Decentralized Estimation and Control Methods for Cooperative Robot Motion

Antonio Franchi

CNRS, LAAS, France, Europe

Seoul National University, Engineering
Seoul, South Korea
25th October, 2016

Table of Contents

Partial State of the Art

Partial list:

- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," Automatica, vol. 46, no. 2. pp. 390–396, Feb. 2010.
- G. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," IEEE Transactions on Robotics , 2012,
- L. Sabattini, C. Secchi, N. Chopra, and A Gasparri. Distributed Control of Multirobot Systems With Global Connectivity Maintenance. Robotics, IEEE Transactions on Robotics, 29(5):1326-1332, 2013.
- D. Carboni, R.K. Williams, A. Gasparri, G. Ulivi, and G.S. Sukhatme. Rigidity-Preserving Team Partitions in Multi-Agent Networks. IEEE Transactions on Cybernetics, pp 1-14, 2014.

Main Sources

If you want to know more about what follows:

- Robuffo Giordano, P., A. Franchi, C. Secchi, and H. H. Bülthoff (2013). *A Passivity-Based Decentralized Strategy for Generalized Connectivity* Maintenance".
  The International Journal of Robotics Research 32.3, pp. 299–323.

- Zelazo, D., A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano (2014). *Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems*.
  The International Journal of Robotics Research 34.1, pp. 105–128.

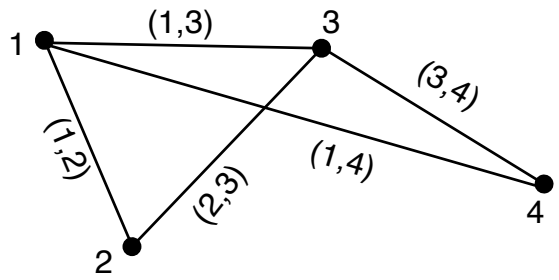
- Nestmeyer T., P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi (2016), *Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots*. Autonomous Robots, online first June 2016, doi:10.1007/s10514-016-9578-9

Graphs, Matrices, and Eigenvalues

Graph

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



A Graph models an Adjacency Structure

$[(i, j)] \in \mathcal{E} \Leftrightarrow$ vertexes i and j are **neighbors** or **adjacent**

- (i, j), $i < j$ representative element of the **equivalence class** $[(i, j)]$

$$\begin{aligned}[\mathcal{V} \times \mathcal{V}] &= \{(1, 2), (1, 3), \ldots, (1, N), \ldots, (N-1, N)\} \\ &= \{e_1, e_2, \ldots e_{N-1}, \ldots, e_{N(N-1)/2}\}\end{aligned}$$

- $[(i, i)] \notin \mathcal{E}, \forall i \in \mathcal{V}$ (**no self-loops**)
- $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$ set of **neighbors** of i

Incidence Matrix

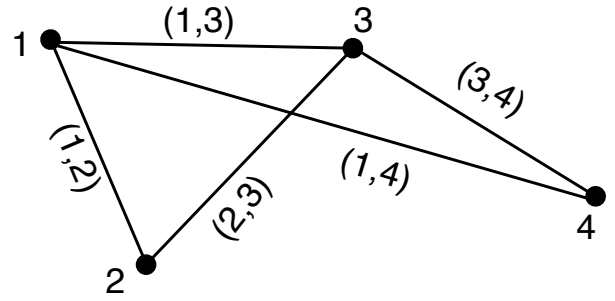
$E \in \mathbb{R}^{N \times N(N-1)/2}$ is the (full) **incidence matrix** of $\mathcal{G}$

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

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

Matricial representation of a graph

Example:



$$E = \begin{array}{c} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & -1 \end{pmatrix} \\ \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \end{array}$$

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:

- $\mathbf{x}_i \in \mathbb{R}^{N_\mathbf{x}}$ **i-th robot configuration**, $i \in 1 \dots N$
- $\mathbf{z} \in \mathbb{R}^{N_\mathbf{z}}$ **environment configuration**

Consider two maps

$$\begin{array}{rrl} \textbf{robot map} & \mathbf{v}: & \mathbb{R}^{N_\mathbf{x}} \ni \mathbf{x}_i \mapsto \mathbf{v}(\mathbf{x}_i) = \mathbf{v}_i \in \mathbb{R}^{N_\mathbf{v}} \\ \textbf{connection map} & \mathbf{w}: & \mathbb{R}^{N_\mathbf{x}} \times \mathbb{R}^{N_\mathbf{x}} \times \mathbb{R}^{N_\mathbf{z}} \ni (\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) \mapsto \mathbf{w}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) = \mathbf{w}_{ij} \in \mathbb{R}_{\geq 0} \end{array}$$

with the properties

- $\mathbf{w}_{ij} = \mathbf{w}_{ji}$ (symmetry)
- $\mathbf{w}_{ii} = 0$

example: what can those maps model?

Associated Graph and Framework

The connection map $\mathbf{w}$ defines an **associated graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

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

Both maps $\mathbf{v}$ and $\mathbf{w}$ define an **associated framework** $(\mathcal{G}, \mathbf{v})$ where

- $\mathcal{G}$ is the associated graph
- $\mathbf{v}_i$ is associated to each vertex $i \in \mathcal{V}$

Adjacency/Weight Matrix

$$A = \begin{pmatrix} 0 & \mathbf{w}_{12} & \dots & \mathbf{w}_{1N} \\ \mathbf{w}_{12} & 0 & \dots & \mathbf{w}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{1N} & \mathbf{w}_{2N} & \dots & 0 \end{pmatrix} \in \mathbb{R}^{N \times N}$$ is the **adjacency** (or **weight**) matrix of $\mathcal{G}$

Note that

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

Properties:

P.1 $A = A(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{z})$

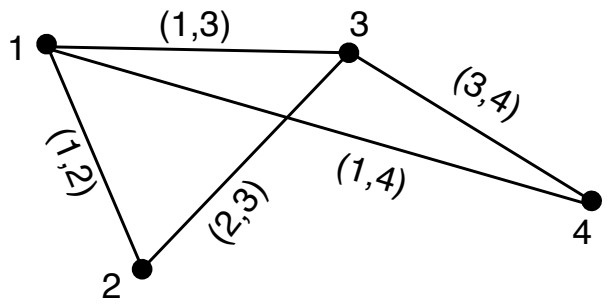
P.2 A is square

P.3 $A_{ij} = A_{ij}$ (symmetric)

P.4 $A_{ij} = A_{ij} \geq 0$ (nonnegative)

P.5 $A_{ii} = 0$

Example:



$$A = \begin{pmatrix} 0 & \mathbf{w}_{12} & \mathbf{w}_{13} & \mathbf{w}_{14} \\ \mathbf{w}_{12} & 0 & \mathbf{w}_{23} & 0 \\ \mathbf{w}_{13} & \mathbf{w}_{23} & 0 & \mathbf{w}_{34} \\ \mathbf{w}_{14} & 0 & \mathbf{w}_{34} & 0 \end{pmatrix}$$

Laplacian Matrix

$$L = \begin{pmatrix} \sum_{j=1}^{n} \mathbf{w}_{1j} & -\mathbf{w}_{12} & \cdots & -\mathbf{w}_{1N} \\ -\mathbf{w}_{12} & \sum_{j=1}^{n} \mathbf{w}_{j2} & \cdots & -\mathbf{w}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{w}_{1N} & -\mathbf{w}_{2N} & \cdots & \sum_{j=1}^{n} \mathbf{w}_{jN} \end{pmatrix} \in \mathbb{R}^{N \times N}$$ is the **Laplacian** matrix of $\mathcal{G}$

Note that

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

  where $\delta_i = \sum_{j=1}^{n} \mathbf{w}_{ij}$ (degree of vertex i)

