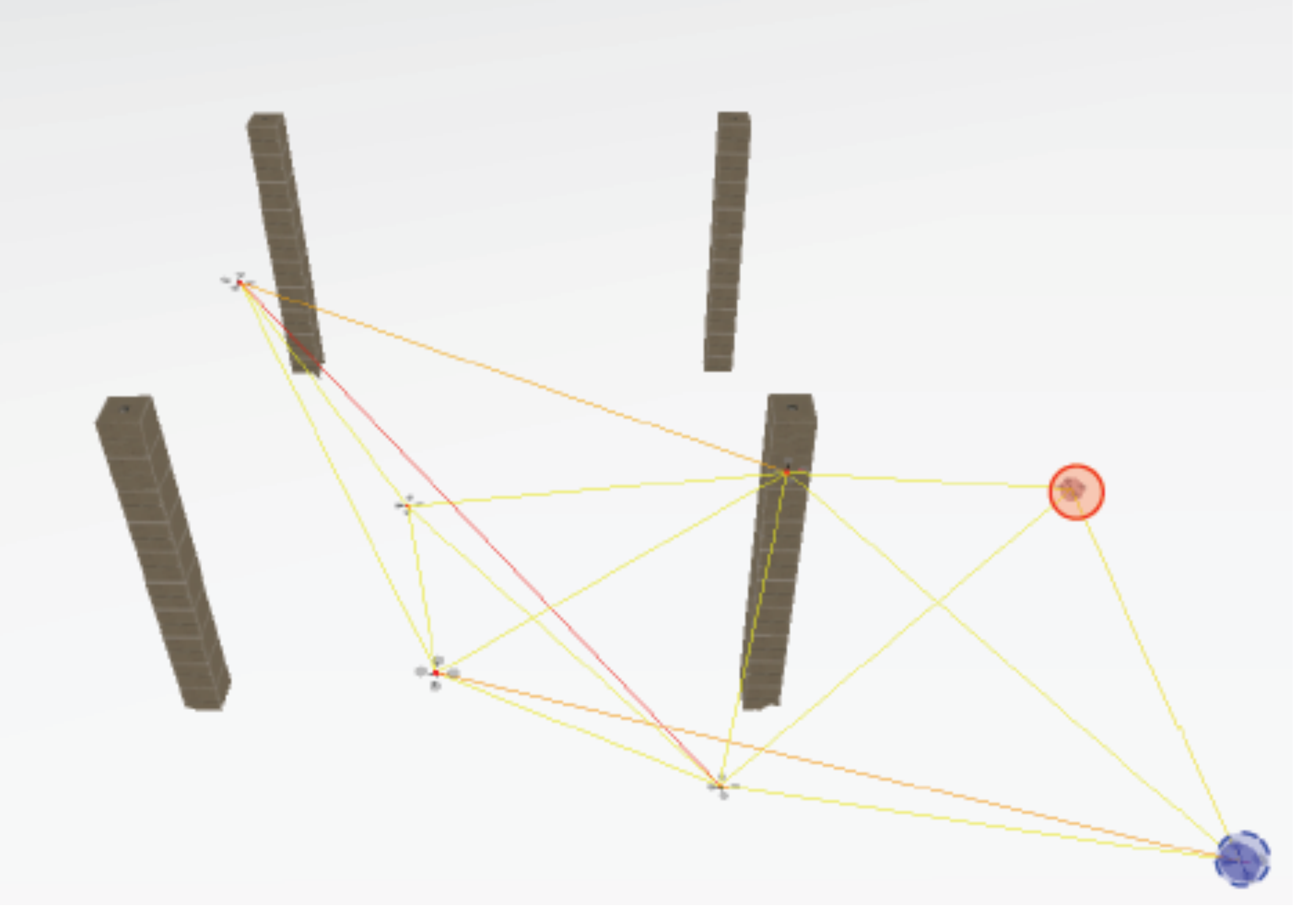
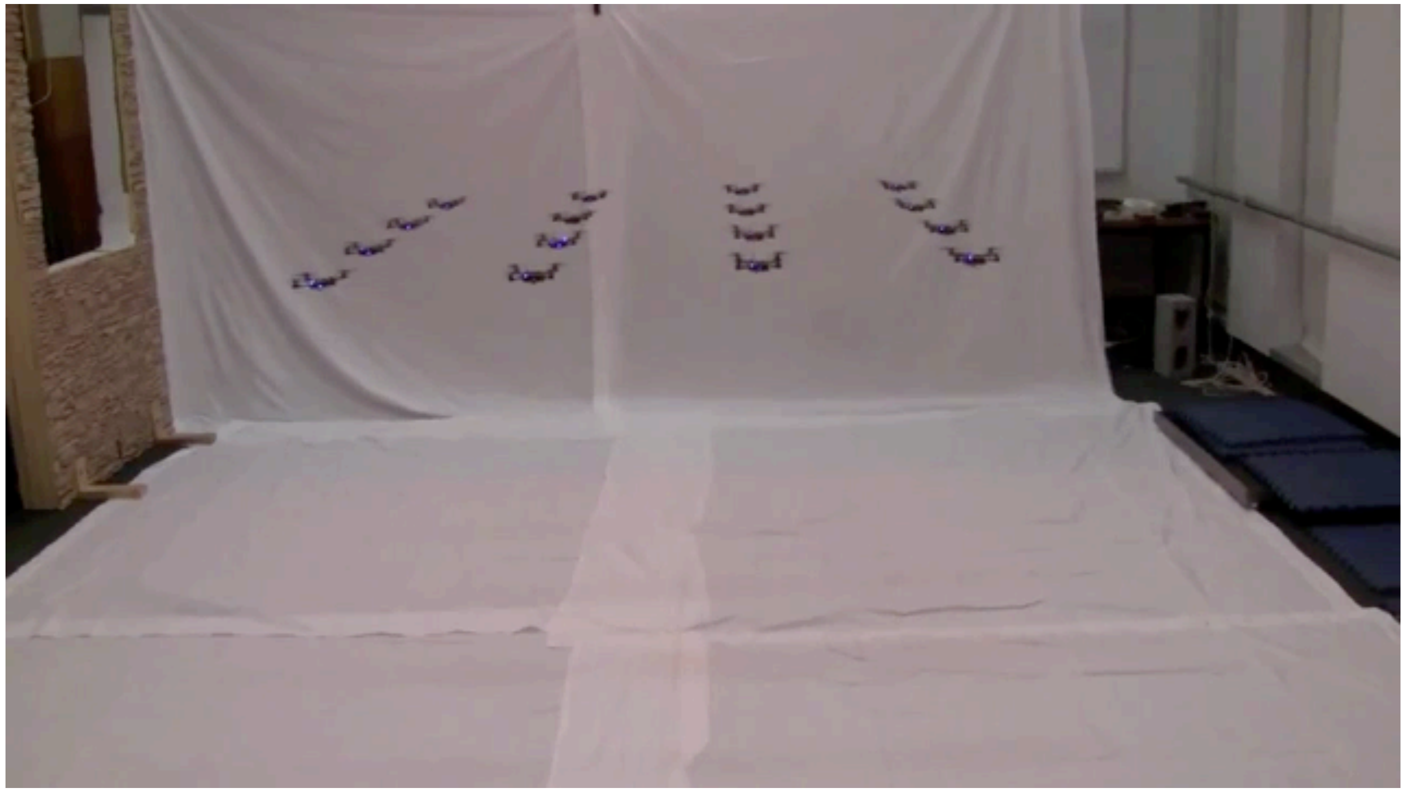
Distributed Rigidity Maintenance with Range-only Sensing



Daniel Zelazo
Faculty of Aerospace Engineering
Technion-Israel Institute of Technology

Tokyo Institute of Technology
September 13, 2013

Coordination in Multi-agent Systems



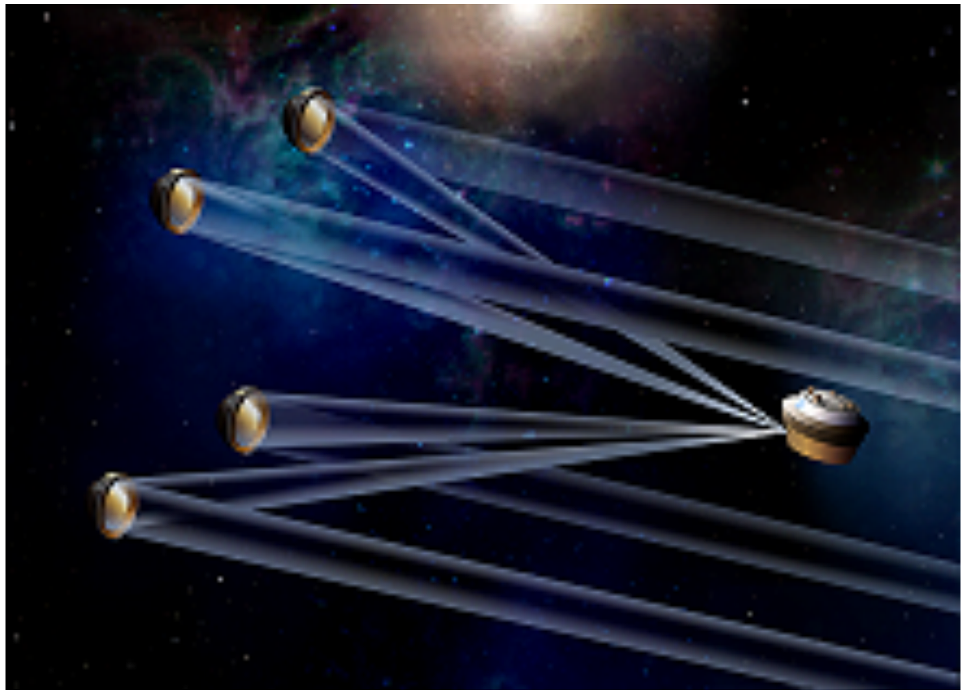
GRASP LABORATORY General Robotics, Automation, Sensing & Perception

System Requirements

- `low-level' control
- sensing and communication
- mission objectives
  - local
  - team
- distributed algorithms

What are the *architectural* requirements for a multi-agent system?

Coordination in Harsh Environments

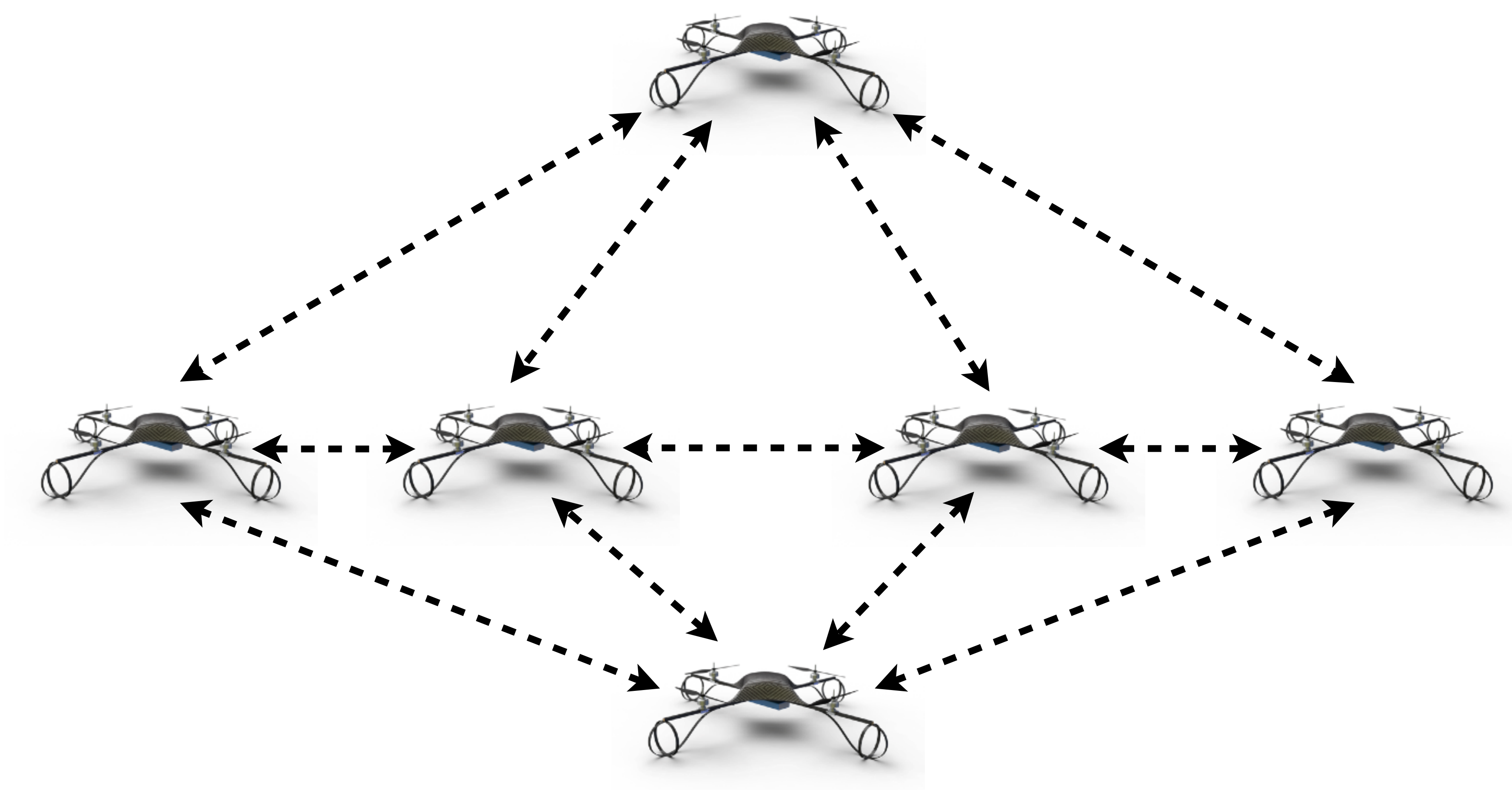


Sensors measuring distances, however,
are very accurate and independent of any
ame

What is th coordination
using rements?

