_{N(N-1)/2}\} = \{(1, 2), (1, 3), \ldots, (1, N), \ldots (N-1, N)\} \]
Network of Robots in an Environment

Assume $N$ mobile robots moving in an environment:

- $x_i \in \mathbb{R}^{nx}$ $i$-th robot configuration, $i \in 1 \ldots N$
- $z \in \mathbb{R}^{nz}$ environment configuration

Consider two maps

robot map $v : \mathbb{R}^{nx} \ni x_i \mapsto v(x_i) = v_i \in \mathbb{R}^{nv}$

connection map $w : \mathbb{R}^{nx} \times \mathbb{R}^{nx} \times \mathbb{R}^{nz} \ni (x_i, x_j, z) \mapsto w(x_i, x_j, z) = w_{ij} \in \mathbb{R}_{\geq 0}$

with the properties

- $w_{ij} = w_{ji}$ (symmetry)
- $w_{ii} = 0$

example: what can those maps model?
The connection map \( w \) defines an **associated graph** \( G = (\mathcal{V}, \mathcal{E}) \), where

- \( \mathcal{V} = \{1, 2, \ldots, N\} \)
- \( \mathcal{E} = \{e_k = (i, j) \mid w_{ij} > 0\} \)
- the **positive weight** \( w_{ij} \) is associated to each edge \((i, j) \in \mathcal{E}\)

Both maps \( v \) and \( w \) define an **associated framework** \((G, v)\) where

- \( G \) is the associated graph
- \( v_i \) is associated to each vertex \( i \in \mathcal{V} \)
Adjacency/Weight Matrix

\[ A = \begin{pmatrix}
  0 & w_{12} & \ldots & w_{1N} \\
  w_{12} & 0 & \ldots & w_{2N} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{1N} & w_{2N} & \ldots & 0
\end{pmatrix} \in \mathbb{R}^{N \times N} \]

is the adjacency (or weight) matrix of \( G \).

Note that

- \( A_{ij} = 0 \) if \((i, j) \notin \mathcal{E}\)
- \( A_{ij} > 0 \) otherwise

Properties:

- **P.1** \( A = A(x_1, \ldots, x_N, z) \)
- **P.2** \( A \) is square
- **P.3** \( A_{ij} = A_{ij} \) (symmetric)
- **P.4** \( A_{ij} = A_{ij} \geq 0 \) (nonnegative)
- **P.5** \( A_{ii} = 0 \)

Example:

\[ A = \begin{pmatrix}
  0 & w_{12} & w_{13} & w_{14} \\
  w_{12} & 0 & w_{23} & 0 \\
  w_{13} & w_{23} & 0 & w_{34} \\
  w_{14} & 0 & w_{34} & 0
\end{pmatrix} \]
Laplacian Matrix

\[
L = \begin{pmatrix}
\sum_{j=1}^{n} w_{1j} & -w_{12} & \ldots & -w_{1N} \\
-w_{12} & \sum_{j=1}^{n} w_{j2} & \ldots & -w_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
-w_{1N} & -w_{2N} & \ldots & \sum_{j=1}^{n} w_{jN}
\end{pmatrix} \in \mathbb{R}^{N \times N}
\]

is the Laplacian matrix of \( G \)

Note that

- \( L = \text{diag}(\delta_i) - A \),

where \( \delta_i = \sum_{j=1}^{n} w_{ij} \)

(degree of vertex \( i \))

Properties:

P.1 \( L = L(x_1, \ldots, x_N, z) \)

P.2 \( L \) is square

P.3 \( L_{ij} = L_{ji} \) (symmetric)

Example:

\[
L = \begin{pmatrix}
w_{12} + w_{13} + w_{14} & -w_{12} & -w_{13} & -w_{14} \\
-w_{12} & w_{12} + w_{23} & -w_{23} & 0 \\
-w_{13} & -w_{23} & w_{13} + w_{23} + w_{34} & -w_{34} \\
-w_{14} & 0 & -w_{34} & w_{14} + w_{34}
\end{pmatrix}
\]
Connected Graph

Connectivity

$G$ is **connected** if there is a **path** between every pair of vertices, i.e.,

$$\forall i \in \mathcal{V} \text{ and } j \in \mathcal{V} \setminus i, \quad \exists \text{ a path (sequence of adjacent edges) from } i \text{ to } j$$

This is a **combinatorial definition** of connectivity

question: connectivity is a **global** property, what does it mean? and why it is global?
Importance of Connectivity

What connectivity can model?

- connected **communication** network
- connected **sensing** network
- connected **control** network
- connected **planning** roadmap

What connectivity is important for?