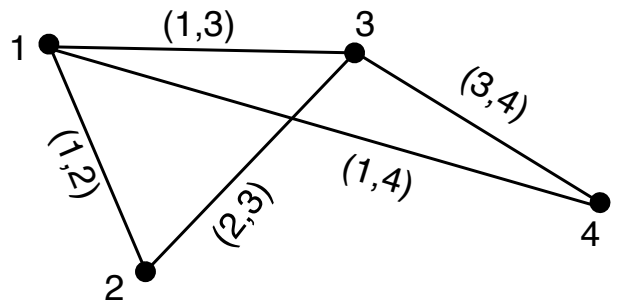
Properties:

P.1 $L = L(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z})$

P.2 L is square

P.3 $L_{ij} = L_{ij}$ (symmetric)

Example:



$$L = \begin{pmatrix} \mathbf{w}_{12}+\mathbf{w}_{13}+\mathbf{w}_{14} & -\mathbf{w}_{12} & -\mathbf{w}_{13} & -\mathbf{w}_{14} \\ -\mathbf{w}_{12} & \mathbf{w}_{12}+\mathbf{w}_{23} & -\mathbf{w}_{23} & 0 \\ -\mathbf{w}_{13} & -\mathbf{w}_{23} & \mathbf{w}_{13}+\mathbf{w}_{23}+\mathbf{w}_{34} & -\mathbf{w}_{34} \\ -\mathbf{w}_{14} & 0 & -\mathbf{w}_{34} & \mathbf{w}_{14}+\mathbf{w}_{34} \end{pmatrix}$$

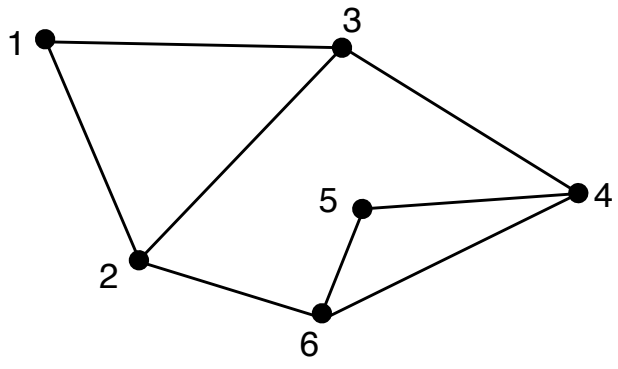
Connected Graph

Connectivity

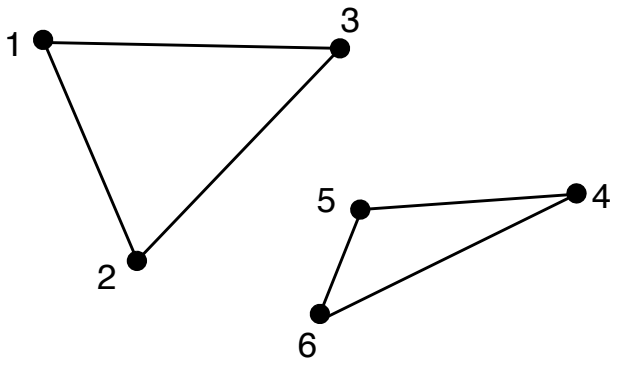
$\mathcal{G}$ is **connected** if there is a **path** between every pair of vertices, i.e.,

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

This is a **combinatorial definition** of connectivity



connected graph



disconnected graph

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

Spectrum of the Laplacian Matrix and Algebraic Connectivity

Additional properties of $L = \mathrm{diag}(\delta_i) - A$

- 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., $L\mathbf{1} = \mathbf{0}$, therefore

$$\lambda_1 = 0 \text{ and it is associated to the eigenvector } \mathbf{1} = \begin{pmatrix} 1 & 1 & \ldots & 1 \end{pmatrix}^T$$

(Fiedler 1973)

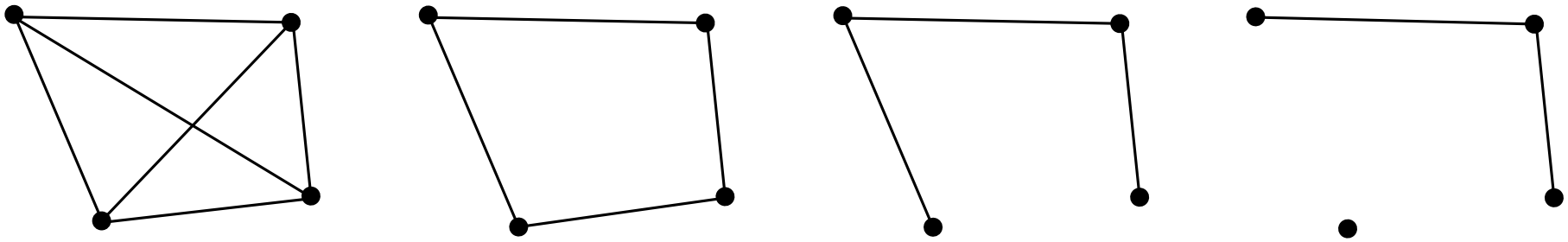
$\lambda_2 > 0$ if the graph $\mathcal{G}$ is **connected** and $\lambda_2 = 0$ otherwise

λ_2 provides an **algebraic definition** of connectivity

$\Rightarrow$ λ_2 is called *algebraic connectivity*, *connectivity eigenvalue*, or **Fiedler eigenvalue**

$\lambda_2 = \lambda_2(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z})$ is a **global** quantity

Example (if $\mathbf{w}_{ij} \in \{0, 1\}$ $\forall (i, j) \in \mathcal{V} \times \mathcal{V}$):



$\lambda_2 = 4$ $\lambda_2 = 2$ $\lambda_2 = 0.58$ $\lambda_2 = 0$

Framework of Positions

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

- if $d = 2$, $\mathbf{v}_i = \mathbf{p}_i = \begin{pmatrix} p_i^x \\ p_i^y \end{pmatrix}$, **2D position** of robot i
- if $d = 3$, $\mathbf{v}_i = \mathbf{p}_i = \begin{pmatrix} p_i^x \\ p_i^y \\ p_i^z \end{pmatrix}$, **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

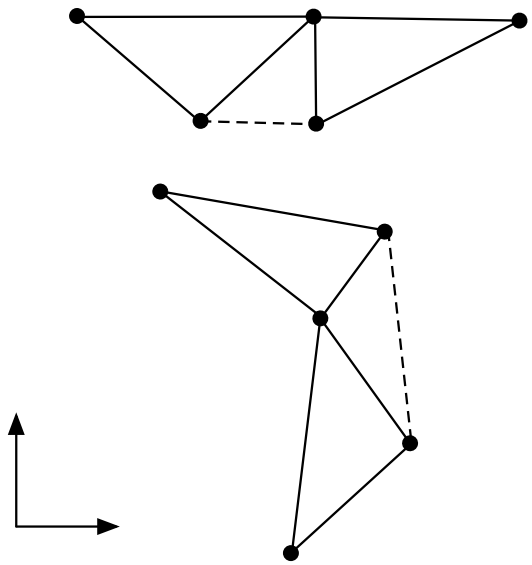
Equivalent and Congruent Frameworks

Consider two frameworks $(\mathcal{G}, \mathbf{p}')$ and $(\mathcal{G}, \mathbf{p}'')$

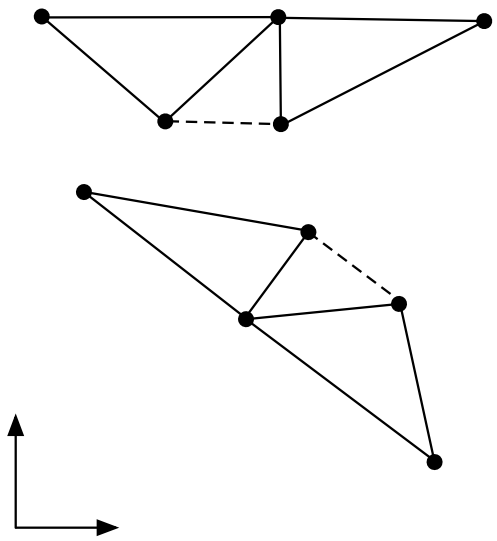
- same graph $\mathcal{G}$
- different positions $\mathbf{p}'$ and $\mathbf{p}''$