Architectural Requirements

"Connectedness" of the sensing and communication network



Architectural Requirements

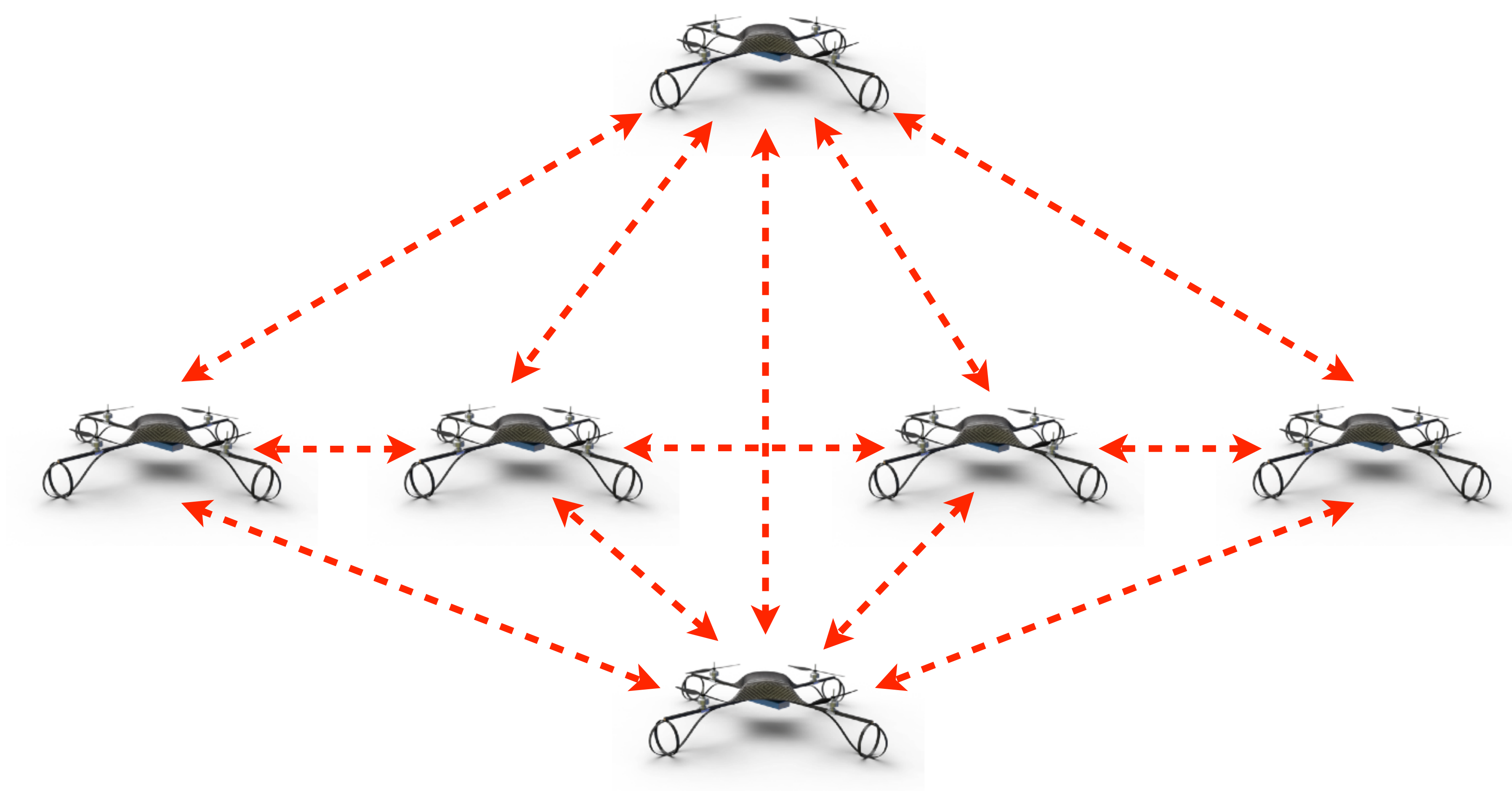
certain “team” objectives and specific sensing/communication capabilities might dictate additional architectural requirements

- formation keeping
- localization

$\Rightarrow$ connectedness might not “be enough”

Architectural Requirements

Architectural Requirements

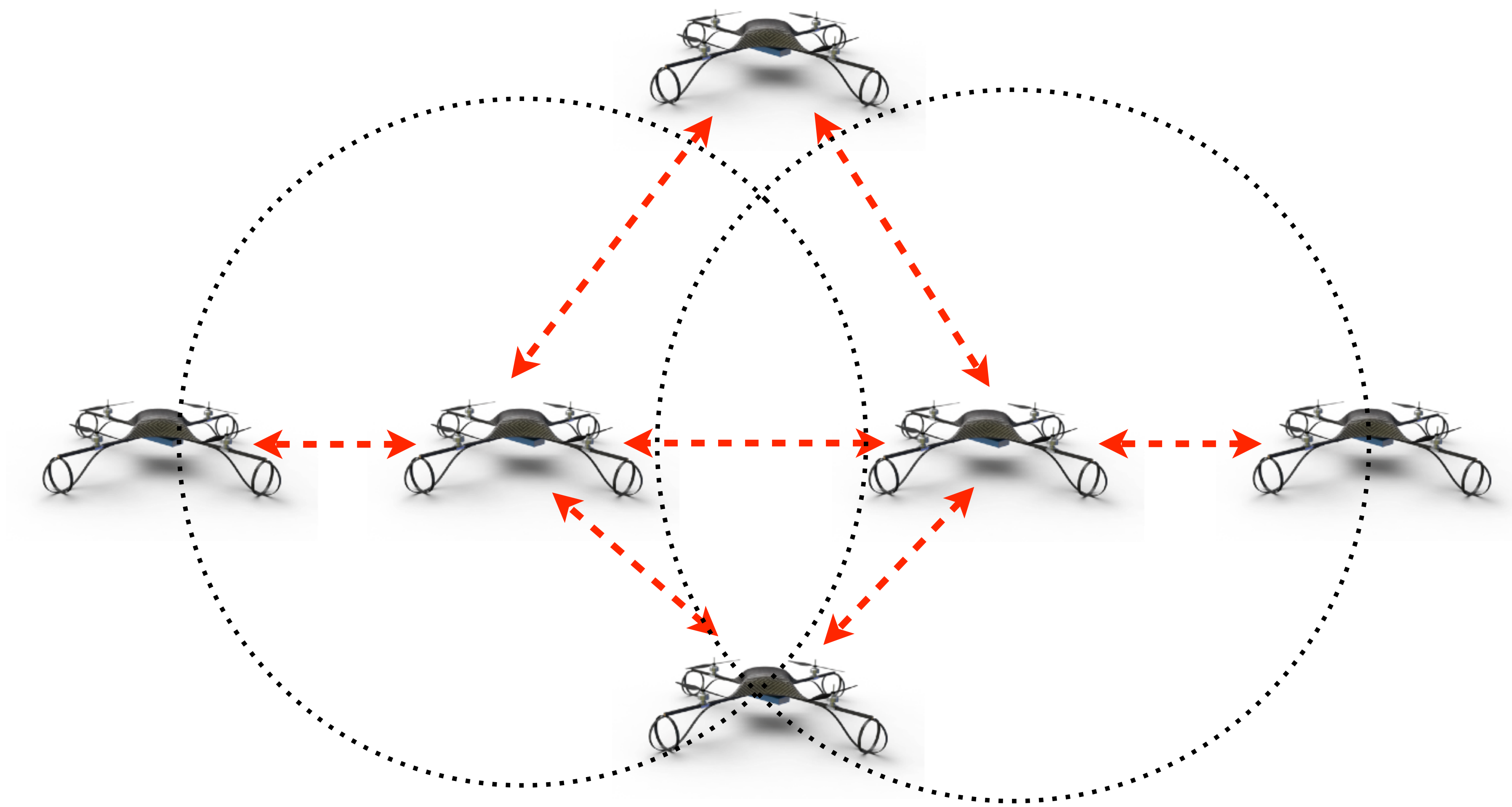


formation specified by a set of inter-agent distances

agents can measure distance to neighbors

sensor limitations only allow a subset of available measurements

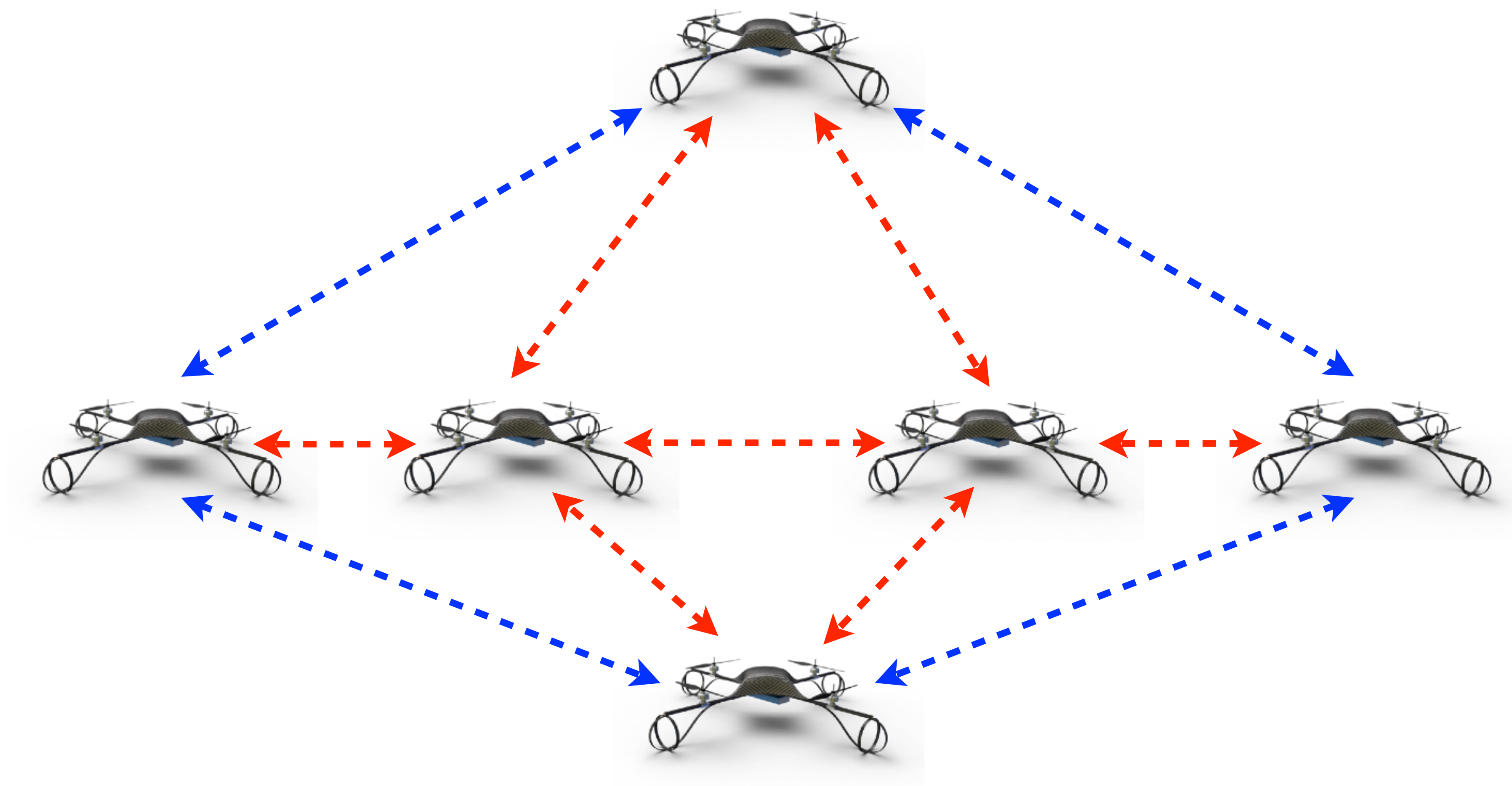
Architectural Requirements



Can the desired formation be maintained using only the available distance measurements?

No!

Architectural Requirements



A *minimum* number of distance measurements are required to *uniquely* determine the desired formation!

Graph Rigidity

Outline

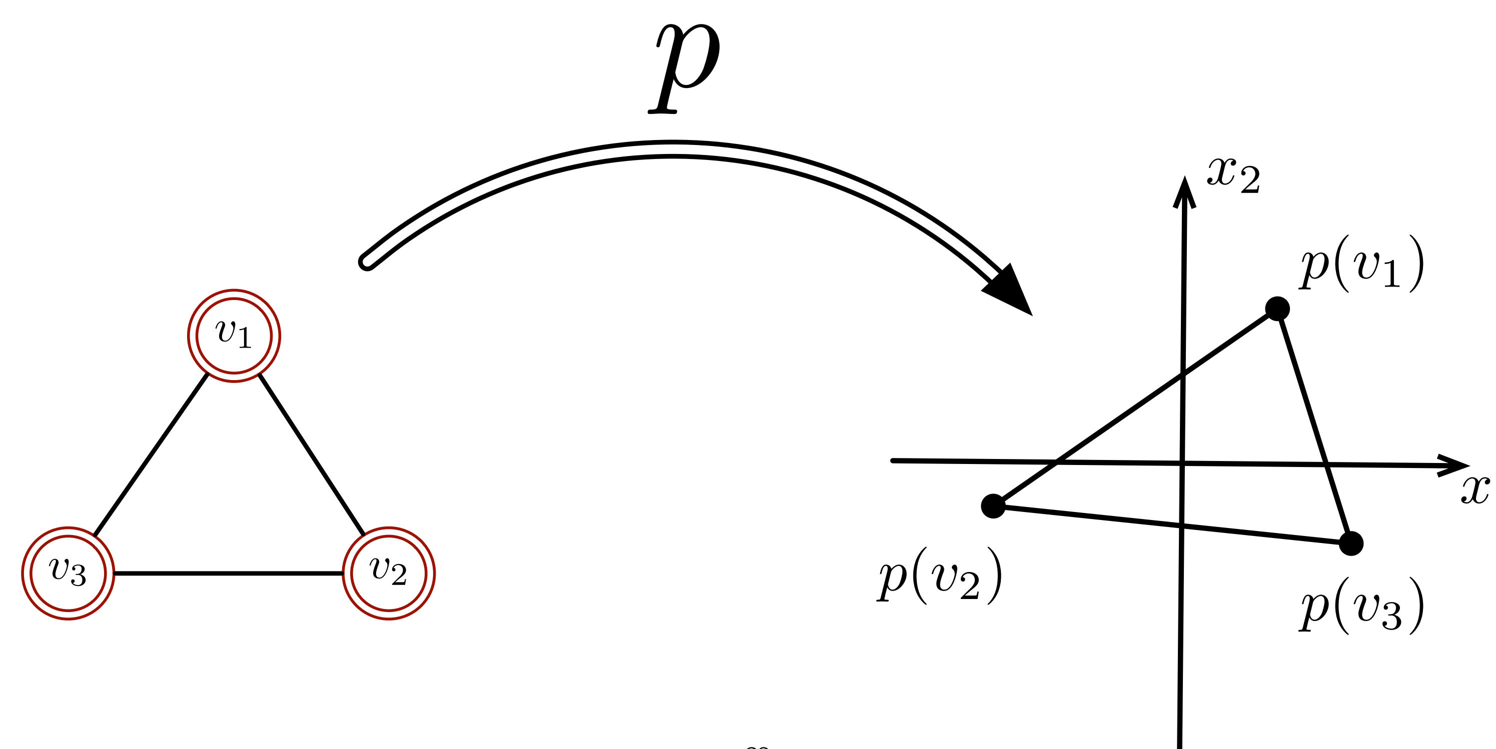
- ✧ Motivation
- ✧ Graph Rigidity and the Rigidity Eigenvalue
- ✧ Distributed Rigidity Maintenance
- ✧ Outlook