- pass a message from *any* robot to *any* other robot
- know the relative position between *any* two robots in a **common frame**
- converge to a **common point**
- **share** a common goal

Related concepts

- group, cohesiveness
- aggregation
- sharing
Additional properties of $L = \text{diag}(\delta_i) - A$

- $L$ is **positive semi-definite**, i.e., all the eigenvalues are real and non-negative

  $$0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$$

- $\sum_{j=1}^{n} L_{ij} = 0 \quad \forall i = 1 \ldots N$, i.e., $L1 = 0$, therefore

  $$\lambda_1 = 0 \text{ and it is associated to the eigenvector } 1 = (1 \ 1 \ \ldots \ \ 1)^T$$

(Fiedler 1973)

$\lambda_2 > 0$ if the graph $G$ is **connected** and $\lambda_2 = 0$ otherwise
$\lambda_2$ provides an **algebraic definition** of connectivity

$\Rightarrow \lambda_2$ is called *algebraic connectivity*, *connectivity eigenvalue*, or **Fiedler eigenvalue**

$\lambda_2 = \lambda_2(x_1, \ldots, x_N, z)$ is a **global** quantity

Example (if $w_{ij} \in \{0, 1\}$):

- $\lambda_2 = 4$
- $\lambda_2 = 2$
- $\lambda_2 = 0.58$
- $\lambda_2 = 0$
A framework of positions is a particular framework \((G, \mathbf{v})\) in the special case in which \(\mathbf{v} : \mathcal{V} \rightarrow \mathbb{R}^d\) maps each vertex to the position in \(\mathbb{R}^d\) of the \(i\)-th robot

- if \(d = 2\), \(\mathbf{v}_i = \mathbf{p}_i = \begin{pmatrix} p_{ix}^i \\ p_{iy}^i \end{pmatrix}\), 2D position of robot \(i\)

- if \(d = 3\), \(\mathbf{v}_i = \mathbf{p}_i = \begin{pmatrix} p_{ix}^i \\ p_{iy}^i \\ p_{iz}^i \end{pmatrix}\), 3D position of robot \(i\)

In the following

- it will be (mainly) \(d = 3\), similar results apply for \(d = 2\)
- we refer only to framework of positions, called simply frameworks
Consider two frameworks \((G, p')\) and \((G, p'')\)

- **same graph** \(G\)
- **different positions** \(p'\) and \(p''\)

Frameworks \((G, p')\) and \((G, p'')\) are

- **equivalent**: if \(\|p'_i - p'_j\| = \|p''_i - p''_j\|\) for all \((i, j) \in E\), and
- **congruent**: if \(\|p'_i - p'_j\| = \|p''_i - p''_j\|\) for all \((i, j) \in V \times V\)

---

**equivalent frameworks**

**congruent frameworks**
The framework \((G, p')\) is **globally rigid** if every other framework \((G, p'')\) which

- is equivalent to \((G, p'')\)

is also congruent to \((G, p')\)

This is, again, a **combinatorial definition**
The framework \((G, p')\) is **rigid** if \(\exists \epsilon > 0\) such that every other framework \((G, p'')\) which

- is equivalent to \((G, p'')\)
- satisfies \(\|p'_i - p''_i\| < \epsilon\) for all \(i \in V\),

is congruent to \((G, p')\). This is, again, a **combinatorial definition**.

**question:** is rigidity a global property of the graph as well?
Importance of Rigidity

What rigidity can model?

- rigid *mechanical structure* made of *bars*
  
  but also:

- rigid *sensing network*

- rigid *control network*

What rigidity is important for?

- *univocally* compute the arrangement (*shape*) of a group of robots only using *inter-distances*

- achieve (or track) a desired shape *only controlling the inter-distances* (formation control)

Related concepts

- parallel rigidity

- persistent graph

- tensegrity
question: do you know an example of use of rigidity in robotics?
Example of use of Rigidity

question: do you know an example of use of rigidity in robotics?

6-DOF **Stewart platform** parallel robot

Credits: Robert L. Williams II
Let’s give a definition of rigidity that is differential (\(\iff\) involves infinitesimal motions)

Consider a trajectory \(p(t)\) with \(t \geq t_0\) and impose equivalence along the trajectory:

\[
\|p_i(t) - p_j(t)\|^2 = \|p_i(t_0) - p_j(t_0)\|^2 = \text{const} \quad \text{for all } (i, j) \in \mathcal{E}, \quad \forall t \geq t_0
\]

Differentiating with respect to time the constraint above:

\[
(p_i(t) - p_j(t))^T (p_i'(t) - p_j'(t)) = 0 \quad \text{for all } (i, j) \in \mathcal{E}, \quad \forall t \geq t_0
\] (1)

**Trivial Motion**

A collective motion that consists of only global roto-translations of the whole set of positions in the framework

**Infinitesimal Rigidity**

The framework \((\mathcal{G}, p(t_0))\) is infinitesimally rigid if every possible motion that satisfies (1) is trivial
question: is this framework rigid in $\mathbb{R}^2$? is it infinitesimally rigid?
question: is this framework rigid in $\mathbb{R}^2$? is it infinitesimally rigid?