Frameworks $(\mathcal{G}, \mathbf{p}')$ and $(\mathcal{G}, \mathbf{p}'')$ are

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



equivalent frameworks



congruent frameworks

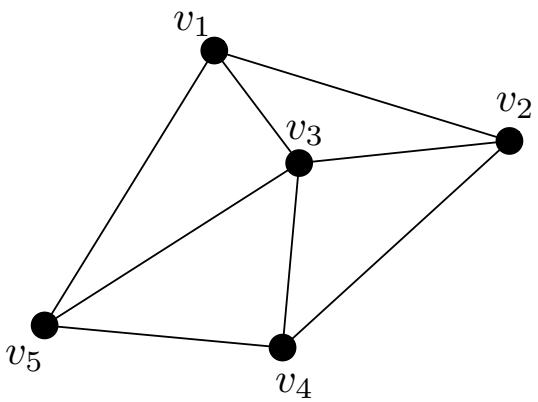
Global Rigidity

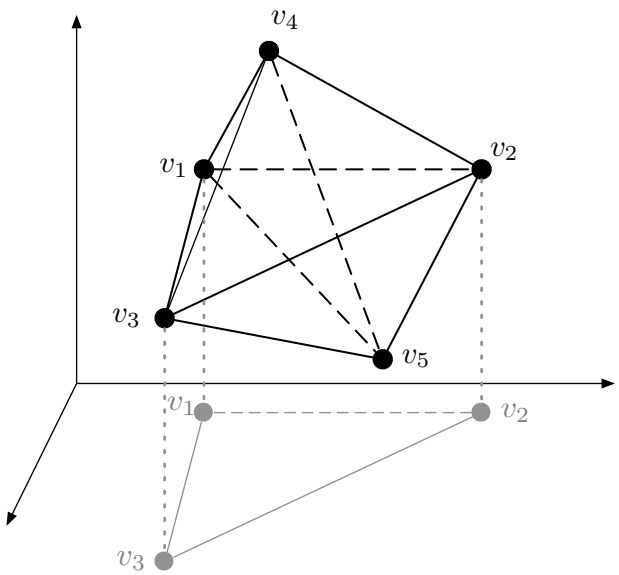
The framework $(\mathcal{G}, \mathbf{p}')$ is **globally rigid** if every other framework $(\mathcal{G}, \mathbf{p}'')$ which

- is equivalent to $(\mathcal{G}, \mathbf{p}'')$

is also congruent to $(\mathcal{G}, \mathbf{p}')$

This is, again, a **combinatorial definition**





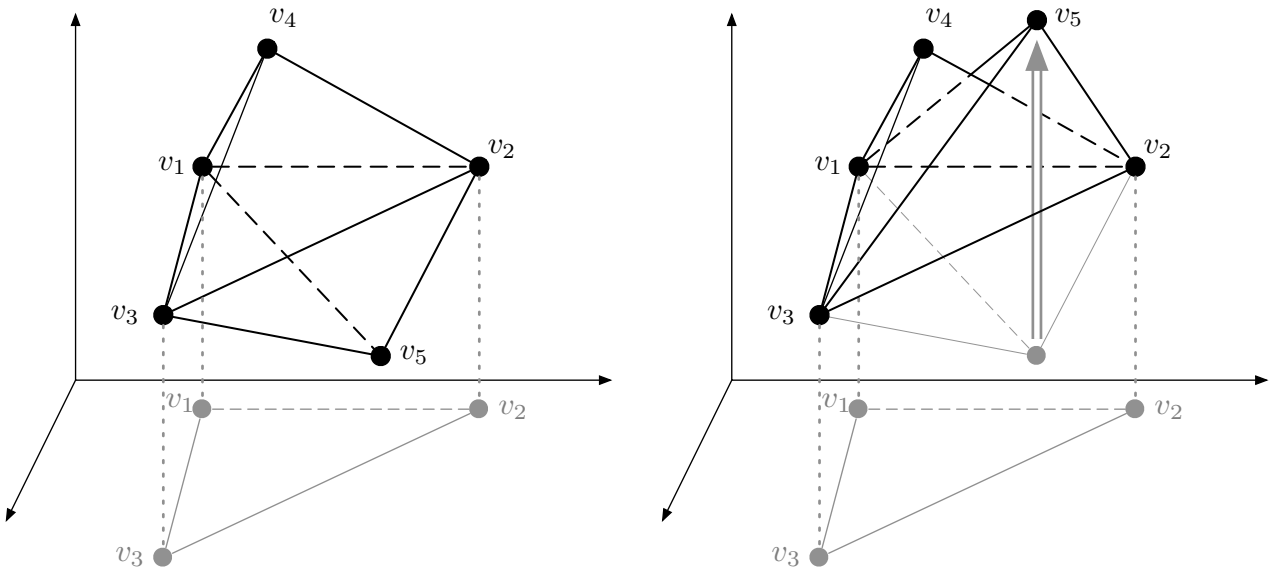
Rigidity

The framework $(\mathcal{G}, \mathbf{p}')$ is **rigid** if $\exists \epsilon > 0$ such that every other framework $(\mathcal{G}, \mathbf{p}'')$ which

- is equivalent to $(\mathcal{G}, \mathbf{p}'')$ and
- satisfies $\|\mathbf{p}'_i - \mathbf{p}''_i\| < \epsilon$ for all $i \in \mathcal{V}$,

is congruent to $(\mathcal{G}, \mathbf{p}')$

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

Example of use of Rigidity

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

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

6-DOF **Stewart platform** parallel robot

Credits: Robert L. Williams II

Infinitesimal Rigidity

Let's give a definition of rigidity that is differential ($\Leftrightarrow$ involves **infinitesimal motions**)

Consider a trajectory $\mathbf{p}(t)$ with $t \geq t_0$ and impose **equivalence** along the trajectory:

$$\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|^2 = \|\mathbf{p}_i(t_0) - \mathbf{p}_j(t_0)\|^2 = \text{const} \quad \text{for all} \quad (i,j) \in \mathcal{E}, \ \ \forall t \geq t_0$$

Differentiating with respect to time the constraint above:

$$\left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \left(\dot{\mathbf{p}}_i(t) - \dot{\mathbf{p}}_j(t)\right) = 0 \quad \text{for all} \quad (i,j) \in \mathcal{E}, \ \ \forall t \geq t_0 \tag{1}$$

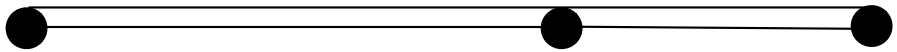
Trivial Motion

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

Infinitesimal Rigidity

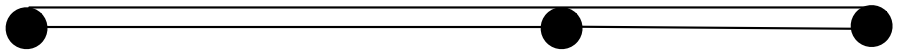
The framework $(\mathcal{G}, \mathbf{p}(t_0))$ is **infinitesimally rigid** if every possible motion that satisfies (1) is **trivial**

question: is this framework rigid in $\mathbb{R}^2$? is it infinitesimally rigid?



Rigidity vs Infinitesimal Rigidity

question: is this framework rigid in $\mathbb{R}^2$? is it infinitesimally rigid?