Graph Rigidity

bar-and-joint frameworks

$$\begin{cases} \mathcal{G} = (\mathcal{V}, \mathcal{E}) \\ p : \mathcal{V} \to \mathbb{R}^2 \end{cases}$$

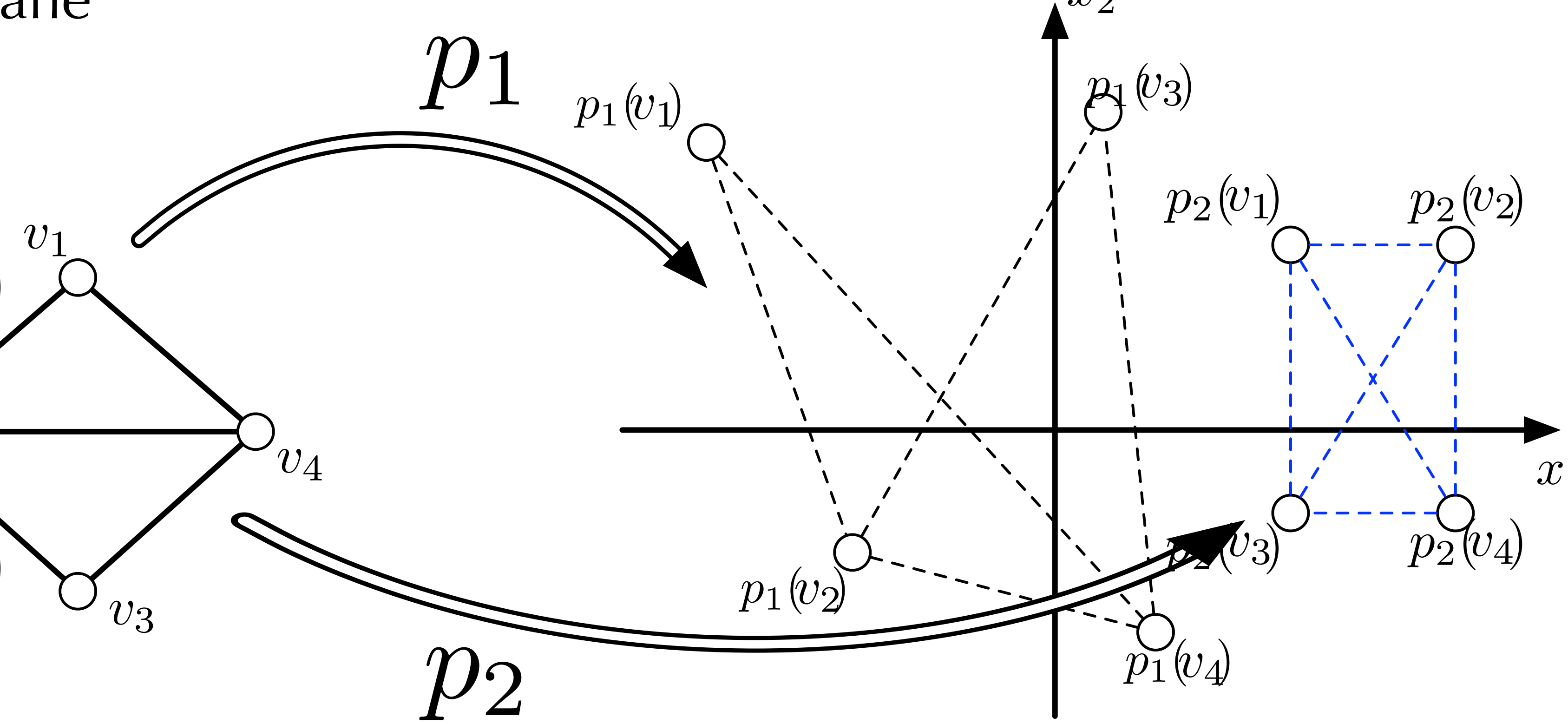
maps every vertex to a point in the plane



example:

$\mathcal{F}_1 = (\mathcal{G}, p_1)$

$\mathcal{F}_2 = (\mathcal{G}, p_2)$

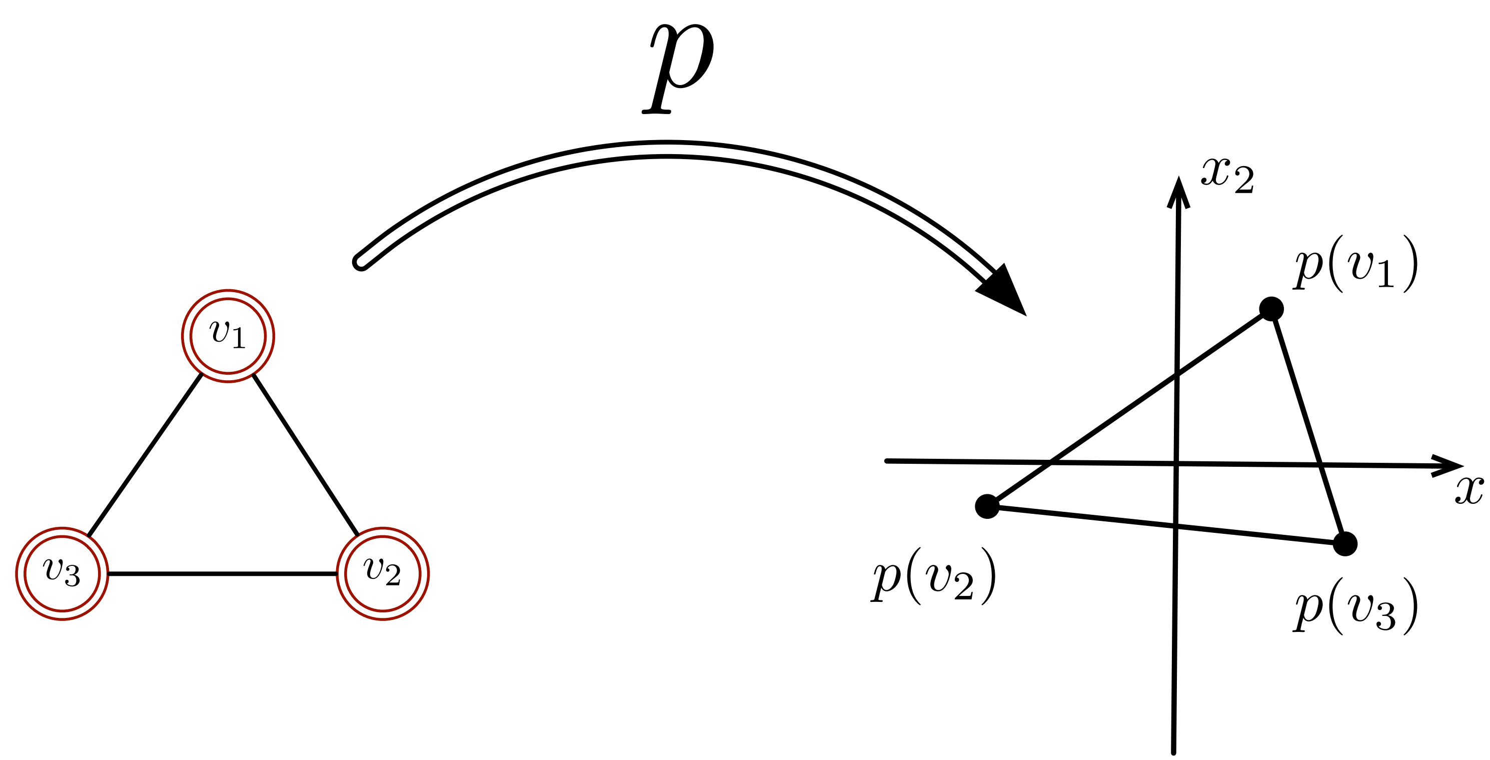


Graph Rigidity

bar-and-joint frameworks

$$\begin{cases} \mathcal{G} = (\mathcal{V}, \mathcal{E}) \\ p : \mathcal{V} \to \mathbb{R}^2 \end{cases}$$