- infinitesimal rigidity $\Rightarrow$ rigidity
- rigidity $\not\Rightarrow$ infinitesimal rigidity
Let us write the infinitesimal rigidity constraint in a matricial form

\[(p_i(t) - p_j(t))^T (\dot{p}_i(t) - \dot{p}_j(t)) = 0 \quad \text{for all } (i, j) \in \mathcal{E}, \forall t \geq t_0\]

\[\updownarrow\]

\[w_{ij} (p_i(t) - p_j(t))^T (\dot{p}_i(t) - \dot{p}_j(t)) = 0 \quad \text{for all } e_k = (i, j) \in [\mathcal{V} \times \mathcal{V}], \forall t \geq t_0\]
Matricial Representation of Infinitesimal Rigidity

\[
0 = w_{ij} (p_i(t) - p_j(t))^T (\dot{p}_i(t) - \dot{p}_j(t)) = \\
= w_{ij} (p_i(t) - p_j(t))^T \dot{p}_i(t) - (p_i(t) - p_j(t))^T \dot{p}_j(t) = \\
= w_{ij} \begin{pmatrix}
-0^T & (p_i(t) - p_j(t))^T & -0^T & (p_j(t) - p_i(t))^T & -0^T
\end{pmatrix} \begin{pmatrix}
\dot{p}_1 \\
\vdots \\
\dot{p}_N
\end{pmatrix}
\]

\[
K_{ij} \in \mathbb{R}^{1 \times 3N}
\]

where \(0 = (0 \ 0 \ \ldots \ 0)^T\)
stacking the previous constraints for every \((i, j) \in \{e_1, e_2 \ldots e_{N-1} \ldots \ldots, e_{N(N-1)/2}\}:

\[
\begin{pmatrix}
    w_{12} & \cdots & w_{N(N-1)} \\
    \vdots & \ddots & \vdots \\
    w_{N(N-1)} & \cdots & w_{N(N-1)}
\end{pmatrix}
\begin{pmatrix}
    K_{12} \\
    \vdots \\
    K_{N(N-1)}
\end{pmatrix}
\begin{pmatrix}
    \dot{p}_1 \\
    \vdots \\
    \dot{p}_N
\end{pmatrix}
= \begin{pmatrix}
    W(w)K(p)
\end{pmatrix}
\dot{p} = R(w, p)\dot{p} = 0
\]

\[W(w) \in \mathbb{R}^{\frac{N(N-1)}{2} \times \frac{N(N-1)}{2}} \quad K(p) \in \mathbb{R}^{\frac{N(N-1)}{2}} \times 3N \quad \dot{p} \in \mathbb{R}^{3N}\]

Rigidity Matrix

\(R(w, p)\) is the (weighted) **rigidity matrix**
Example of Rigidity Matrix

\[ R(w, p) = \begin{pmatrix}
    w_{12}(p_1^x - p_2^x) & w_{12}(p_1^y - p_2^y) & \ldots & 0 \\
    w_{13}(p_1^x - p_3^x) & w_{13}(p_1^y - p_3^y) & \ldots & 0 \\
    w_{14}(p_1^x - p_4^x) & w_{14}(p_1^y - p_4^y) & \ldots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{34}(p_3^x - p_4^x) & w_{34}(p_3^y - p_4^y) & \ldots & \ldots \\
\end{pmatrix} \]

\[ d = 2 \ (\mathbb{R}^2) \]
\[ N = 4 \]
\[ N(N - 1)/2 = 6 \]
• rigidity is defined \textbf{combinatorially} ("...s.t. every other framework...")

• infinitesimal rigidity implies rigidity

• converse not true (degenerate cases) but...

• infinitesimal rigidity can be defined \textit{algebraically}, in fact...
• **collective roto-translations** in $\mathbb{R}^3$ keep constant all the distances, by definition, i.e., if $\dot{p}$ is trivial then $R(w, p)\dot{p} = 0$

• $\Rightarrow \text{Dim} (\ker[R(w, p)]) \geq 6$ always

• for infinitesimally rigid frameworks the motion that keep constant all the distances are only **collective roto-translations** in $\mathbb{R}^3$
  
  i.e., if $R(w, p)\dot{p} = 0$ then $\dot{p}$ is trivial

• infinitesimally rigidity $\Rightarrow \text{Dim} (\ker[R(w, p)]) = 6$

---

(Tay and Whiteley 1985) and (Zelazo et al. 2014)

A framework is infinitesimally rigid if and only if $\text{rank}[R(w, p)] = 3N - 6$

• despite its name, the rigidity matrix is actually characterizing **infinitesimal rigidity** (rather than **rigidity**)

---

Antonio Franchi  
29 of 69
Symmetric Rigidity Matrix

\[ S(w, p) = R(w, p)^T R(w, p) \in \mathbb{R}^{3N \times 3N} \] is the symmetric rigidity matrix

(Zelazo et al. 2014)

Properties:

P.1 \( S = S(w, p) = S(x_1, \ldots, x_N, z) \)

P.2 \( S \in \mathbb{R}^{3N \times 3N} \) (square)

P.3 \( S_{ij} = S_{ji} \) (symmetric)

P.4 \( \text{Dim (ker}[S(w, p)])) \geq 6 \)

(Zelazo et al. 2014)