- infinitesimal rigidity $\Rightarrow$ rigidity
- rigidity $\not\Rightarrow$ infinitesimal rigidity

Matricial Representation of Infinitesimal Rigidity

Let us write the infinitesimal rigidity constraint in a matricial form

$$\left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \left(\dot{\mathbf{p}}_i(t) - \dot{\mathbf{p}}_j(t)\right) = 0 \quad \text{for all} \quad (i,j) \in \mathcal{E}, \ \ \forall t \geq t_0$$

$$\Updownarrow$$

$$\mathbf{w_{ij}} \left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \left(\dot{\mathbf{p}}_i(t) - \dot{\mathbf{p}}_j(t)\right) = 0 \quad \text{for all} \quad e_k = (i,j) \in [\mathcal{V} \times \mathcal{V}], \ \ \forall t \geq t_0$$

Matricial Representation of Infinitesimal Rigidity

$$
\begin{aligned}
0 &= \mathbf{w}_{ij} \left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \left(\dot{\mathbf{p}}_i(t) - \dot{\mathbf{p}}_j(t)\right) = \\
&= \mathbf{w}_{ij} \left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \dot{\mathbf{p}}_i(t) - \left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T \dot{\mathbf{p}}_j(t) = \\
&= \mathbf{w}_{ij} \underbrace{\begin{pmatrix} -\mathbf{0}^T - & \underbrace{\left(\mathbf{p}_i(t) - \mathbf{p}_j(t)\right)^T}_{\text{vertex } i} & -\mathbf{0}^T - & \underbrace{\left(\mathbf{p}_j(t) - \mathbf{p}_i(t)\right)^T}_{\text{vertex } j} & -\mathbf{0}^T - \end{pmatrix}}_{K_{ij} \in \mathbb{R}^{1 \times 3N}} \begin{pmatrix} \dot{\mathbf{p}}_1 \\ \vdots \\ \dot{\mathbf{p}}_N \end{pmatrix}
\end{aligned}
$$

where $\mathbf{0} = \begin{pmatrix} 0 & 0 & \dots & 0 \end{pmatrix}^T$

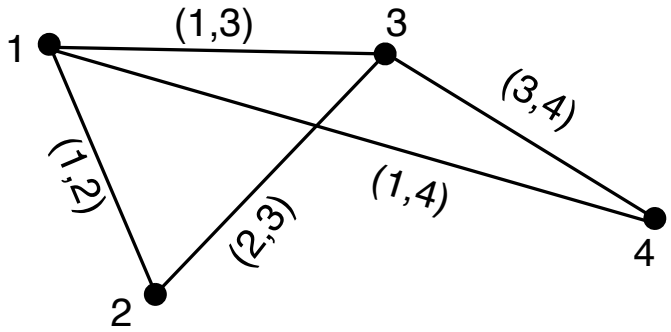
stacking the previous constraints for every $(i,j) \in \{e_1,\ e_2 \dots e_{N-1} \dots\dots, e_{N(N-1)/2}\}$:

$$\underbrace{\begin{pmatrix} \mathbf{w}_{12} & & \\ & \ddots & \\ & & \mathbf{w}_{N(N-1)} \end{pmatrix}}_{W(\mathbf{w}) \in \mathbb{R}^{\frac{N(N-1)}{2} \times \frac{N(N-1)}{2}}} \underbrace{\begin{pmatrix} K_{12} \\ \vdots \\ K_{N(N-1)} \end{pmatrix}}_{K(\mathbf{p})} \underbrace{\begin{pmatrix} \dot{\mathbf{p}}_1 \\ \vdots \\ \dot{\mathbf{p}}_N \end{pmatrix}}_{\dot{\mathbf{p}} \in \mathbb{R}^{3N}} = \underbrace{W(\mathbf{w})K(\mathbf{p})}_{R(\mathbf{w},\mathbf{p}) \in \mathbb{R}^{\frac{N(N-1)}{2} \times 3N}} \quad \dot{\mathbf{p}} = R(\mathbf{w},\mathbf{p})\dot{\mathbf{p}} = \mathbf{0}$$

Rigidity Matrix

$R(\mathbf{w},\mathbf{p})$ is the (weighted) **rigidity matrix**

Example of Rigidity Matrix



$d = 2\ (\mathbb{R}^2)$
$N = 4$
$N(N-1)/2 = 6$

$$R(\mathbf{w}, \mathbf{p}) =$$

$$\begin{pmatrix}
\mathbf{w}_{12}(p_1^x - p_2^x) & \mathbf{w}_{12}(p_1^y - p_2^y) & \mathbf{w}_{12}(p_2^x - p_1^x) & \mathbf{w}_{12}(p_2^y - p_1^y) & 0 & 0 & 0 & 0 \\
\mathbf{w}_{13}(p_1^x - p_3^x) & \mathbf{w}_{13}(p_1^y - p_3^y) & 0 & 0 & \mathbf{w}_{13}(p_3^x - p_1^x) & \mathbf{w}_{13}(p_3^y - p_1^y) & 0 & 0 \\
\mathbf{w}_{14}(p_1^x - p_4^x) & \mathbf{w}_{14}(p_1^y - p_4^y) & 0 & 0 & 0 & 0 & \mathbf{w}_{14}(p_4^x - p_1^x) & \mathbf{w}_{14}(p_4^y - p_1^y) \\
0 & 0 & \mathbf{w}_{23}(p_2^x - p_3^x) & \mathbf{w}_{23}(p_2^y - p_3^y) & \mathbf{w}_{23}(p_3^x - p_2^x) & \mathbf{w}_{23}(p_3^y - p_2^y) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{w}_{34}(p_3^x - p_4^x) & \mathbf{w}_{34}(p_3^y - p_4^y) & \mathbf{w}_{34}(p_4^x - p_3^x) & \mathbf{w}_{34}(p_4^y - p_3^y)
\end{pmatrix}$$

Rigidity vs Infinitesimal Rigidity

- rigidity is defined **combinatorially** ("...s.t. every other framework...")
- infinitesimal rigidity implies rigidity
- converse not true (degenerate cases) but...
- infinitesimal rigidity can be defined **algebraically**, in fact...

Rank of Rigidity Matrix

- **collective roto-translations** in $\mathbb{R}^3$ keep constant all the distances, by definition, i.e., if $\dot{\mathbf{p}}$ is trivial then $R(\mathbf{w}, \mathbf{p})\dot{\mathbf{p}} = 0$
- $\Rightarrow$ Dim $(\ker[R(\mathbf{w}, \mathbf{p})]) \geq 6$ 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 $R(\mathbf{w}, \mathbf{p})\dot{\mathbf{p}} = 0$ then $\dot{\mathbf{p}}$ is trivial
- infinitesimally rigidity $\Rightarrow$ Dim $(\ker[R(\mathbf{w}, \mathbf{p})]) = 6$

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

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

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

Symmetric Rigidity Matrix

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

(Zelazo et al. 2014)

Properties:

P.1 $S = S(\mathbf{w}, \mathbf{p}) = S(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z})$

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

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

P.4 $\mathrm{Dim}\left(\ker[S(\mathbf{w}, \mathbf{p})]\right) \geq 6$

(Zelazo et al. 2014)

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

Additional properties of $S = R^T R$

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

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

- $\mathrm{Dim}\left(\ker[S(\mathbf{w}, \mathbf{p})]\right) \geq 6$, 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

ς_7 provides an **algebraic definition** of infinitesimal rigidity

$\Rightarrow$ ς_7 is called the **rigidity eigenvalue** (Zelazo et al. 2014)

$\varsigma_7 = \varsigma_7(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z})$ is a **global** quantity

Connectivity vs Infinitesimal Rigidity

Similarities between Connectivity and Infinitesimal Rigidity

Connectivity

$\exists$ a path between any pair of vertexes

- depends on $\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}$ (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 $\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}$ (global property)
- symmetric rigidity matrix $S \in \mathbb{R}^{3N \times 3N}$
- $\Leftrightarrow$ rigidity eigenvalue $\varsigma_7 > 0$