maps every vertex to a point in the plane



Two frameworks are *equivalent if*

$(\mathcal{G}, p_0)$ $(\mathcal{G}, p_1)$

$$\|p_0(v_i) - p_0(v_j)\| = \|p_1(v_i) - p_1(v_j)\|$$

$$\forall \{v_i, v_j\} \in \mathcal{E}$$

Two frameworks are *congruent if*

$(\mathcal{G}, p_0)$ $(\mathcal{G}, p_1)$

$$\|p_0(v_i) - p_0(v_j)\| = \|p_1(v_i) - p_1(v_j)\|$$

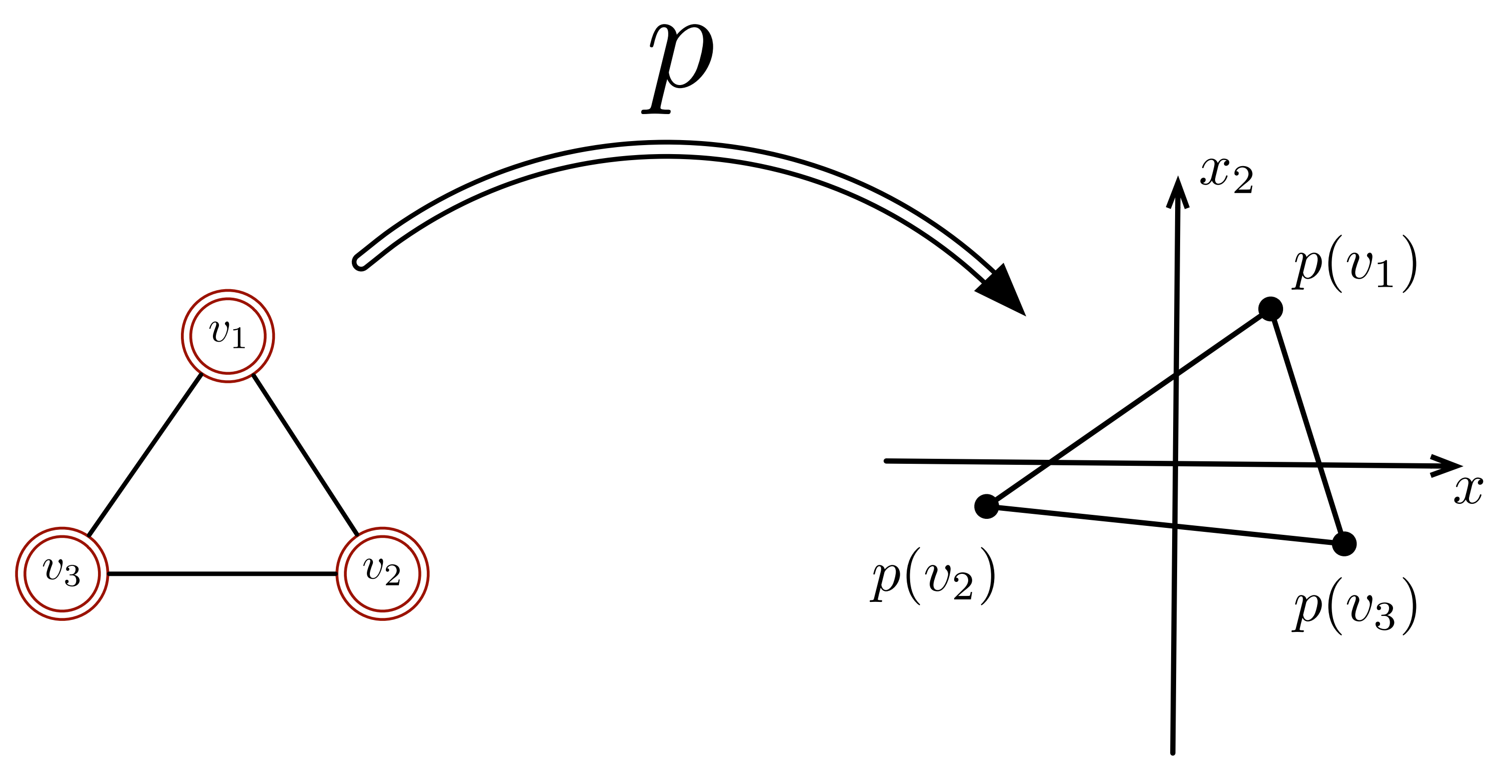
$$\forall v_i, v_j \in \mathcal{V}$$

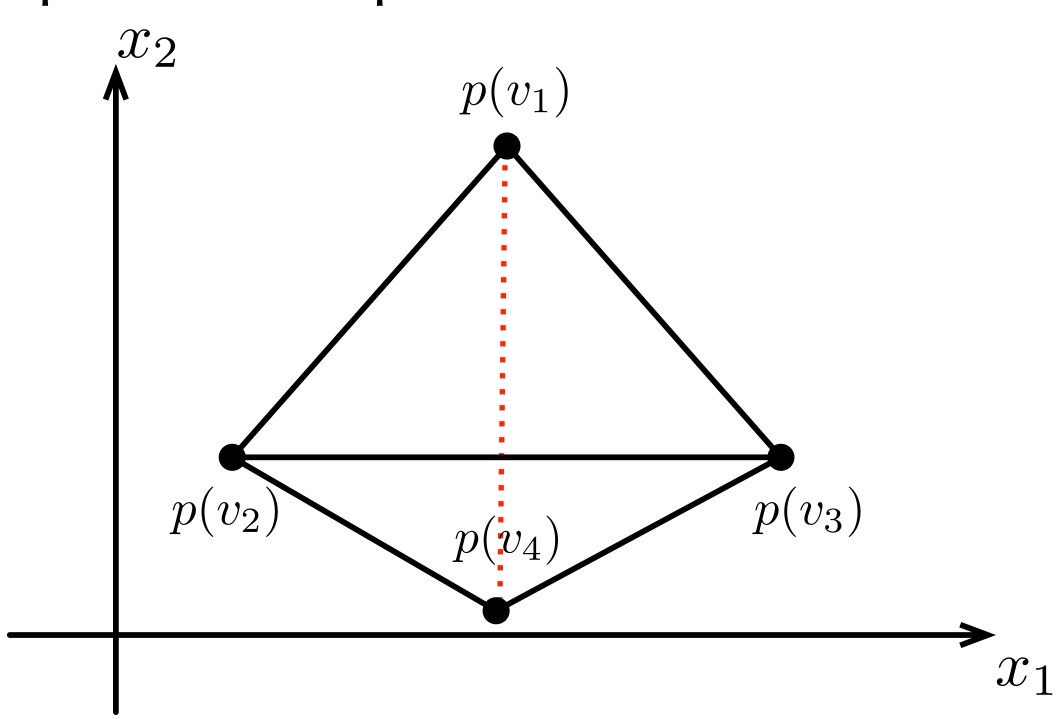
Graph Rigidity

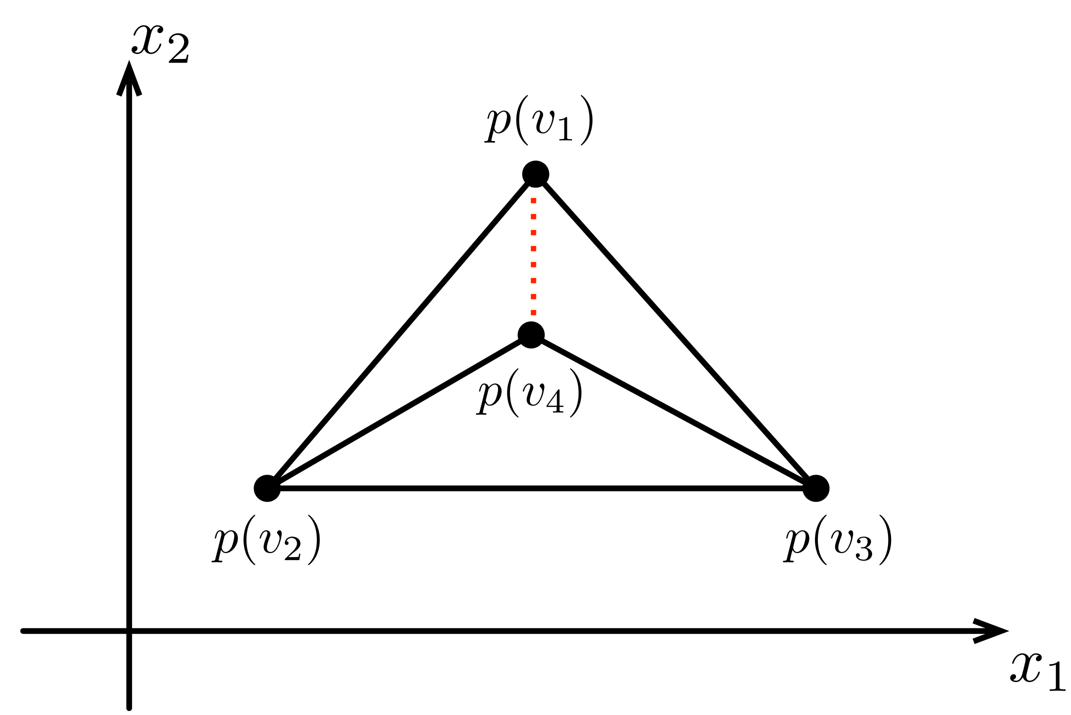
bar-and-joint frameworks

$$\begin{cases} \mathcal{G} = (\mathcal{V}, \mathcal{E}) \\ p : \mathcal{V} \to \mathbb{R}^2 \end{cases}$$

maps every vertex to a point in the plane







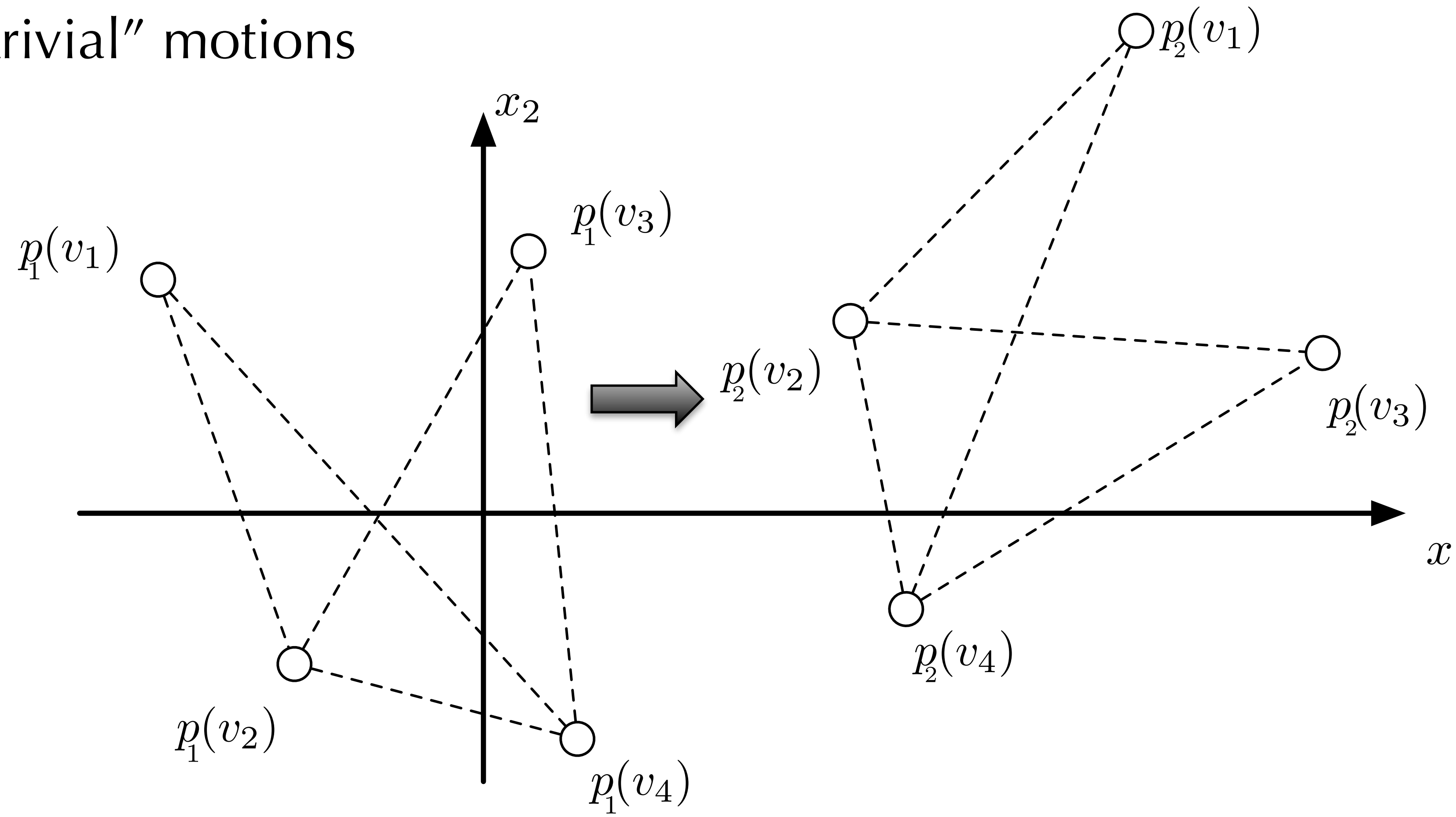
Graph Rigidity

A framework $(\mathcal{G}, p_0)$ is *globally rigid* if every framework that is equivalent to $(\mathcal{G}, p_0)$ is congruent to $(\mathcal{G}, p_0)$.

frameworks that are both *equivalent* and *congruent* are related by only "trivial" motions

- translations
- rotations

minimally rigid

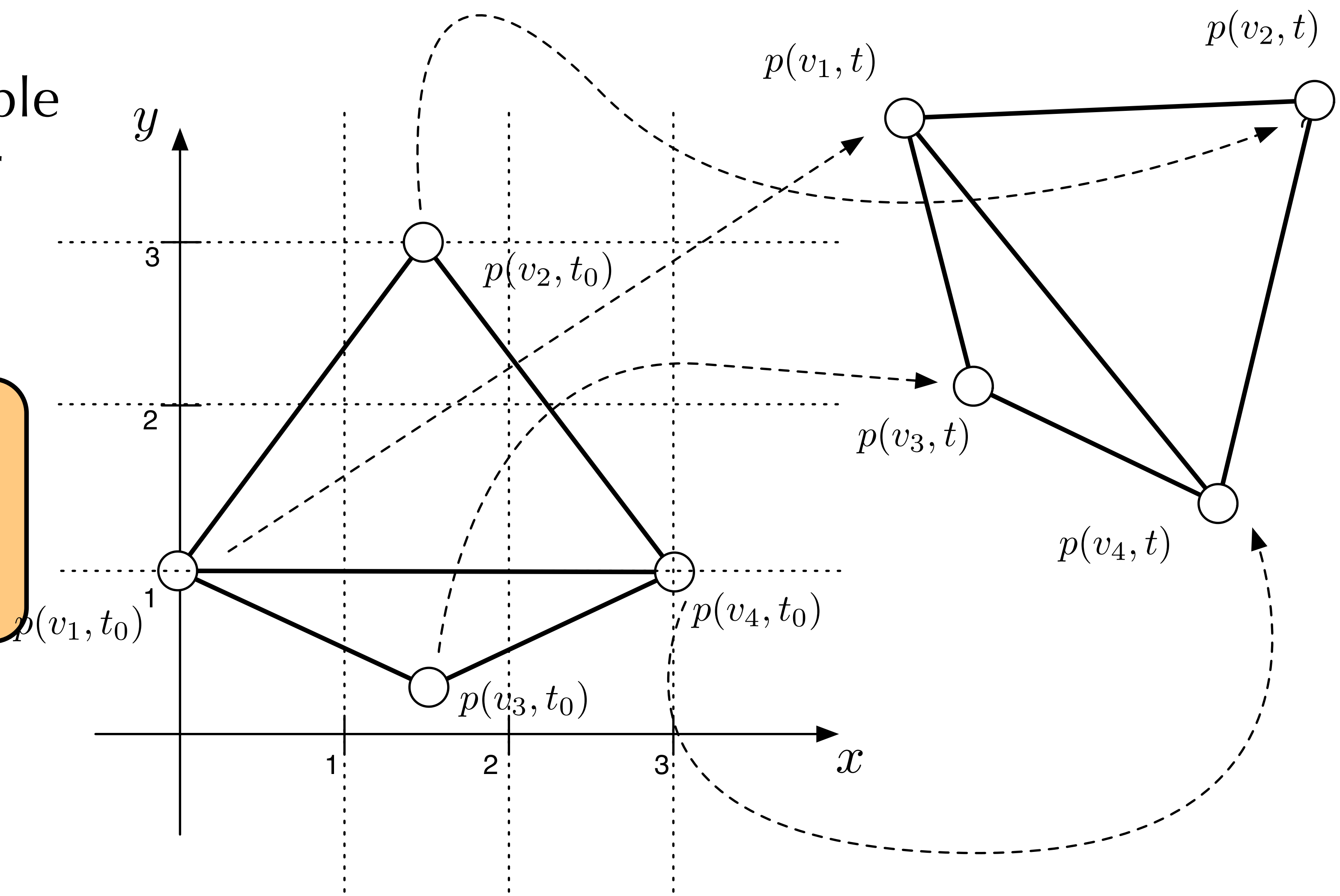


Graph Rigidity

parameterizing frameworks by a variable representing "time" allows to consider "motions" of a framework

$(\mathcal{G}, p, t)$

A trajectory is *edge consistent* if $\|p(v,t) - p(u,t)\|$ is constant for all $\{v,u\} \in \mathcal{E}$ and all t.