A framework is infinitesimally rigid if and only if \( \text{rank}[S(w, p)] = 3N - 6 \)
Additional properties of $S = R^T R$

- $S$ is **positive semi-definite**, i.e., all the eigenvalues are real and non-negative

  $$0 \leq \varsigma_1 \leq \varsigma_2 \leq \ldots \leq \varsigma_6 \leq \varsigma_7 \leq \ldots \leq \varsigma_{3N}$$

- $\text{Dim} (\ker[S(w, p)]) \geq 6$, therefore

  $$\varsigma_1 = \varsigma_2 = \varsigma_3 = \varsigma_4 = \varsigma_5 = \varsigma_6 = 0$$

(Zelazo et al. 2014)

$\varsigma_7 > 0$ if the framework is **infinitesimally rigid** and $\varsigma_7 = 0$ otherwise

$\varsigma_7$ provides an **algebraic definition** of infinitesimal rigidity

$\Rightarrow \varsigma_7$ is called the **rigidity eigenvalue** (Zelazo et al. 2014)

$\varsigma_7 = \varsigma_7(x_1, \ldots, x_N, z)$ is a **global** quantity
Connectivity vs Infinitesimal Rigidity
Similarities between Connectivity and Infinitesimal Rigidity

**Connectivity**

∃ a path between any pair of vertexes

- depends on \( x_1, \ldots, x_N, z \) (global property)
- Laplacian matrix \( L \in \mathbb{R}^{N \times N} \)
- \( \Leftrightarrow \) Fidler eigenvalue \( \lambda_2 > 0 \)

**Infinitesimal rigidity**

distance-preservation on the edges forces a trivial (roto-translational) movement

- depends on \( x_1, \ldots, x_N, z \) (global property)
- symmetric rigidity matrix \( S \in \mathbb{R}^{3N \times 3N} \)
- \( \Leftrightarrow \) rigidity eigenvalue \( \varsigma_7 > 0 \)
(Infinitesimal) Rigidity $\Rightarrow$ Connectivity, i.e., $\varsigma_7 > 0 \Rightarrow \lambda_2 > 0$

In fact, e.g., by contradiction:

- not connected implies at least two connected components
- distance between the two connected components can change still preserving equivalence

$\Rightarrow$ by enforcing infinitesimal rigidity one enforces connectivity as well
### Differences between Connectivity and Infinitesimal Rigidity

<table>
<thead>
<tr>
<th>Connectivity</th>
<th>Infinitesimal rigidity</th>
</tr>
</thead>
<tbody>
<tr>
<td>• applicable to any graph</td>
<td>• applicable only to frameworks (graphs + positions)</td>
</tr>
<tr>
<td>• depends only on (w)</td>
<td>• depends both on (w) and (v = p)</td>
</tr>
<tr>
<td>• (\not\Rightarrow) infinitesimal rigidity</td>
<td>• (\Rightarrow) connectivity</td>
</tr>
</tbody>
</table>

Infinitesimal rigidity is a **stronger property** and applies to a **more particular** structure (framework).
Maintenance Problems and Methods
Assume each robot $i = 1, \ldots, N$

- can **control** $x_i(t)$, $\forall t \geq t_0$ (with $x_i(t)$ smooth enough)
- has some objectives (**mission**)

### Maintenance problem(s)

- assume $\mathcal{G}$ is connected (or $(\mathcal{G}, p)$ is infinitesimally rigid) for $t = t_0$
- control $x_1(t), \ldots, x_N(t)$ such that
  1. $\mathcal{G}$ stays connected (or $(\mathcal{G}, p)$ stays infinitesimally rigid) $\forall t > t_0$
  2. the mission of each robot is accomplished

### Maintenance

- eventual achievement
- periodical achievement
Using the **algebraic formulation** of connectivity and infinitesimal rigidity

**Connectivity maintenance**

- Assume $\lambda_2(t_0) > 0$
- For $t > t_0$
  - **Maintain** $\lambda_2(x_1(t), \ldots, x_N(t), z) > 0$
  - And accomplish the mission

**Infinitesimal rigidity maintenance**

- Assume $\varsigma_7(t_0) > 0$
- For $t > t_0$
  - **Maintain** $\varsigma_7(x_1(t), \ldots, x_N(t), z) > 0$
  - And accomplish the mission
Gradient-based Maintenance Methods

Assume robot $i$ can control $x_i^{(h)} = \frac{\text{d}^{h}}{\text{d}t^{h}} x_i$ for a certain $h \geq 1$

1. define **potential function** $V : (\mu_{\text{min}}, +\infty) \rightarrow \mathbb{R}^+$, that
   - grows unbounded as $\mu \rightarrow +\infty$  \( \mu_{\text{min}} > 0 \)
   - vanishes (with vanishing derivatives) as $\mu \geq \mu^0 > \mu_{\text{min}}$
   - is, at least, $C^1$, i.e., it exists $\frac{\text{d}V}{\text{d}\mu}$, \( \forall \mu > \mu_{\text{min}} \)