Rigidity implies Connectivity

(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

Connectivity

- applicable to any graph
- depends only on $\mathbf{w}$
- $\not\Rightarrow$ infinitesimal rigidity

Infinitesimal rigidity

- applicable only to frameworks (graphs + positions)
- depends both on $\mathbf{w}$ and $\mathbf{v} = \mathbf{p}$
- $\Rightarrow$ connectivity

Infinitesimal rigidity is a **stronger property**
and
applies to a **more particular** structure (framework)

Maintenance Problems and Methods

Maintenance Problems

Assume each robot $i = 1, \ldots, N$

- can **control** $\mathbf{x}_i(t)$, $\forall t \geq t_0$ (with $\mathbf{x}_i(t)$ smooth enough)
- has some objectives (**mission**)

Maintenance problem(s)

- assume $\mathcal{G}$ is connected (or $(\mathcal{G}, \mathbf{p})$ is infinitesimally rigid) for $t = t_0$
- control $\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t)$ such that
  1. $\mathcal{G}$ **stays connected** (or $(\mathcal{G}, \mathbf{p})$ **stays infinitesimally rigid**) $\forall t > t_0$
  2. the mission of each robot is accomplished

Maintenance $\neq$

- 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** $\lambda_2(\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}) > 0$
  - and accomplish the mission

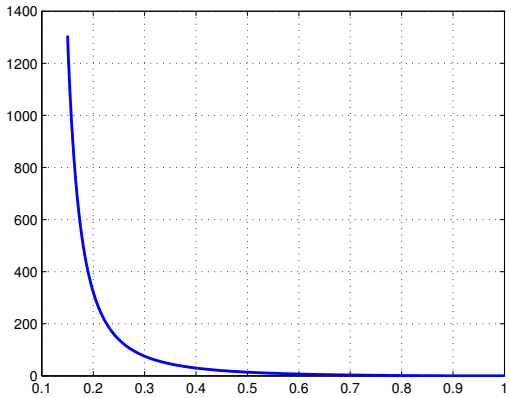
Infinitesimal rigidity maintenance

- assume $\varsigma_7(t_0) > 0$
- for $t > t_0$
  - **maintain** $\varsigma_7(\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}) > 0$
  - and accomplish the mission

Gradient-based Maintenance Methods

Assume robot i can control $\mathbf{x}_i^{(h)} = \frac{\mathrm{d}^h}{\mathrm{d}t^h}\mathbf{x}_i$ for a certain $h \geq 1$

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



2. let each robot **command**

$$\mathbf{x}_i^{(h)} = \left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\lambda_2(t)} \left.\frac{\partial \lambda_2}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})} + u_i$$

(for **connectivity maintenance**)

$$\mathbf{x}_i^{(h)} = \left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\varsigma_7(t)} \left.\frac{\partial \varsigma_7}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})} + u_i$$

(for **infinitesimal rigidity maintenance**)

where u_i is a properly designed additional control input accounting for

- **accomplishment of mission**
- **stability**

Gradient Computation

connectivity maintenance

$$\left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\lambda_2(t)} \left.\frac{\partial \lambda_2}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})}$$

infinitesimal rigidity maintenance

$$\left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\varsigma_7(t)} \left.\frac{\partial \varsigma_7}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})}$$

Gradient computation is composed by **two parts**

Gradient Computation

First part: computation of $\left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\lambda_2(t)}$ (or $\left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\varsigma_7(t)}$)

requires that **each robot knows**:

- the function V
- $\lambda_2(t)$ (or $\varsigma_7(t)$)

Second part: Computation of $\left.\frac{\partial \lambda_2}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})}$ (or $\left.\frac{\partial \varsigma_7}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})}$)

requires in general

- the **analytic expression** of the **gradient of** λ_2 (or ς_7) with respect to $\mathbf{x}_i$

Gradient of λ_2 and ς_7

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

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

Connectivity

$$\lambda_2 = \mathbf{u}^T L \mathbf{u}$$

differentiating, we obtain (Yang et al. 2010)

$$\frac{\partial \lambda_2}{\partial \mathbf{x}_i} = \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} (\mathbf{u}_j - \mathbf{u}_h)^2$$

Infinitesimal rigidity

$$\varsigma_7 = \mathbf{u}^T S \mathbf{u}$$

differentiating, we obtain (Zelazo et al. 2014)

$$\frac{\partial \varsigma_7}{\partial \mathbf{x}_i} = \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} \mathbf{s}_{jh} + \frac{\partial \mathbf{s}_{jh}}{\partial \mathbf{x}_i} \mathbf{w}_{jh}$$

$$\begin{aligned}
\mathbf{s}_{jh} = \Big(& (p_j^x - p_h^x)^2 (\mathbf{u}_j^x - \mathbf{u}_h^x)^2 + \\
& (p_j^y - p_h^y)^2 (\mathbf{u}_j^y - \mathbf{u}_h^y)^2 + \\
& (p_j^z - p_h^z)^2 (\mathbf{u}_j^z - \mathbf{u}_h^z)^2 + \\
& 2(p_j^x - p_h^x)(p_j^y - p_h^y)(\mathbf{u}_j^x - \mathbf{u}_h^x)(\mathbf{u}_j^y - \mathbf{u}_h^y) + \\
& 2(p_j^x - p_h^x)(p_j^z - p_h^z)(\mathbf{u}_j^x - \mathbf{u}_h^x)(\mathbf{u}_j^z - \mathbf{u}_h^z) + \\
& 2\,(p_j^y - p_h^y)(p_j^z - p_h^z)(\mathbf{u}_j^y - \mathbf{u}_h^y)(\mathbf{u}_j^z - \mathbf{u}_h^z)\Big)
\end{aligned}$$

Decentralized Control Law

Decentralized control law

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**

$$\dot{\mathbf{x}}_i = \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j - \mathbf{x}_i) \quad \forall i$$

The two control laws shown so far, i.e.,

connectivity maintenance

$$\frac{\mathrm{d}V}{\mathrm{d}\mu}\bigg|_{\mu=\lambda_2} \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i}(\mathbf{u}_j - \mathbf{u}_h)^2$$

infinitesimal rigidity maintenance

$$\frac{\mathrm{d}V}{\mathrm{d}\mu}\bigg|_{\mu=\varsigma_7} \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i}\mathbf{s}_{jh} + \frac{\partial \mathbf{s}_{jh}}{\partial \mathbf{x}_i}\mathbf{w}_{jh}$$

are they decentralized control law?

Gradient-based Maintenance Methods

The two control laws shown so far, i.e.,

connectivity maintenance

$$\frac{\mathrm{d}V}{\mathrm{d}\mu}\bigg|_{\mu=\lambda_2} \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i}(\mathbf{u}_j - \mathbf{u}_h)^2$$

infinitesimal rigidity maintenance

$$\frac{\mathrm{d}V}{\mathrm{d}\mu}\bigg|_{\mu=\varsigma_7} \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i}\mathbf{s}_{jh} + \frac{\partial \mathbf{s}_{jh}}{\partial \mathbf{x}_i}\mathbf{w}_{jh}$$

They are **not decentralized** control law because

- each robot must know λ_2 (or ς_7) that depends on $\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}$
- each robot must know $\mathbf{w}_{jh}$ and $\mathbf{s}_{jh}$, $\forall (j,h) \in \mathcal{E}$, and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ that also depend on $\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}$

Goal: make the control law **decentralized**

Decentralized Gradient-based Methods

Locality assumption for the connection map w

$$\forall i \in \mathcal{V},\ \forall (j,h) \in \mathcal{E} \quad \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} = 0 \text{ if neither } j = i \text{ nor } h = i$$