edge consistent trajectories generate a family of equivalent frameworks

$$\{p(u) \in \mathbb{R}^2 \mid \|p(u) - p(v)\|_2^2 = \ell_{uv}^2, \forall \{u,v\} \in \mathcal{E}\}$$

$$\Rightarrow \frac{d}{dt}\|x_u(t) - x_v(t)\| = 0, \forall \{u,v\} \in \mathcal{E}$$

$$\Rightarrow (\dot{x}_u(t) - \dot{x}_v(t))^T (x_u(t) - x_v(t)) = 0$$

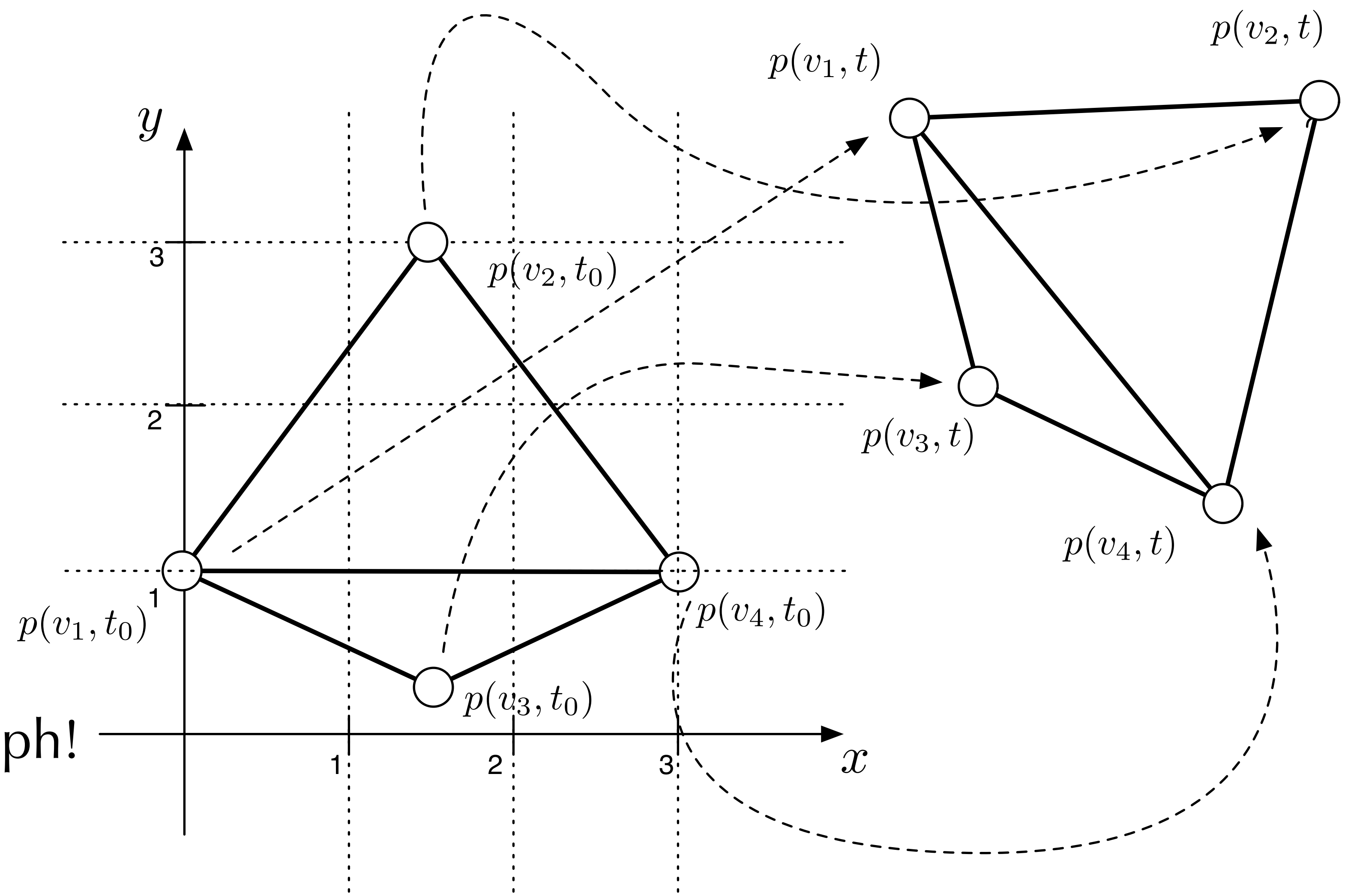
infinitesimal motions

Graph Rigidity

A framework is ***infinitesimally rigid*** if every infinitesimal motion is *trivial*

A *graph* is ***generically rigid*** if it has an infinitesimally rigid framework realization

generic rigidity is a property of the graph!

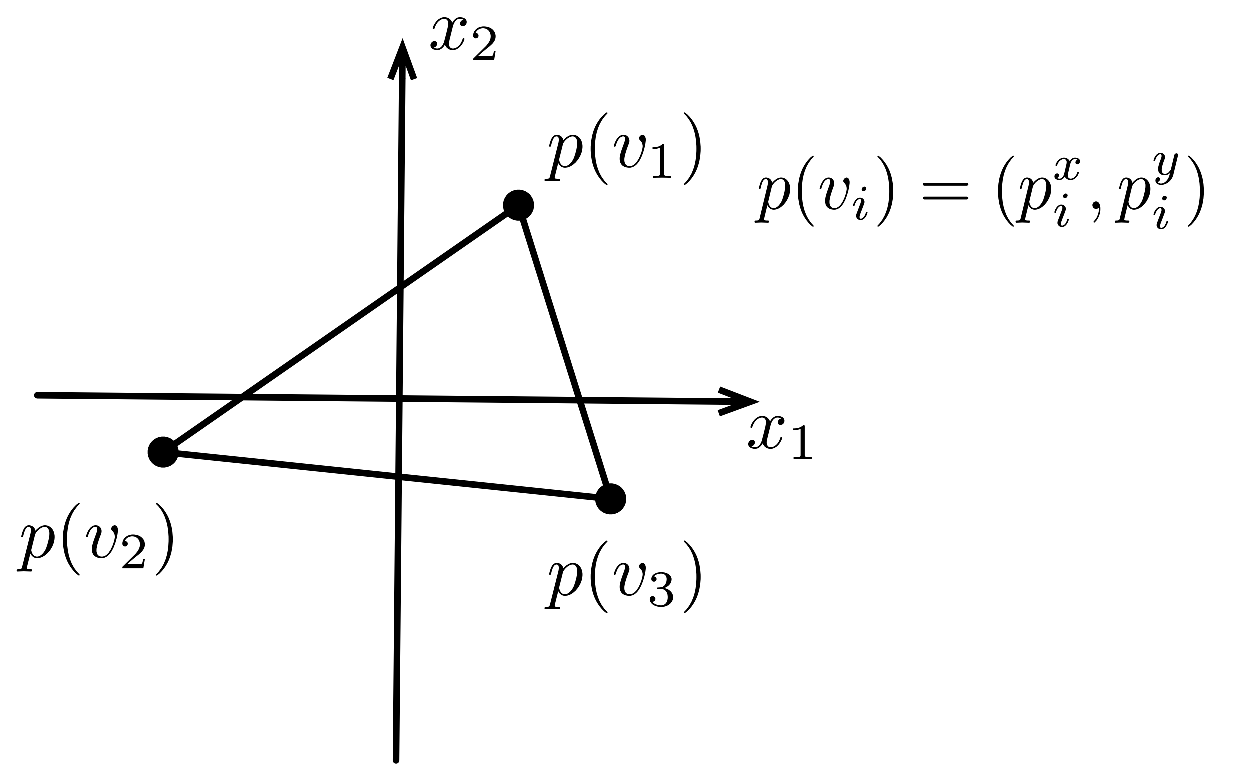


How can we check if a graph is generically or infinitesimally rigid?

The Rigidity Matrix

infinitesimal motions define a system of equations...

$$(\xi(v_i) - \xi(v_j))^T(p(v_i) - p(v_j)) = 0$$



The Rigidity Matrix

$$R(p) \in \mathbb{R}^{|\mathcal{E}|\times 2|\mathcal{V}|}$$

$$R(p) = \begin{bmatrix} p_1^x - p_2^x & p_1^y - p_2^y & p_2^x - p_1^x & p_2^y - p_1^y & 0 & 0 \\ p_1^x - p_3^x & p_1^y - p_3^y & 0 & 0 & p_3^x - p_1^x & p_3^y - p_1^y \\ 0 & 0 & p_2^x - p_3^x & p_2^y - p_3^y & p_3^x - p_2^x & p_3^y - p_2^y \end{bmatrix}$$

Lemma 1 (Tay1984) *A framework $(\mathcal{G}, p)$ is infinitesimally rigid if and only if* $\mathbf{rk}[R] = 2|\mathcal{V}| - 3$

The Symmetric Rigidity Matrix

The Symmetric Rigidity Matrix

$$\mathcal{R} = R(p)^T R(p)$$

λ_4 the *Rigidity Eigenvalue*

(λ_7)

a symmetric positive semi-definite matrix with eigenvalues

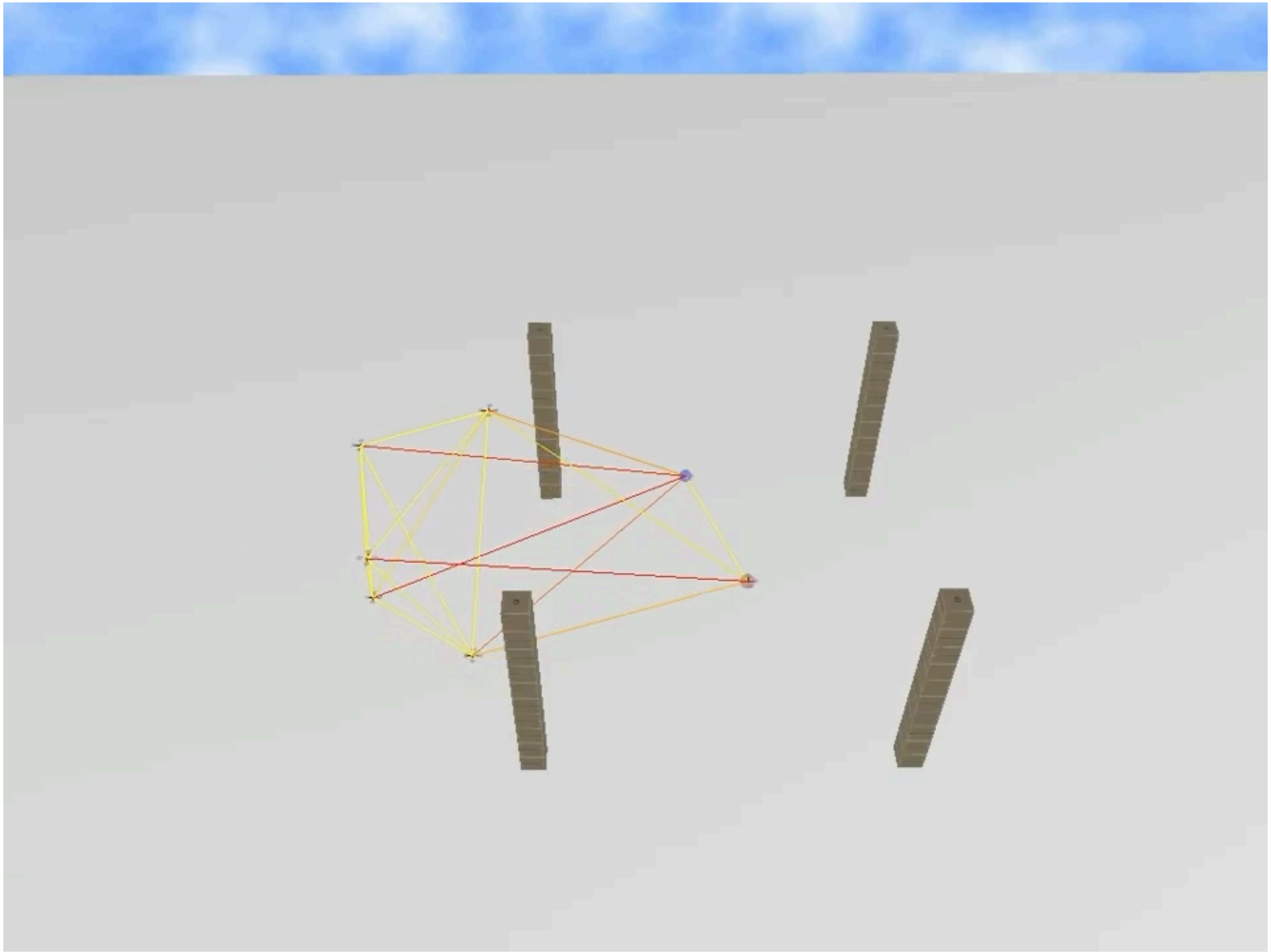
$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{2|\mathcal{V}|}$$

Theorem 1 *A framework is infinitesimally rigid if and only if the* rigidity eigenvalue *is strictly positive; i.e.* $\lambda_4 > 0$.

proof: $$P\mathcal{R}P^T = (I_2 \otimes E(\mathcal{G})) \begin{bmatrix} W_x^2 & W_{xy} \\ W_{xy} & W_y^2 \end{bmatrix} (I_2 \otimes E(\mathcal{G})^T)$$

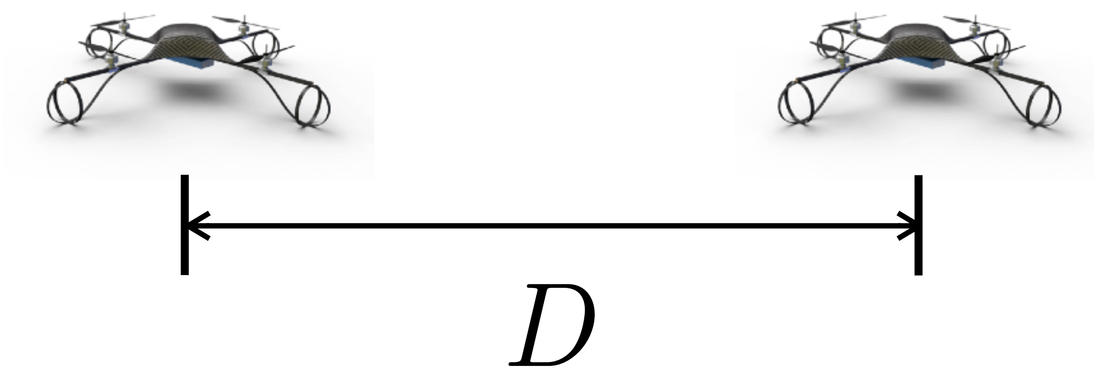
weights depend on *relative positions* $[W_x^2]_{kk} = (p_i^x - p_j^x)^2$

Frameworks for Dynamic Environments

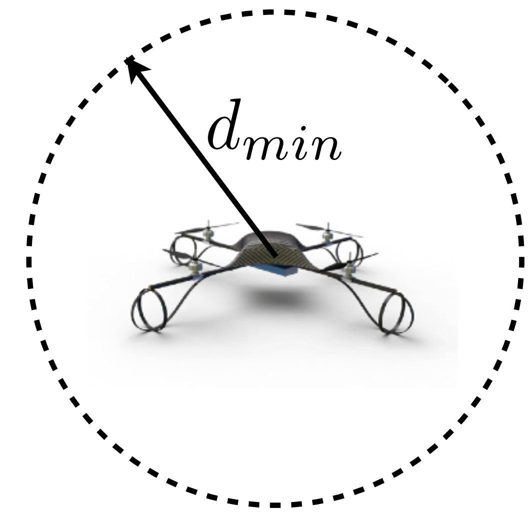


Weighted Frameworks

When is there a sensing link between agents?