2. let each robot **command**

\[
\begin{align*}
x_i^{(h)} &= \left. \frac{\text{d}V}{\text{d}\mu} \right|_{\lambda_2(t)} \left. \frac{\partial \lambda_2}{\partial x_i} \right|_{(x_1, \ldots, x_N, z)} + u_i \quad \text{(for connectivity maintenance)} \\
x_i^{(h)} &= \left. \frac{\text{d}V}{\text{d}\mu} \right|_{\varsigma_7(t)} \left. \frac{\partial \varsigma_7}{\partial x_i} \right|_{(x_1, \ldots, x_N, z)} + u_i \quad \text{(for infinitesimal rigidity maintenance)}
\end{align*}
\]

where $u_i$ is a properly designed additional control input accounting for
   - accomplishment of mission
   - stability
Gradient Computation

connectivity maintenance

\[ \frac{dV}{d\mu} \bigg|_{\lambda_2(t)} \frac{\partial \lambda_2}{\partial x_i} \bigg|_{(x_1,...,x_N,z)} \]

infinitesimal rigidity maintenance

\[ \frac{dV}{d\mu} \bigg|_{\varsigma_7(t)} \frac{\partial \varsigma_7}{\partial x_i} \bigg|_{(x_1,...,x_N,z)} \]

Gradient computation is composed by two parts
Gradient Computation

First part: computation of \( \frac{dV}{d\mu} \rvert_{\lambda_2(t)} \) (or \( \frac{dV}{d\mu} \rvert_{\varsigma_7(t)} \))

requires that each robot knows:

- the function \( V \)
- \( \lambda_2(t) \) (or \( \varsigma_7(t) \))
Gradient Computation

Second part: Computation of $\frac{\partial \lambda_2}{\partial x_i} \bigg|_{(x_1,\ldots,x_N,z)}$ (or $\frac{\partial \varsigma_7}{\partial x_i} \bigg|_{(x_1,\ldots,x_N,z)}$)

requires in general

- the **analytic expression** of the gradient of $\lambda_2$ (or $\varsigma_7$) with respect to $x_i$
Gradient of $\lambda_2$ and $\varsigma_7$

Given a matrix $M$, any eigenvalue can be written as $\mu = u^T M u$, where

- $u$ is a normalized eigenvector associated to $\mu$ (i.e., $M u = \mu u$ and $u^T u = 1$)

**Connectivity**

$$\lambda_2 = u^T L u$$

differentiating, we obtain (Yang et al. 2010)

$$\frac{\partial \lambda_2}{\partial x_i} = \sum_{(j, h) \in E} \frac{\partial w_{jh}}{\partial x_i} (u_j - u_h)^2$$

**Infinitesimal rigidity**

$$\varsigma_7 = u^T S u$$

differentiating, we obtain (Zelazo et al. 2014)

$$\frac{\partial \varsigma_7}{\partial x_i} = \sum_{(j, h) \in E} \frac{\partial w_{jh}}{\partial x_i} s_{jh} + \frac{\partial s_{jh}}{\partial x_i} w_{jh}$$

$$s_{jh} = \left( (p_j^x - p_h^x) (u_j^x - u_h^x)^2 + (p_j^y - p_h^y) (u_j^y - u_h^y)^2 + (p_j^z - p_h^z) (u_j^z - u_h^z)^2 + 2(p_j^x - p_h^x)(p_j^y - p_h^y)(u_j^x - u_h^x)(u_j^y - u_h^y) + 2(p_j^x - p_h^x)(p_j^z - p_h^z)(u_j^x - u_h^x)(u_j^z - u_h^z) + 2(p_j^y - p_h^y)(p_j^z - p_h^z)(u_j^y - u_h^y)(u_j^z - u_h^z) \right)$$
Consider a network of robots performing a control law. The control law is decentralized if, for each robot $i$, the size of the:

- communication bandwidth
- computation time (per step)
- memory used (inputs, outputs, local variables)

depends only on $|\mathcal{N}_i|$ and not on $N$.

- A control law that is not decentralized is not scalable.

Example of decentralized control law: consensus

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i) \quad \forall i$$
The two control laws shown so far, i.e.,

**connectivity maintenance**

\[
\frac{dV}{d\mu} \bigg|_{\mu=\lambda_2} \sum_{(j,h) \in E} \frac{\partial w_{jh}}{\partial x_i} (u_j - u_h)^2
\]

**infinitesimal rigidity maintenance**

\[
\frac{dV}{d\mu} \bigg|_{\mu=\varsigma_7} \sum_{(j,h) \in E} \frac{\partial w_{jh}}{\partial x_i} s_{jh} + \frac{\partial s_{jh}}{\partial x_i} w_{jh}
\]

are **not decentralized** control law because

- each robot must know \( \lambda_2 \) (or \( \varsigma_7 \)) that depends on \( x_1(t), \ldots, x_N(t), z \)
- each robot must know \( w_{jh} \) and \( s_{jh} \), \( \forall (j, h) \in E \), and \( u_1, \ldots, u_N \) that also depend on \( x_1(t), \ldots, x_N(t), z \)