Consequence for connectivity gradient

$$\frac{\partial \lambda_2}{\partial \mathbf{x}_i} = \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} (\mathbf{u}_j - \mathbf{u}_h)^2 = \sum_{j\in\mathcal{N}_i} \frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i} (\mathbf{u}_i - \mathbf{u}_j)^2$$

$$\frac{\partial \lambda_2}{\partial \mathbf{x}_i} = \sum_{j\in\mathcal{N}_i} \mathbf{f}_\lambda \left(\frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}, \mathbf{w}_{ij}, \mathbf{x}_i, \mathbf{x}_j, \mathbf{u}_i, \mathbf{u}_j \right)$$

Locality assumption for the connection map w

$$\forall i \in \mathcal{V},\ \forall (j,h) \in \mathcal{E} \quad \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} = 0 \text{ if neither } j = i \text{ nor } h = i$$

Consequence for infinitesimal rigidity gradient

$$\frac{\partial \varsigma_7}{\partial \mathbf{x}_i} = \sum_{(j,h)\in\mathcal{E}} \frac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i}\mathbf{s}_{jh} + \frac{\partial \mathbf{s}_{jh}}{\partial \mathbf{x}_i}\mathbf{w}_{jh} = \sum_{j\in\mathcal{N}_i} \frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}\mathbf{s}_{ij} + \frac{\partial \mathbf{s}_{ij}}{\partial \mathbf{x}_i}\mathbf{w}_{ij} =$$

$$\sum_{j\in\mathcal{N}_i} \frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}\Big((p_{ij}^x)^2(\mathbf{u}_i^x - \mathbf{u}_j^x)^2 + (p_{ij}^y)^2(\mathbf{u}_i^y - \mathbf{u}_j^y)^2 + (p_{ij}^z)^2(\mathbf{u}_i^z - \mathbf{u}_j^z)^2 +$$

$$2p_{ij}^x p_{ij}^y(\mathbf{u}_i^x - \mathbf{u}_j^x)(\mathbf{u}_i^y - \mathbf{u}_j^y) + 2p_{ij}^x p_{ij}^z(\mathbf{u}_i^x - \mathbf{u}_j^x)(\mathbf{u}_i^z - \mathbf{u}_j^z) + 2p_{ij}^y p_{ij}^z(\mathbf{u}_i^y - \mathbf{u}_j^y)(\mathbf{u}_i^z - \mathbf{u}_j^z)\Big)$$

$$+ \sum_{j\in\mathcal{N}_i} \begin{pmatrix} \mathbf{u}_i^x - \mathbf{u}_j^x \\ \mathbf{u}_i^y - \mathbf{u}_j^y \\ \mathbf{u}_i^z - \mathbf{u}_j^z \end{pmatrix} 2\mathbf{w}_{ij}\left(p_{ij}^x(\mathbf{u}_i^x - \mathbf{u}_j^x) + p_{ij}^y(\mathbf{u}_i^y - \mathbf{u}_j^y) + p_{ij}^z(\mathbf{u}_i^z - \mathbf{u}_j^z)\right)$$

$$\frac{\partial \varsigma_7}{\partial \mathbf{x}_i} = \sum_{j\in\mathcal{N}_i} \mathbf{f}_\varsigma\left(\frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}, \mathbf{w}_{ij}, \mathbf{x}_i, \mathbf{x}_j, \mathbf{u}_i, \mathbf{u}_j\right)$$

where $p_{ij} = p_i - p_j$

Locality assumption for the connection map w

$\forall i \in \mathcal{V}, \forall (j,h) \in \mathcal{E} \quad \dfrac{\partial \mathbf{w}_{jh}}{\partial \mathbf{x}_i} = 0$ if neither $j = i$ nor $h = i$

The two gradient-based control laws with **locality assumption**

connectivity maintenance

$$V'(\lambda_2) \sum_{j \in \mathcal{N}_i} \mathbf{f}_\lambda \left(\frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}, \mathbf{w}_{ij}, \mathbf{x}_i, \mathbf{x}_j, \mathbf{u}_i, \mathbf{u}_j \right)$$

infinitesimal rigidity maintenance

$$V'(\varsigma_7) \sum_{j \in \mathcal{N}_i} \mathbf{f}_\varsigma \left(\frac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}, \mathbf{w}_{ij}, \mathbf{x}_i, \mathbf{x}_j, \mathbf{u}_i, \mathbf{u}_j \right)$$

become **partially decentralized** control law, each robot must know:

- λ_2 (or ς_7) that depends on $\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}$ (**not decentralized**)
- $\mathbf{x}_i$, $\mathbf{w}_{ij}$, $\dfrac{\partial \mathbf{w}_{ij}}{\partial \mathbf{x}_i}$, and $\mathbf{x}_j$, $\forall j \in \mathcal{N}_i$, and $\mathbf{z}$, (**decentralized**)
- $\mathbf{u}_i$ and $\mathbf{u}_j$, $\forall j \in \mathcal{N}_i$ that depend on $\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t), \mathbf{z}$ (**not decentralized**)

Goal: compute λ_2 (or ς_7), $\mathbf{u}_i$ and $\mathbf{u}_j$ $\forall j \in \mathcal{N}_i$ in a decentralized way

Computation of λ_2 and ς_7

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

An **iterative algorithm** to get an estimate $\hat{\mu}$ and $\hat{\mathbf{u}}$ of the the l-th eigenvalue μ and the associated eigenvector $\mathbf{u}$ 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

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

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

The **eigenvalue** is estimated as

$$\hat{\mu} = \frac{k_3}{k_2} \left(1 - \|\hat{\mathbf{u}}\|^2 \right)$$

Decentralized Computation of λ_2 and ς_7

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

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

connectivity maintenance	infinitesimal rigidity maintenance
$M = L$	$M = S$
$T = \mathbf{1}$	$T \in \mathbb{R}^{3N \times 6}$ def. in (Zelazo et al. 2014)

The only remaining global quantities

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

can be estimated using the
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)

Limits of Decentralized Methods

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

Communication and Sensing Objectives

Connectivity in a network of robots is typically associated to

Inter-robot

- **communication**
- **relative sensing**

Quality of inter-robot sensing/communication modeled by a sufficiently **smooth non-negative scalar** function

$$\gamma_{ij} = \gamma(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}) \geq 0$$

Measures the **quality** of the **mutual information exchange**

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

Straightforward use:

$$\mathbf{w}_{ij} = \gamma_{ij}$$

Handling Multiple Objectives: Generalized Connectivity/Rigidity

In order to handle multiple objectives define

$$\mathbf{w}_{ij} = \alpha_{ij}\beta_{ij}\gamma_{ij}$$

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 $\mathrm{HD}_1, \mathrm{HD}_2, \ldots$ that must be **true** $\forall t \geq 0$

Maintenance methods **automatically** keep true a hard constraint: $\mathbf{HD}_0 \equiv$ **connectivity**

Idea: define α_{ij} such that

- $\mathrm{not}\ \mathrm{HD}_h$ for some $h \Rightarrow \mathrm{not}\ \mathrm{HD}_0$

How? Just define α_{ij} s.t.