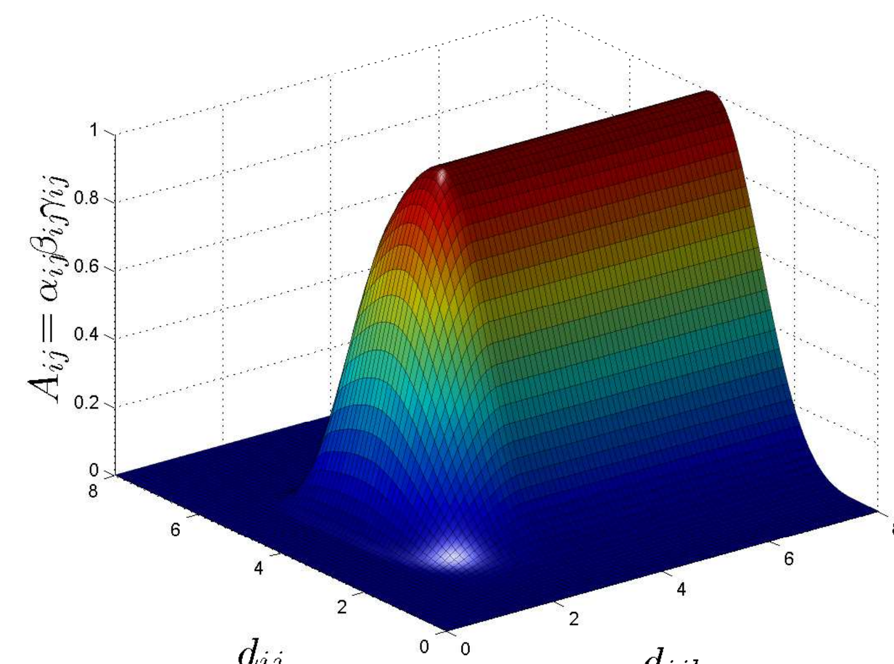
sensing range

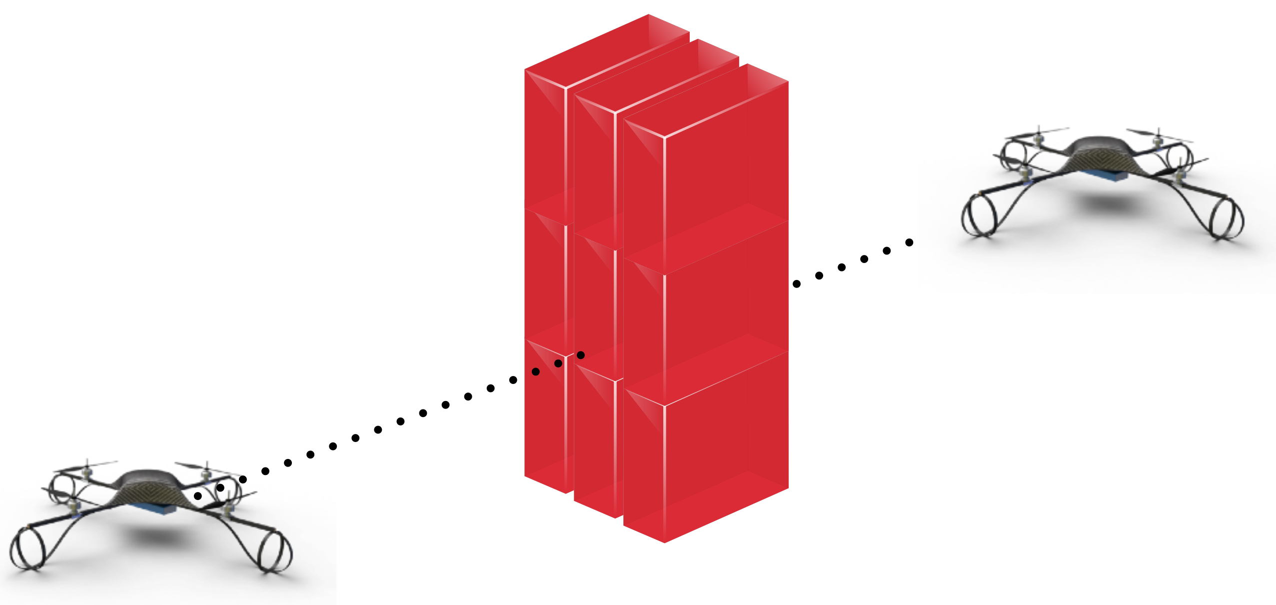


safety zone

composite weight between neighboring agents

A_{ij}



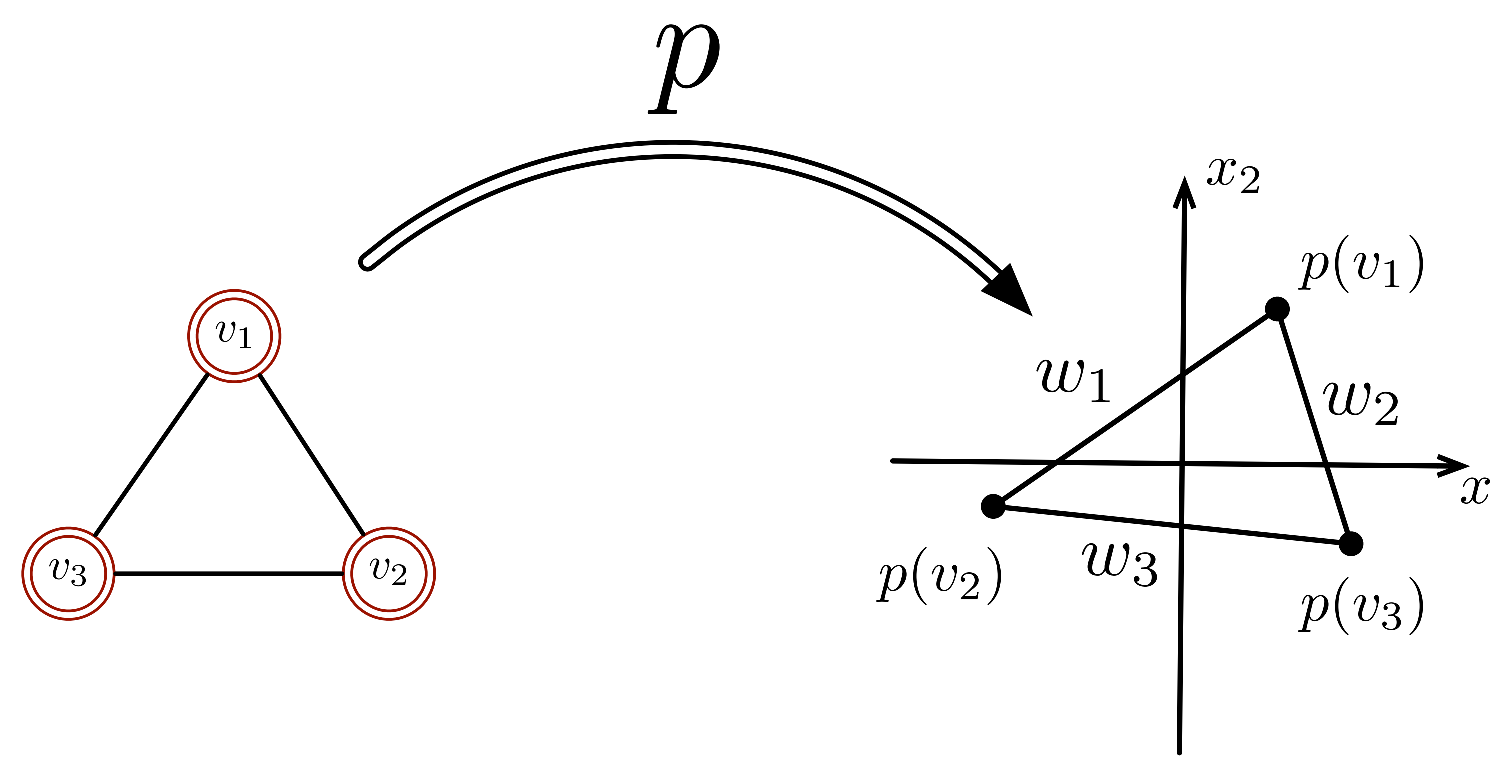


no line-of-sight occlusion

Weighted Frameworks

weighted frameworks

$$\begin{cases} \mathcal{G} = (\mathcal{V}, \mathcal{E}) \\ p : \mathcal{V} \to \mathbb{R}^2 \\ \mathcal{W} : (\mathcal{G}, p) \to \mathbb{R}^{|\mathcal{E}|} \end{cases}$$



weighted rigidity matrix

$$R(p, \mathcal{W}) = W(\mathcal{G}, p) R(p)$$

weighted rigidity matrix

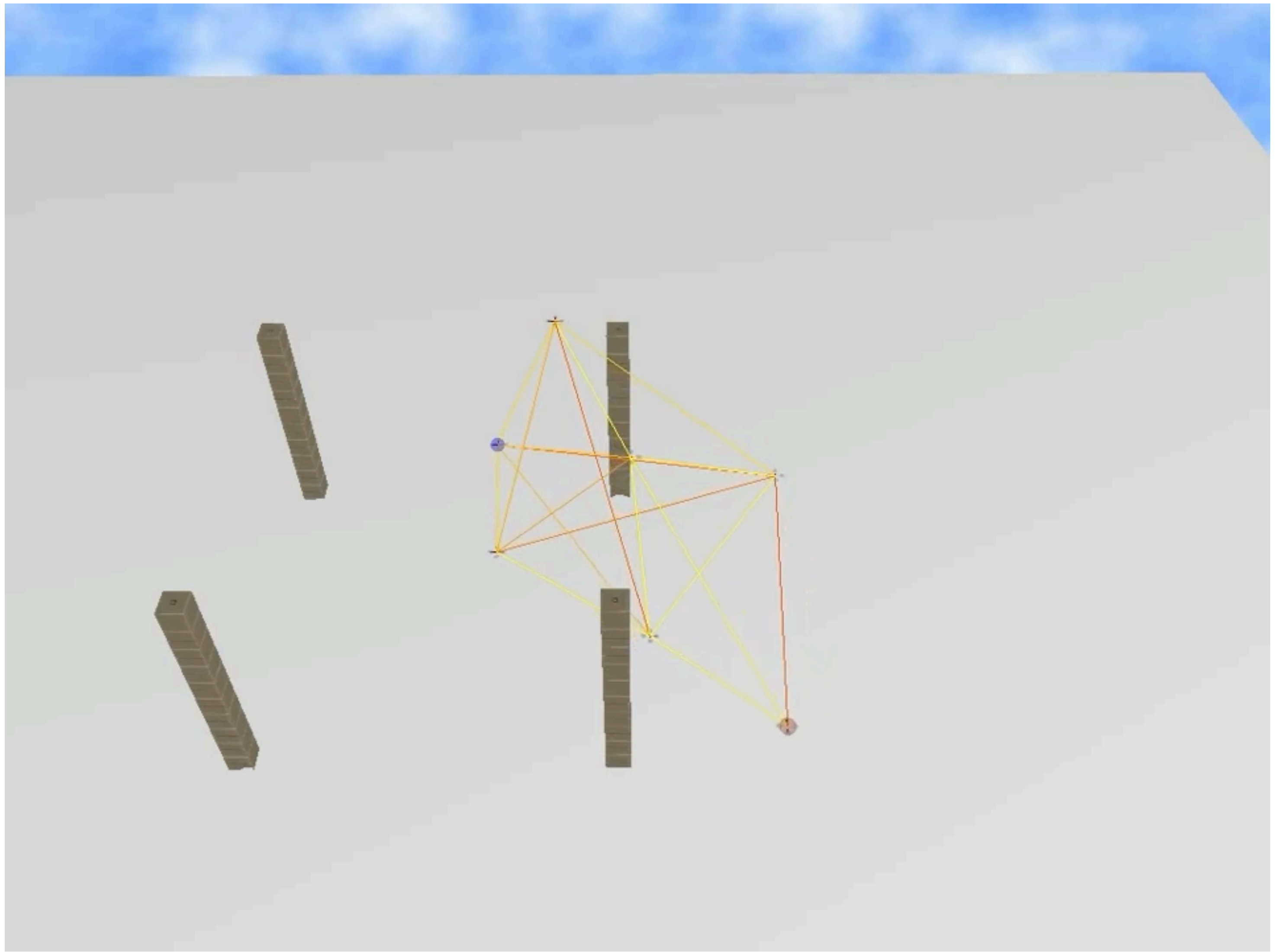
$$\mathcal{R} = R(p, \mathcal{W})^T R(p, \mathcal{W})$$

Corollary 1 *A weighted framework* $(\mathcal{G}, p, \mathcal{W})$ *is infinitesimally rigid if and only if the weighted rigidity eigenvalue is strictly positive.*

Outline

- ✧ Motivation
- ✧ Graph Rigidity and the Rigidity Eigenvalue
- ✧ Distributed Rigidity Maintenance
- ✧ Outlook

Rigidity Maintenance



When *relative sensing* is used, rigidity becomes an important *architectural requirement* for a multi-agent system

$\Longrightarrow$ to achieve higher level objectives (i.e. formation control, localization), the rigidity property must be maintained *dynamically*