**Goal**: make the control law **decentralized**
Locality assumption for the connection map $w$

\[ \forall i \in \mathcal{V}, \forall (j, h) \in \mathcal{E} \quad \frac{\partial w_{jh}}{\partial x_i} = 0 \text{ if neither } j = i \text{ nor } h = i \]

Consequence for connectivity gradient

\[ \frac{\partial \lambda_2}{\partial x_i} = \sum_{(j, h) \in \mathcal{E}} \frac{\partial w_{jh}}{\partial x_i} (u_j - u_h)^2 = \sum_{j \in \mathcal{N}_i} \frac{\partial w_{ij}}{\partial x_i} (u_i - u_j)^2 \]

\[ \frac{\partial \lambda_2}{\partial x_i} = \sum_{j \in \mathcal{N}_i} f_\lambda \left( \frac{\partial w_{ij}}{\partial x_i}, w_{ij}, x_i, x_j, u_i, u_j \right) \]
Decentralized Gradient-based Methods

Locality assumption for the connection map $w$

$$\forall i \in \mathcal{V}, \forall (j, h) \in \mathcal{E} \quad \frac{\partial w_{jh}}{\partial x_i} = 0 \text{ if neither } j = i \text{ nor } h = i$$

Consequence for infinitesimal rigidity gradient

$$\frac{\partial \varsigma_7}{\partial x_i} = \sum_{(j, h) \in \mathcal{E}} \frac{\partial w_{jh}}{\partial x_i} s_{jh} + \frac{\partial s_{jh}}{\partial x_i} w_{jh} = \sum_{j \in \mathcal{N}_i} \frac{\partial w_{ij}}{\partial x_i} s_{ij} + \frac{\partial s_{ij}}{\partial x_i} w_{ij} =$$

$$\sum_{j \in \mathcal{N}_i} \frac{\partial w_{ij}}{\partial x_i} \left( (p_{ij}^x)^2 (u_i^x - u_j^x)^2 + (p_{ij}^y)^2 (u_i^y - u_j^y)^2 + (p_{ij}^z)^2 (u_i^z - u_j^z)^2 + 2p_{ij}^x p_{ij}^y (u_i^x - u_j^x)(u_i^y - u_j^y) + 2p_{ij}^x p_{ij}^z (u_i^x - u_j^x)(u_i^z - u_j^z) + 2p_{ij}^y p_{ij}^z (u_i^y - u_j^y)(u_i^z - u_j^z) \right)$$

$$+ \sum_{j \in \mathcal{N}_i} \begin{pmatrix} u_i^x - u_j^x \\ u_i^y - u_j^y \\ u_i^z - u_j^z \end{pmatrix} \begin{pmatrix} 2w_{ij} (p_{ij}^x (u_i^x - u_j^x) + p_{ij}^y (u_i^y - u_j^y) + p_{ij}^z (u_i^z - u_j^z)) \end{pmatrix}$$

$$\frac{\partial \varsigma_7}{\partial x_i} = \sum_{j \in \mathcal{N}_i} f_\varsigma \left( \frac{\partial w_{ij}}{\partial x_i}, w_{ij}, x_i, x_j, u_i, u_j \right)$$

where $p_{ij} = p_i - p_j$
Gradient-based Maintenance Methods

Locality assumption for the connection map $w$

$\forall i \in V, \forall (j, h) \in E \quad \frac{\partial w_{jh}}{\partial x_i} = 0$ if neither $j = i$ nor $h = i$

The two gradient-based control laws with locality assumption

- Connectivity maintenance
  
  $$V'(\lambda_2) \sum_{j \in N_i} f_{\lambda} \left( \frac{\partial w_{ij}}{\partial x_i}, w_{ij}, x_i, x_j, u_i, u_j \right)$$

- Infinitesimal rigidity maintenance
  
  $$V'(\varsigma_7) \sum_{j \in N_i} f_{\varsigma} \left( \frac{\partial w_{ij}}{\partial x_i}, w_{ij}, x_i, x_j, u_i, u_j \right)$$

become partially decentralized control law, each robot must know:

- $\lambda_2$ (or $\varsigma_7$) that depends on $x_1(t), \ldots, x_N(t), z$ (not decentralized)
- $x_i, w_{ij}, \frac{\partial w_{ij}}{\partial x_i}$, and $x_j, \forall j \in N_i$, and $z$, (decentralized)
- $u_i$ and $u_j, \forall j \in N_i$ that depend on $x_1(t), \ldots, x_N(t), z$ (not decentralized)

**Goal**: compute $\lambda_2$ (or $\varsigma_7$), $u_i$ and $u_j, \forall j \in N_i$ in a decentralized way
Computation of $\lambda_2$ and $\varsigma_7$

Continuous power iteration method (Yang et al. 2010; Zelazo et al. 2014)

An iterative algorithm to get an estimate $\hat{\mu}$ and $\hat{u}$ of the $l$-th eigenvalue $\mu$ and the associated eigenvector $u$ of a positive semidefinite matrix $M \in \mathbb{R}^n$.