- $\mathrm{not}\ \mathrm{HD}_h$ 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

α_{ij} must be smooth enough to allow for gradient computation

- the more $\alpha_{ij} \to 0$ the closer to $\mathrm{not}\ \mathrm{HD}_h$

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., λ_2 or ς_7)
- maintenance eigenvalues monotonically increase w.r.t. $\mathbf{w}_{ij}$ $\forall (i,j) \in \mathcal{E}$

Idea: define β_{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
  $\downarrow \beta_{ij} \Rightarrow \downarrow \mathbf{w}_{ij} \Rightarrow \downarrow \lambda_2$ (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

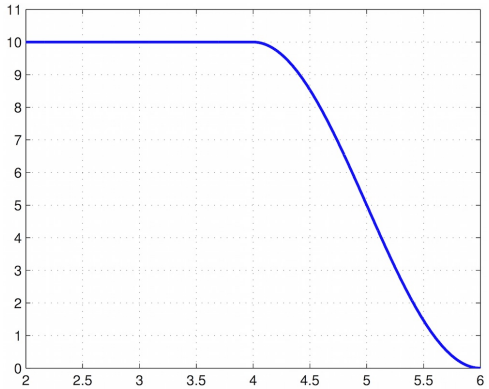
Applications

Particular Choices of the Weights

Communication/sensing objectives $\rightarrow \gamma_{ij}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z})$

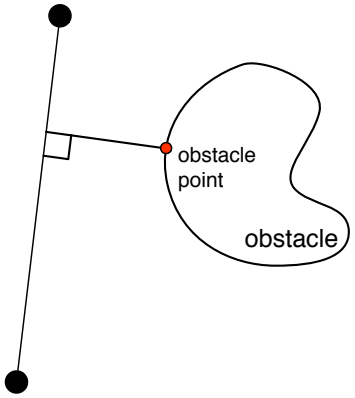
Proximity sensing model:

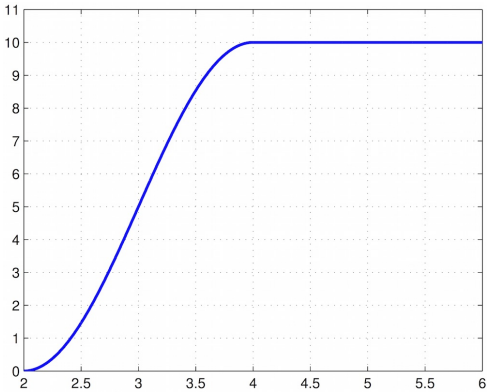
- $D > 0$ is a suitable sensing/communication **maximum range** (e.g, radio signal)
- robot i and j able to interact iff $\|\mathbf{x}_i - \mathbf{x}_j\| < D$,



Proximity-visibility sensing model (e.g., onboard cameras):

- S_{ij} **line-of-sight** segment joining $\mathbf{x}_i$ and $\mathbf{x}_j$
- robot i and j able to interact iff $\|\mathbf{x}_i - \mathbf{x}_j\| < D$, and $\mathrm{dist}\left(S_{ij(\mathbf{x}_i,\mathbf{x}_j)}, \mathrm{obst}(\mathbf{z})\right) > D_{\mathrm{vis}}$

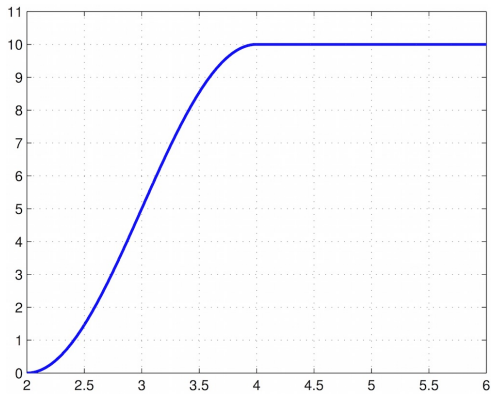




Particular Choices of the Weights

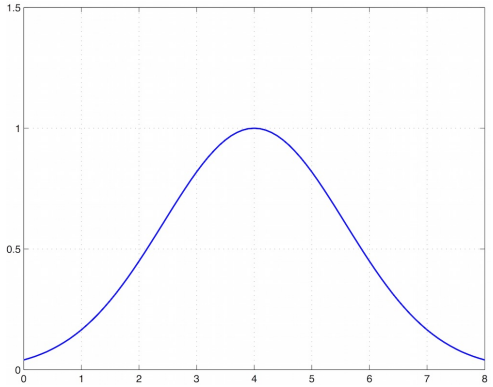
Hard constraints $\rightarrow \alpha_{ij}$

e.g., **inter-robot collision avoidance**:
$\|\mathbf{x}_i - \mathbf{x}_j\| > d_0$



Soft requirements $\rightarrow \beta_{ij}$

e.g., **formation control**, e.g.,
$\|\mathbf{x}_i - \mathbf{x}_j\| \simeq d_{\mathrm{des}}$



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:

$$\ddot{\mathbf{x}}_i = \left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\lambda_2(t)} \left.\frac{\partial \lambda_2}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})} + u_i$$

$$u_i = -B\dot{\mathbf{x}}_i + f_i^{\mathrm{expl}}$$

- $-B\dot{\mathbf{x}}_i$ stabilizing damping
- f_i^{expl} multi-target exploration force (Nestmeyer et al. 2016)