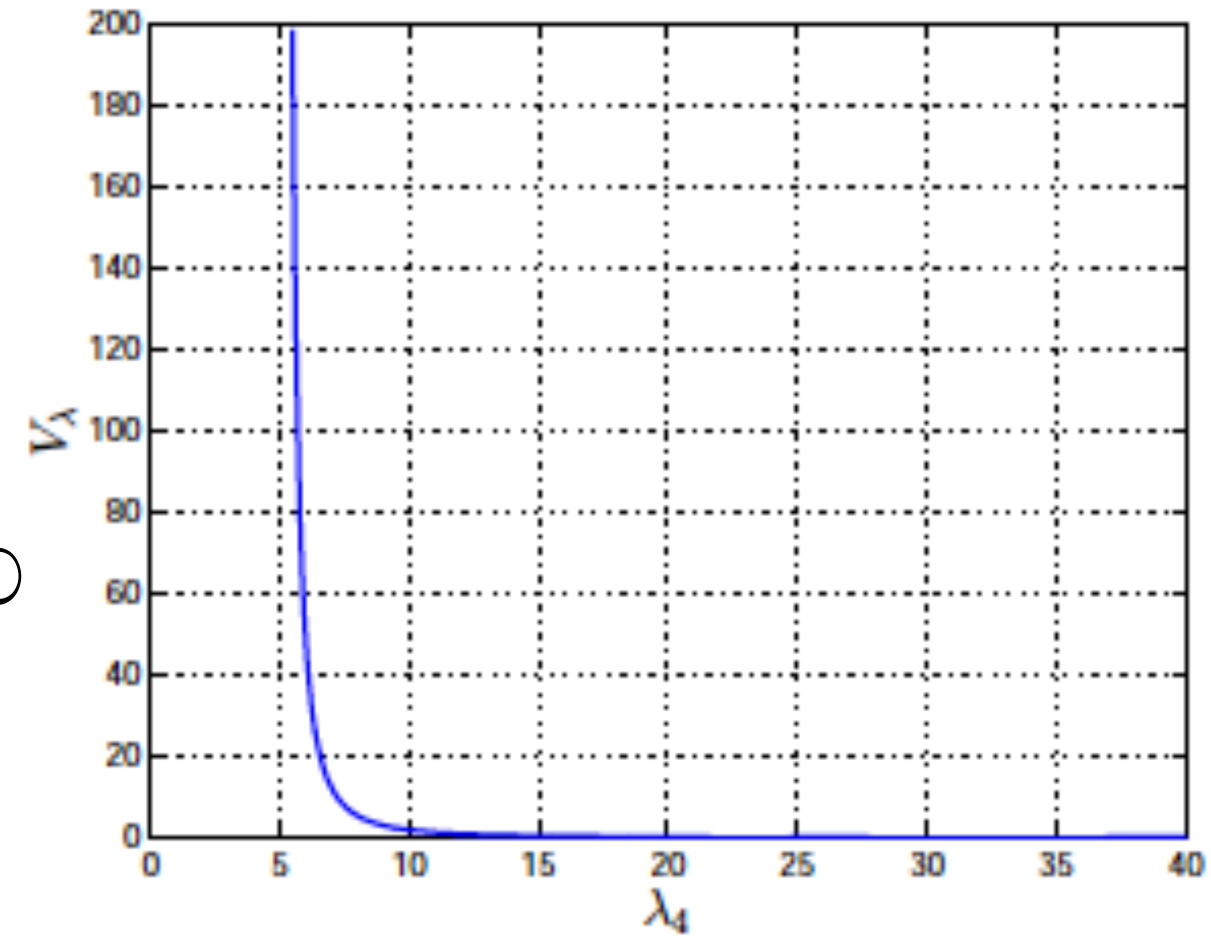
The Rigidity Potential

How can rigidity be maintained with only local information?

Define a scalar potential function V_λ

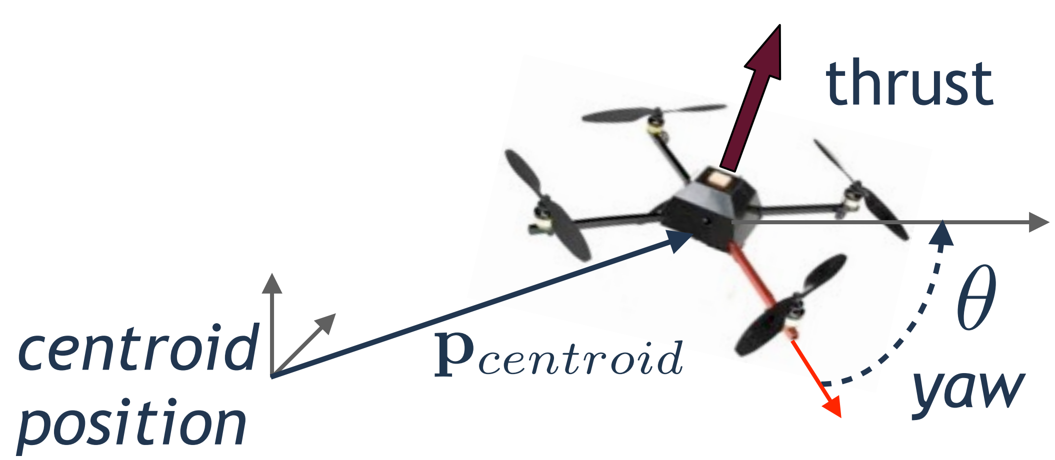
grows unbounded as $\lambda_4 \to 0$

vanishes as $\lambda_4 \to \infty$



velocity command

$$\xi_i = -\frac{\partial V_\lambda}{\partial \lambda_4}\left(\frac{\partial \lambda_4}{\partial p_i}\right)$$



treat like integrator

The Rigidity Potential

How can rigidity be maintained with only local information?

Key observation: Gradient of rigidity eigenvalue has a distributed structure!

$$\lambda_4 = v_4^T P\mathcal{R}P^T v_4$$

$$P\mathcal{R}P^T = (I_2 \otimes E(\mathcal{G})) \begin{bmatrix} W_x & W_{xy} \\ W_{xy} & W_y \end{bmatrix} (I_2 \otimes E(\mathcal{G})^T)$$

requires a common inertial reference frame

"global" quantity

$$\frac{\partial \lambda_4}{\partial p_i^x} = 2\sum_{i\sim j} \mathcal{W}_{ij} \left((p_i^x - p_j^x)(v_i^x - v_j^x)^2 + (p_i^y - p_j^y)(v_i^x - v_j^x)(v_i^y - v_j^y)\right) + \frac{\partial \mathcal{W}_{ij}}{\partial p_i^x}(*)$$

gradient is only a function of *relative* quantities!

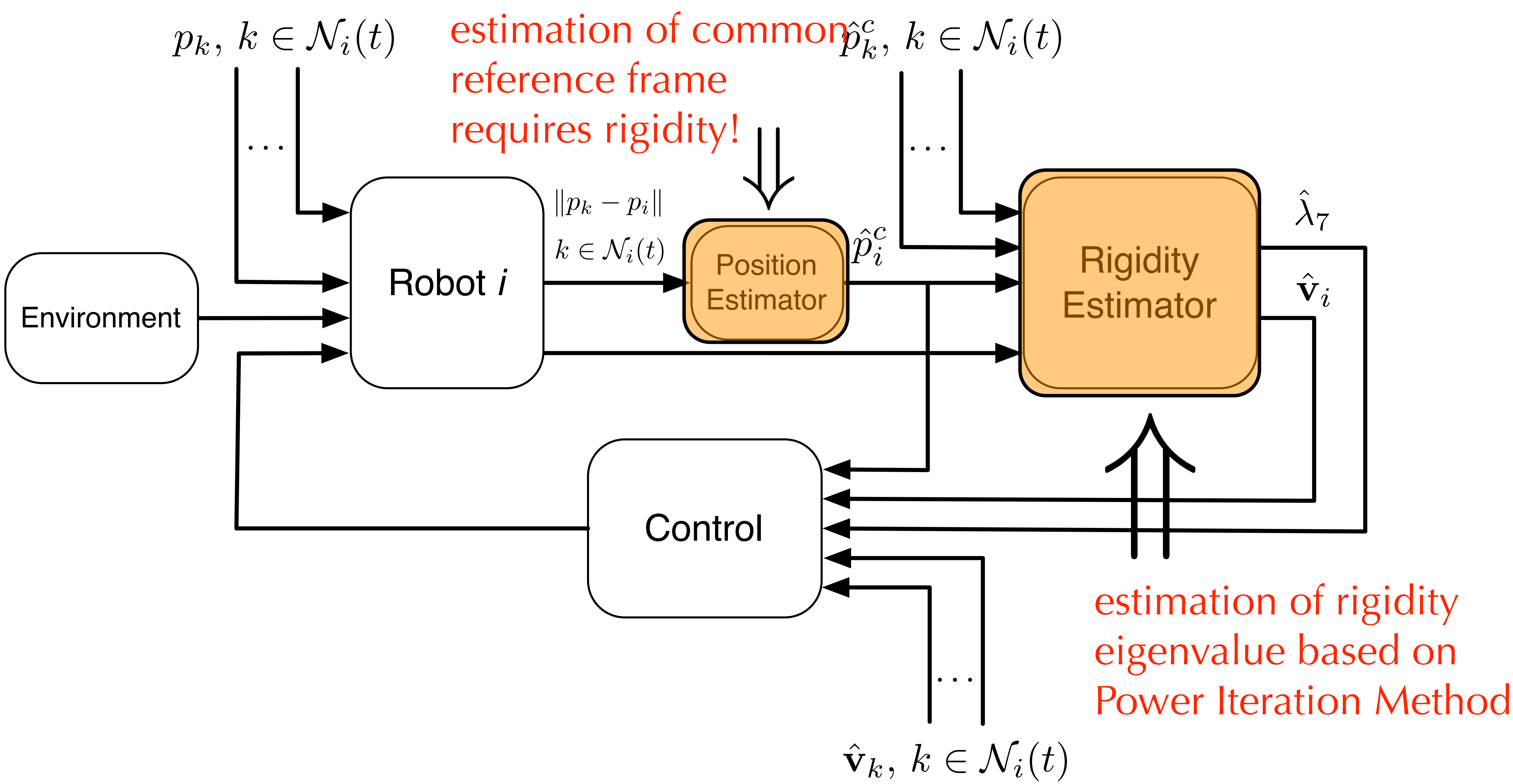
can be computed locally by each agent*

$$\frac{\partial \mathcal{W}_{ij}}{p_i} = 0 \Leftrightarrow j \notin \mathcal{N}_i$$

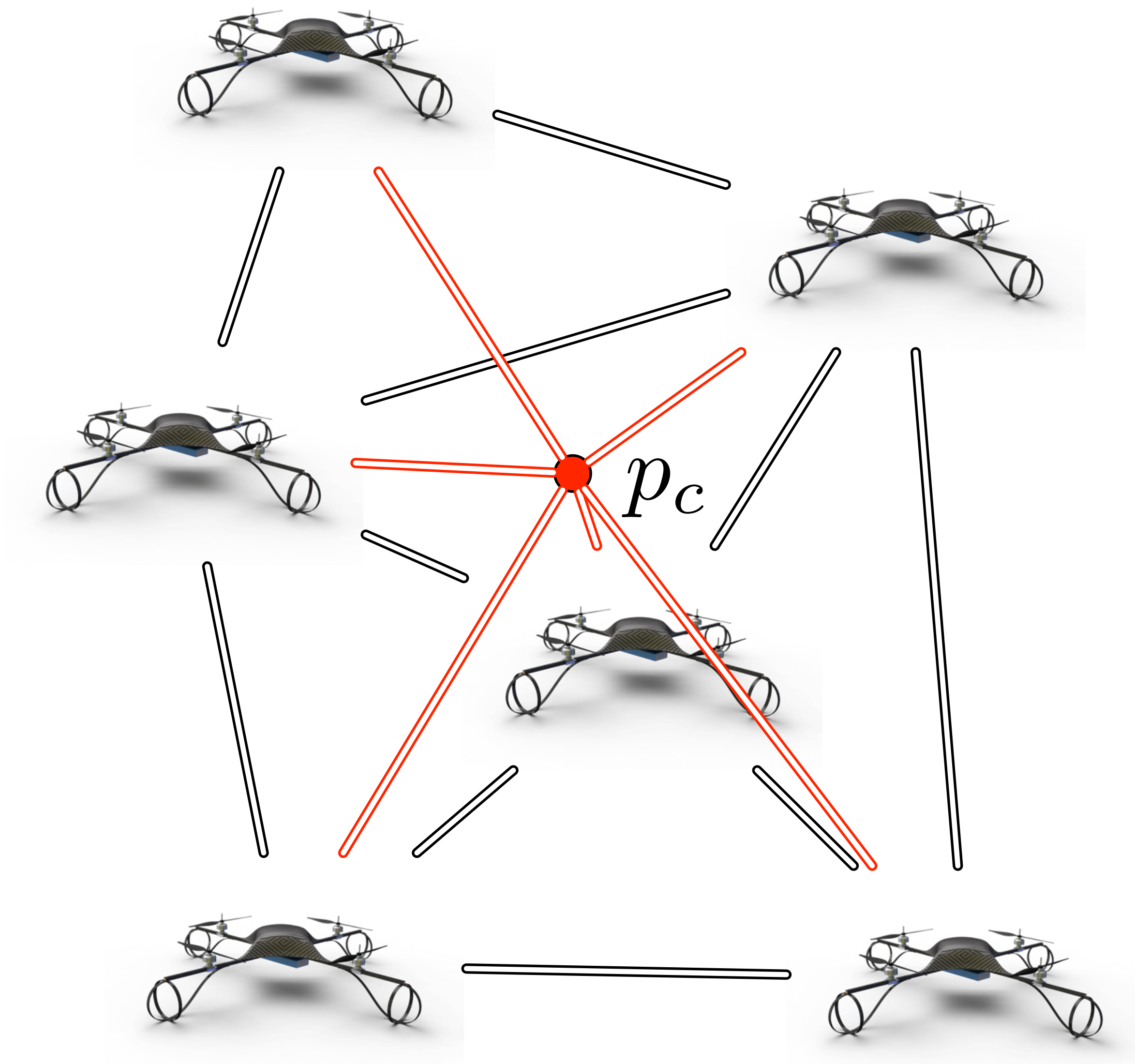
Distributed Rigidity Maintenance

a distributed implementation requires

- estimation of a common inertial frame
- estimation of the rigidity eigenvalue and eigenvector