Denote with $T \in \mathbb{R}^{n \times l-1}$ the image matrix of the first $l-1$ eigenvectors.

\[
\dot{\hat{u}} = -k_1 TT^T \hat{u} - k_2 M \hat{u} - k_3 \left( \frac{\hat{u}^T \hat{u}}{n} - 1 \right)
\]

- $-k_1 TT^T \hat{u}$: deflation, to remove the components spanned by the first $l-1$ eigenvectors.
- $-k_2 M \hat{u}$: direction update, to move towards $u$.
- $-k_3 \left( \frac{\hat{u}^T \hat{u}}{n} - 1 \right)$: renormalization to stay away from the null vector.

The eigenvalue is estimated as

\[
\hat{\mu} = \frac{k_3}{k_2} \left( 1 - \|\hat{u}\|^2 \right)
\]
Decentralized Computation of $\lambda_2$ and $\varsigma_7$

Decentralized power iteration method (Yang et al. 2010; Zelazo et al. 2014)

\[ \hat{u} = -k_1 TT^T \hat{u} - k_2 M \hat{u} - k_3 \left( \frac{\hat{u}^T \hat{u}}{n} - 1 \right) \]

connectivity maintenance

\[
M = L \\
T = 1
\]

infinitesimal rigidity maintenance

\[
M = S \\
T \in \mathbb{R}^{3N \times 6} \quad \text{def. in (Zelazo et al. 2014)}
\]

The only remaining global quantities

- $T^T \hat{u}$
- $\hat{u}^T \hat{u}$

can be estimated using the \textbf{proportional/integral-average consensus estimator} (PI-ACE) (Yang et al. 2010)
Limits of Gradient-based methods

Possible **limits** of the gradient-based methods

- the robot could be **unable to follow** the gradient because of, e.g., input saturation
- possibility of **local minima** (depending on the environment complexity)
Possible **limits** of the decentralized methods:

- need for **time-scale separation**: decentralized estimator dynamics must be faster than motion control dynamics

- the **gains** of the decentralized estimator must be carefully tuned depending on $N$

- decentralized power iteration does not work for eigenvalues with **multiplicity** $> 1$

- (decentralized) power iteration has a relatively **slow convergence**

Possible destabilization due to non-perfect estimation can be mitigated using **passivity theory** (Robuffo Giordano et al. 2013)
Handling Multiple Objectives in Maintenance Problems
Connectivity in a network of robots is typically associated to inter-robot communication and relative sensing. Quality of inter-robot sensing/communication is modeled by a sufficiently smooth non-negative scalar function:

\[ \gamma_{ij} = \gamma(x_i, x_j, z) \geq 0 \]

Measures the quality of the mutual information exchange:

- \( \gamma_{ij} = 0 \) if no exchange is possible and
- \( \gamma_{ij} > 0 \) otherwise
- the larger \( \gamma_{ij} \) the better the quality

Straightforward use:

\[ w_{ij} = \gamma_{ij} \]
In order to handle multiple objectives define

\[ w_{ij} = \alpha_{ij} \beta_{ij} \gamma_{ij} \]

where

- \( \alpha_{ij} \geq 0 \) encodes hard constraints
- \( \beta_{ij} \geq 0 \) encodes soft requirements
- \( \gamma_{ij} \geq 0 \) encodes the communication/sensing objectives (defined before)

this defines the

- generalized connectivity, and a
- generalized infinitesimal rigidity
Hard Constraints

Hard constraints: conditions $HD_1, HD_2, \ldots$ that must be true $\forall t \geq 0$

Maintenance methods automatically keep true a hard constraint: $HD_0 \equiv$ connectivity

Idea: define $\alpha_{ij}$ such that
- not $HD_h$ for some $h \Rightarrow$ not $HD_0$

How? Just define $\alpha_{ij}$ s.t.
- not $HD_h$ for some $h \Rightarrow \alpha_{ij} = 0, \forall j = 1, \ldots, N$

Why only $\alpha_{ij} = 0, \forall j = 1, \ldots, N$?
- it is enough for non-connectivity
  ($\alpha_{ij} = 0, \forall j = 1, \ldots, N$ implies robot $i$ becomes disconnected from the rest)
- is intrinsically decentralized

$\alpha_{ij}$ must be smooth enough to allow for gradient computation
- the more $\alpha_{ij} \to 0$ the closer to not $HD_h$
Soft Requirements

Soft requirements: should be **preferably** realized by the individual pair \((i, j)\)

Notes:
- gradient-based maintenance methods tend to maximize the **maintenance eigenvalues** (e.g., \(\lambda_2\) or \(\varsigma_7\))
- maintenance eigenvalues monotonically increase w.r.t. \(w_{ij} \ \forall (i,j) \in \mathcal{E}\)

**Idea:** define \(\beta_{ij}\) such that
- has a unique maximum when the soft constraints are realized
- monotonically decreases down to \(\beta_{ij} = 0\) otherwise