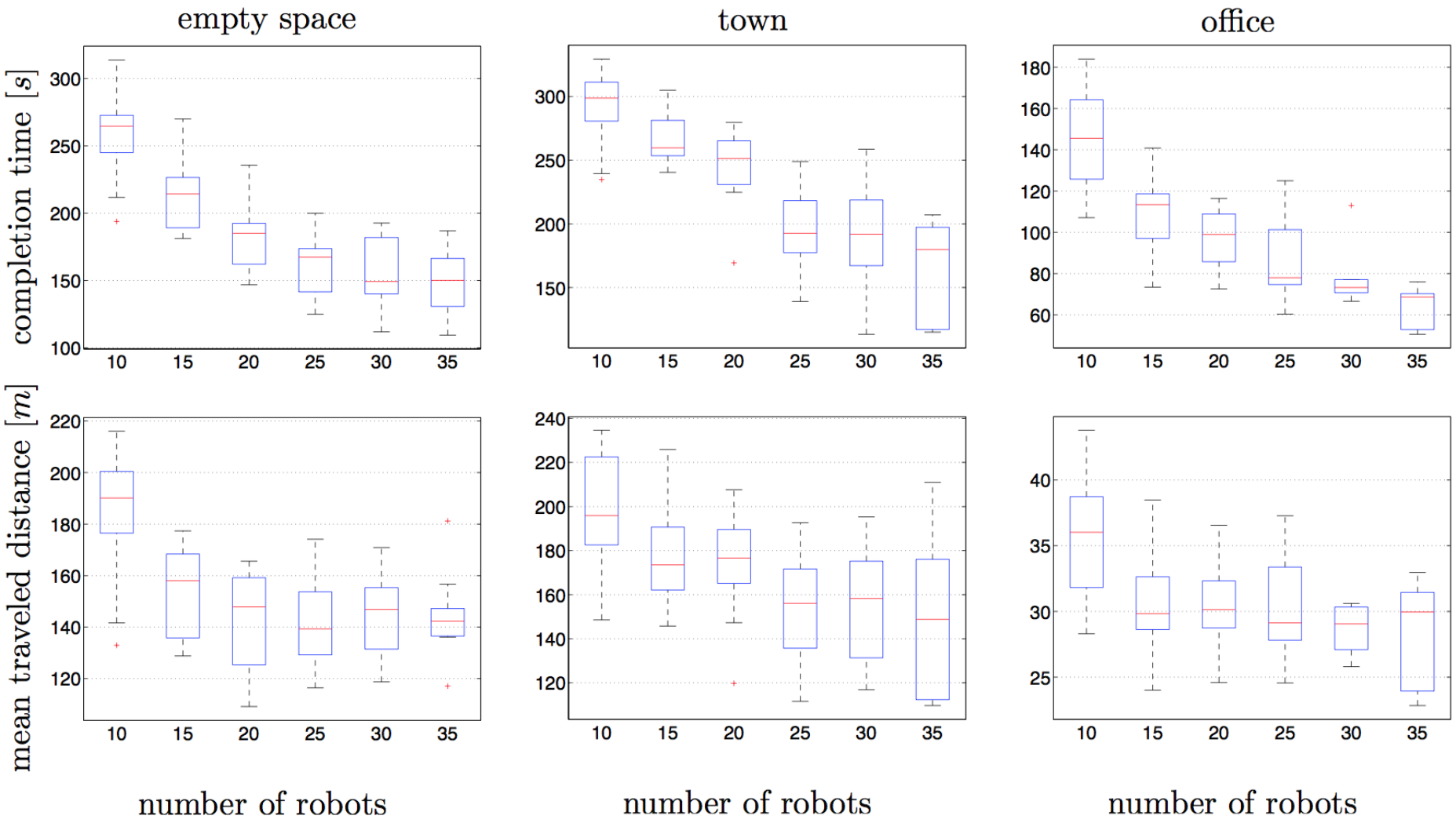
Simulation in Empty Space

Simulation in Office-like Environment

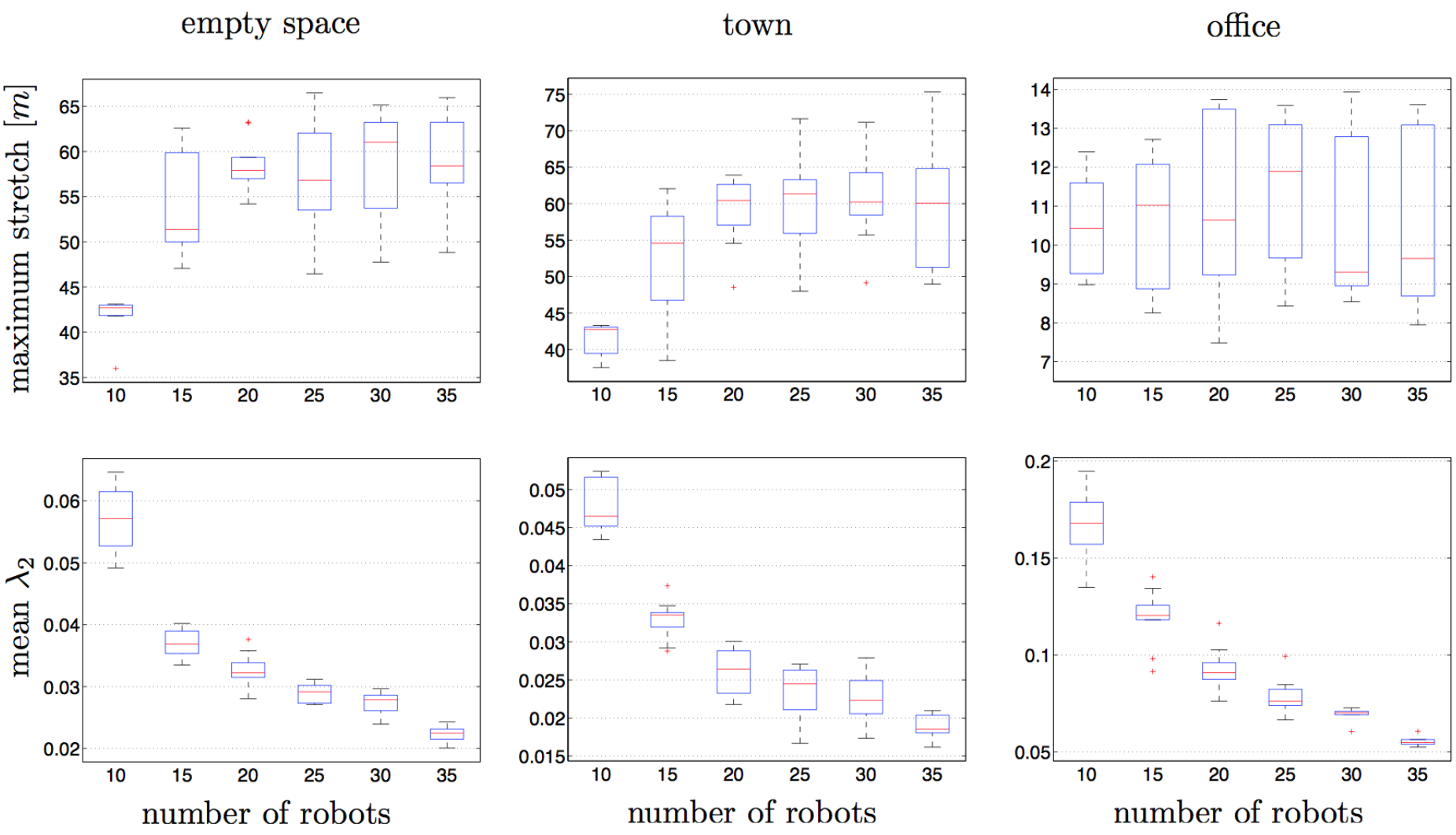
Simulation in Outdoor Environment

Experiment

Multi-Target Exploration with Connectivity



Multi-Target Exploration with Connectivity



Infinitesimal Rigidity Maintenance

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:

$$\dot{\mathbf{x}}_i = \left.\frac{\mathrm{d}V}{\mathrm{d}\mu}\right|_{\varsigma_7(t)} \left.\frac{\partial \varsigma_7}{\partial \mathbf{x}_i}\right|_{(\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{z})} + u_i$$

$$u_i = \begin{cases} v_i^h & \text{if connected to a human} \\ 0 & \text{otherwise} \end{cases}$$

- v_i^h **desired velocity** commanded by a **human**

Experiment

Short Summary and Open Problems

Short summary

- Single **scalars** can define **fundamental global properties**
  - λ_2 Fiedler eigenvalue (Fiedler 1973)
  - ς_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?)

Bearing rigidity (from $SE(3)$ to $SE(3)$)

- D. Zelazo, Franchi, A., and Robuffo Giordano, P., "Rigidity Theory in SE(2) for Unscaled Relative Position Estimation using only Bearing", in 2014 European Control Conference, Strasbourg, France, 2014, pp. 2703-2708.
- D. Zelazo, Robuffo Giordano, P., and Franchi, A., "Bearing-Only Formation Control Using an SE(2) Rigidity Theory", in 54rd IEEE Conference on Decision and Control, Osaka, Japan, 2015
- F. Schiano, Franchi A, Zelazo D, Robuffo Giordano P., "A Rigidity-Based Decentralized Bearing Formation Controller for Groups of Quadrotor UAVs", in 2016 IEEE/RSJ Int. Conf. on Intelligent Robots and System, Daejeon, South Korea, 2016.
- G. Michieletto, Cenedese A, Franchi A. "Bearing Rigidity Theory in SE(3)", in 55th IEEE Conference on Decision and Control, Las Vegas, NV, 2016.

References

- Fiedler, M. (1973). "Algebraic connectivity of Graphs". In: *Czechoslovak Mathematical Journal* 23.98, pp. 298–305.
- Tay, T. and W. Whiteley (1985). "Generating Isostatic Frameworks". In: *Structural Topology* 11.1, pp. 21–69.
- Yang, P., R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar (2010). "Decentralized estimation and control of graph connectivity for mobile sensor networks". In: *Automatica* 46.2, pp. 390–396.
- Robuffo Giordano, P., A. Franchi, C. Secchi, and H. H. Bülthoff (2013). "A Passivity-Based Decentralized Strategy for Generalized Connectivity Maintenance". In: *The International Journal of Robotics Research* 32.3, pp. 299–323.
- Zelazo, D., A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano (2014). "Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems". In: *The International Journal of Robotics Research* 34.1, pp. 105–128.
- Nestmeyer, T., P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi (2016). "Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots". In: *Autonomous Robots*.

Questions?

Decentralized Estimation and Control Methods for Cooperative Robot Motion

Antonio Franchi
CNRS, LAAS, France, Europe

Seoul National University, Engineering
Seoul, South Korea
25th October, 2016

IEEE RAS **Technical Committee** on **Multi-Robot Systems**:

`http://multirobotsystems.org/`

- recently founded (Fall 2014)
- 330 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`