Non-perfect compliance with a soft requirement leads to
- corresponding decrease of maintenance eigenvalue
  \[\downarrow \beta_{ij} \Rightarrow \downarrow w_{ij} \Rightarrow \downarrow \lambda_2 \text{ (or } \downarrow \varsigma_7)\]

Complete violation of soft requirement
- leads to disconnected edge \((i, j)\), but
- does not (in general) result in a global loss of connectivity for the graph
Applications
Particular Choices of the Weights

Communication/sensing objectives $\rightarrow \gamma_{ij}(x_i, x_j, z)$

**Proximity** sensing model:
- $D > 0$ is a suitable sensing/communication **maximum range** (e.g., radio signal)
- robot $i$ and $j$ able to interact iff $\|x_i - x_j\| < D$

**Proximity-visibility** sensing model (e.g., onboard cameras):
- $S_{ij}$ **line-of-sight** segment joining $x_i$ and $x_j$
- robot $i$ and $j$ able to interact iff $\|x_i - x_j\| < D$, and $\text{dist} (S_{ij(x_i,x_j)}, \text{obst}(z)) > D_{\text{vis}}$
Particular Choices of the Weights

Hard constraints $\rightarrow \alpha_{ij}$

- e.g., inter-robot collision avoidance:
  $\|x_i - x_j\| > d_0$

Soft requirements $\rightarrow \beta_{ij}$

- e.g., formation control, e.g.,
  $\|x_i - x_j\| \sim d_{des}$
Multi-Target Exploration with Connectivity

Mission: concurrent exploration of a sequence of targets
While maintaining “generalized” connectivity, i.e., including

- proximity/visibility sensing model
- collision avoidance
- preferred inter-distance

Connectivity maintenance in case of, e.g., second order systems:

\[
\dot{x}_i = \frac{dV}{d\mu} \left| \lambda_2(t) \frac{\partial \lambda_2}{\partial x_i} \right|_{(x_1, \ldots, x_N, z)} + u_i
\]

\[
u_i = -B \dot{x}_i + f_{i}^{\text{expl}}
\]

- \(-B \dot{x}_i\) stabilizing damping
- \(f_{i}^{\text{expl}}\) multi-target exploration force (Nestmeyer et al. 2015, Under Review)

Multi-Target Exploration with Connectivity

empty space

completion time [s]

100 200 300
150 250 350
10 15 20 25 30 35

mean traveled distance [m]

120 140 160 180 200 220
10 15 20 25 30 35

number of robots

town

completion time [s]

100 200 300
150 250 350
10 15 20 25 30 35

mean traveled distance [m]

120 140 160 180 200 220
10 15 20 25 30 35

number of robots

office

completion time [s]

100 120 140 160 180
10 15 20 25 30 35

mean traveled distance [m]

10 12 14 16 18
10 15 20 25 30 35

number of robots

Antonio Franchi
Connectivity, Rigidity and Online Decentralized Maintenance Methods – http://homepages.laas.fr/afranchi/robotics/
Mission: **unilateral multi-user teleoperation** of some robots in the team
While maintaining “generalized” **infinitesimal rigidity**, i.e., including
- proximity/visibility sensing model
- collision avoidance
- preferred inter-distance

Infinitesimal rigidity maintenance in case of, e.g., **first order** systems:

\[
\dot{x}_i = \frac{dV}{d\mu} \left|_{S_7(t)} \right. \frac{\partial S_7}{\partial x_i} \bigg|_{(x_1, \ldots, x_N, z)} + u_i \\
\]

\[
u_i^h \quad \text{if connected to a human} \quad \text{otherwise}
\]

- \(v_i^h\) **desired velocity** commanded by a **human**

videos: http://homepages.laas.fr/afranchi/robotics/?q=node/134
Short summary

- Single **scalars** can define **fundamental global properties**
  - $\lambda_2$ Fiedler eigenvalue (Fiedler 1973)
  - $\varsigma_7$ rigidity eigenvalue (Zelazo et al. 2014)

- **Distributed** computation of the **gradient** is possible
  - + smooth
  - + online computation (fast)
  - - presence of local minima

Some open problems

- coinciding eigenvalues
- local minima (using decentralized global planning?)
Decentralized multi-target exploration with connectivity maintenance


Bearing rigidity (in $SE(3)$)


Questions?

Connectivity, Rigidity and Online Decentralized Maintenance Methods

Antonio Franchi
CNRS, LAAS, France, Europe

2015 IROS Workshop on ‘On-line decision-making in multi-robot coordination’
(DEMUR’15)
Hamburg, Germany
12th October, 2015
IEEE RAS Technical Committee on Multi-Robot Systems:

http://multirobotsystems.org/

- recently founded (Fall 2014)
- 260 members
- identifying and constantly tracking the common characteristics, problems, and achievements of multi-robot systems research in its several and diverse domains
  - robotics
  - automatic control
  - telecommunications
  - computer science / AI
  - optimization
  - ...

If you work/are interested on multi-robot/agent systems then **become a member!**

http://multirobotsystems.org/?q=user/register