Estimation of a Common Frame



Agents do not have access to relative positions, only distance

$$p_i - p_j \qquad \|p_i - p_j\|$$

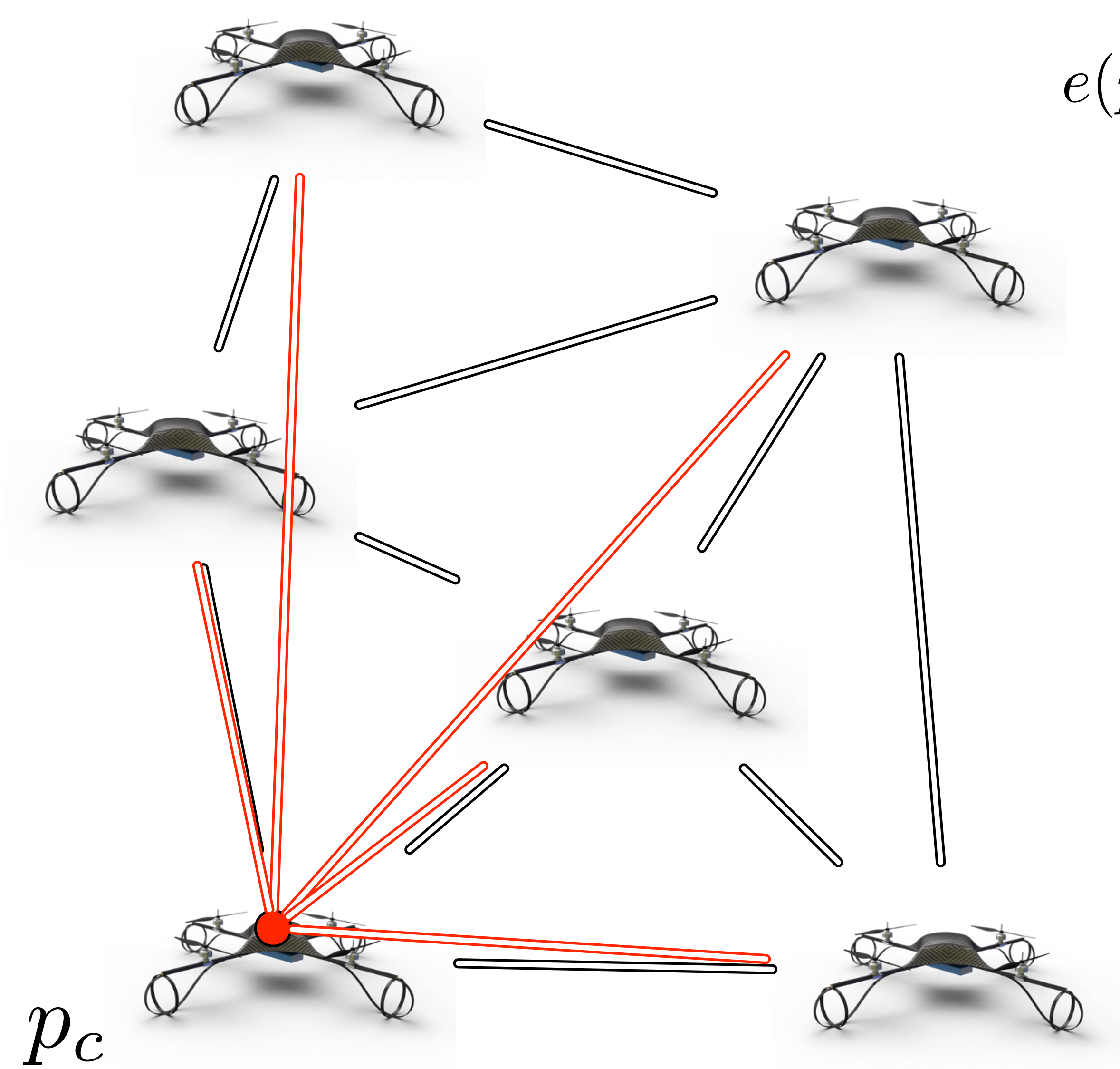
rigidity of formation can be used for each agent to estimate relative position to a *common point*

$$p_i - p_c$$

- one agent endowed with *special ability*
- able to measure *relative position* w.r.t to two agents
- all other agents only measure distances

Estimation of a Common Frame

$\hat{p}_{i,c}$ is estimate of $p_i - p_c$



measured distance

$$e(\hat{p}) = \frac{1}{4}\sum_{\{i,j\}\in\mathcal{E}}\left(\|\hat{p}_{j,c}-\hat{p}_{i,c}\|^2 - \ell_{ij}^2\right)^2 + \frac{1}{2}\|\hat{p}_{i_c,c}\|^2 + \frac{1}{2}\|\hat{p}_{\iota,c} - (p_\iota - p_{i_c})\|^2 + \frac{1}{2}\|\hat{p}_{\kappa,c} - (p_\kappa - p_{i_c})\|^2$$

measured by "special" agent

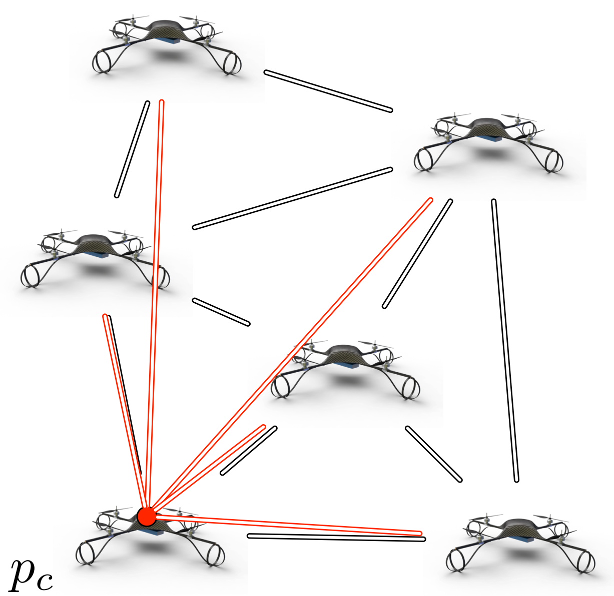
- "special" agent becomes the (moving) point each agent is trying to estimate the relative position of

Properties of error function

- non-negative and convex function
- **= 0** if and only if estimated distances equal measured distances

*based on approach of Calafiore *et al.*, 2010.

Estimation of a Common Frame



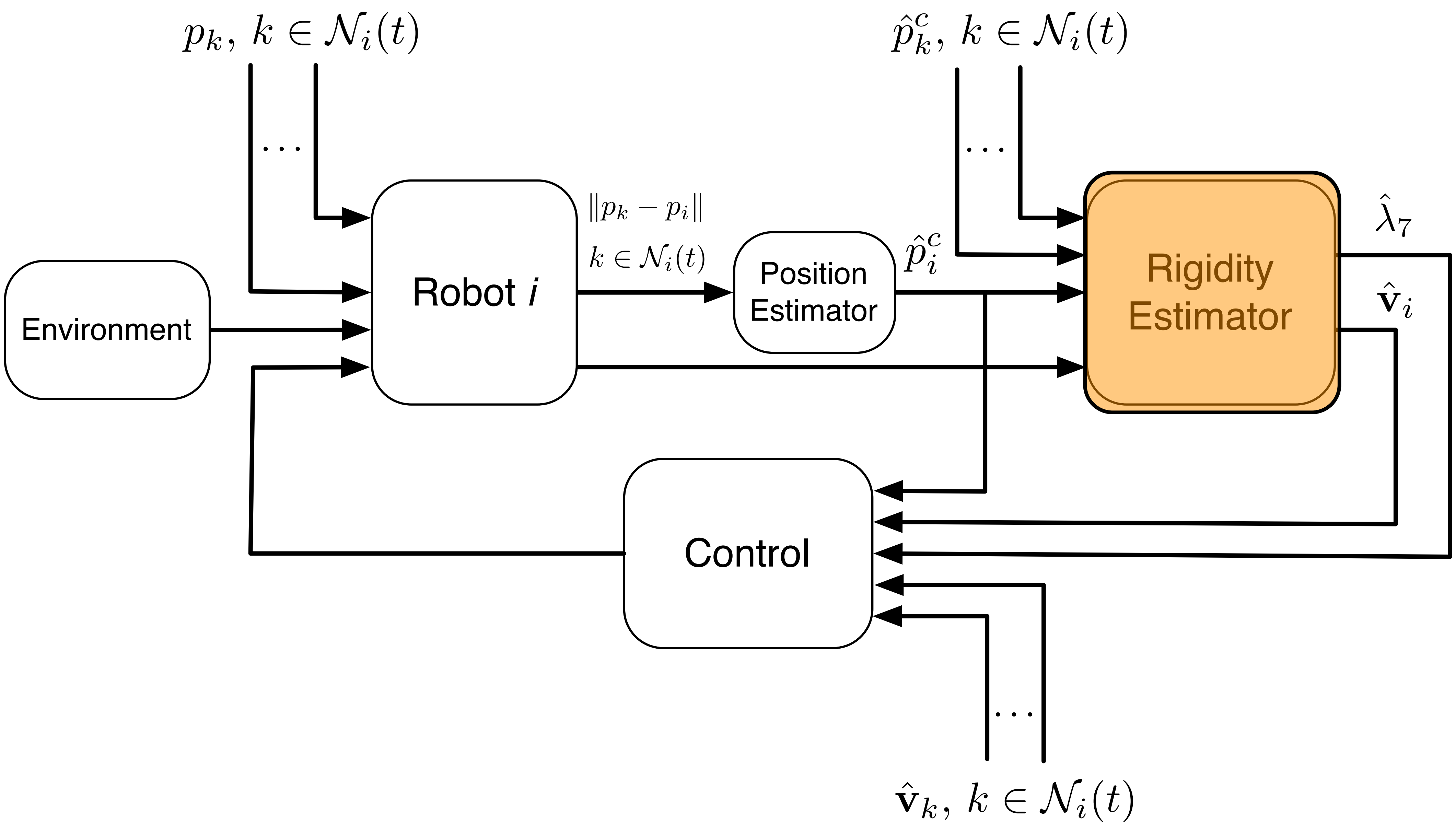
First-Order Gradient Descent

$$\dot{\hat{p}} = -\frac{\partial e}{\partial \hat{p}} = -\mathcal{R}(\hat{p})\hat{p} + R(\hat{p})\ell + \Delta^c$$

Proposition *If the framework is (infinitesimally) rigid then the vector of true values* $p - (\mathbb{1} \otimes p_c) = \begin{bmatrix} (p_1 - p_c)^T & \cdots & (p_{|\mathcal{V}|} - p_c)^T \end{bmatrix}^T$ *is an isolated local minimizer of* $e(\hat{p})$*. Therefore, there exists an* $\epsilon > 0$ *such that, for all initial conditions satisfying* $\|\hat{p}(0) - p - (\mathbb{1} \otimes p_c)\| < \epsilon$*, the estimation* $\hat{p}$ *converges to* $p - (\mathbb{1} \otimes p_c)$.

*proof based on Krick *et al.*, 2009

Estimation of Rigidity Eigenvalue



Estimation of Rigidity Eigenvalue

recall...

A framework is infinitesimally rigid if and only if the rigidity eigenvalue is positive

$\Rightarrow$

$$\xi_i = -\frac{\partial V_\lambda}{\partial \lambda_4}\left(\frac{\partial \lambda_4}{\partial p_i}\right)$$

requires all agents to have knowledge of rigidity eigenvalue and eigenvector

strategy

① **Power Iteration Method**

algorithm for estimating the *dominant eigenvalue* of a matrix

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}$$

② **Distributed Implementation**

use of dynamic consensus filters

Power Iteration Method

Rigidity eigenvalue is *not* the dominant eigenvalue of symmetric rigidity matrix

power iteration on "deflated" matrix

$$\tilde{\mathcal{R}} = I - TT^T - \alpha\mathcal{R}$$

$$\mathrm{IM}[T] = \mathrm{span}[\mathcal{N}(\mathcal{R})]$$

recall...

$$PRP^T = (I_2 \otimes E(\mathcal{G})) \begin{bmatrix} W_x & W_{xy} \\ W_{xy} & W_y \end{bmatrix} (I_2 \otimes E(\mathcal{G})^T)$$

$$\mathcal{N}(\mathcal{R}) = \mathrm{span}\left\{ \begin{bmatrix} \mathbb{1} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \mathbb{1} \end{bmatrix}, \begin{bmatrix} p^y - p_c^y \mathbb{1} \\ p_c^x \mathbb{1} - p^x \end{bmatrix} \right\}$$

relative position to a common point

Power Iteration Method

continuous-time centralized power iteration method

$$\dot{\hat{\mathbf{v}}}(t) = -\left(k_1 T T^T + k_2 \mathcal{R} + k_3 \left(\frac{\hat{\mathbf{v}}(t)^T \hat{\mathbf{v}}(t)}{3|\mathcal{V}|} - 1\right) I\right) \hat{\mathbf{v}}(t)$$

Theorem *Assume that the symmetric rigidity matrix* $\mathcal{R}$ *has distinct non-zero eigenvalues, and let* $\mathbf{v}$ *denote the rigidity eigenvector. Then for any initial condition* $\hat{\mathbf{v}}(t_0) \in \mathbb{R}^{3|\mathcal{V}|}$ *such that* $\mathbf{v}^T \hat{\mathbf{v}}(t_0) \neq 0$, *the trajectories of (17) converge to the subspace spanned by the rigidity eigenvector, i.e.,* $\lim_{t\to\infty} \hat{\mathbf{v}}(t) = \gamma \mathbf{v}$ *for* $\gamma \in \mathbb{R}$, *if and only if the gains* k_1, k_2 *and* k_3 *satisfy the following conditions:*

1) $k_1, k_2, k_3 > 0$,
2) $k_1 > k_2 \lambda_7$,
3) $k_3 > k_2 \lambda_7$.

*adapted from Yang *et al.*, 2010

Distributed Power Iteration

$$\dot{\hat{\mathbf{v}}}(t) = -\left(k_1 TT^T + k_2\mathcal{R} + k_3\left(\frac{\hat{\mathbf{v}}(t)^T\hat{\mathbf{v}}(t)}{3|\mathcal{V}|} - 1\right) I\right)\hat{\mathbf{v}}(t)$$

① symmetric rigidity matrix is a "naturally" distributed operator

$$P\mathcal{R}P^T = (I_2 \otimes E(\mathcal{G}))\begin{bmatrix} W_x & W_{xy} \\ W_{xy} & W_y \end{bmatrix}(I_2 \otimes E(\mathcal{G})^T)$$

② $$\left(\frac{\hat{\mathbf{v}}(t)^T\hat{\mathbf{v}}(t)}{3|\mathcal{V}|} - 1\right)\hat{\mathbf{v}}(t) = (Avg(\hat{\mathbf{v}}(t) \circ \hat{\mathbf{v}}(t)) - 1)\,\hat{\mathbf{v}}(t)$$

average of a vector can be distributedly computed using consensus algorithm*

PI-Consensus Filter [Freeman *et al.* 2006]

$$\begin{bmatrix} \dot{z}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} -\gamma I - K_P L(\mathcal{G}(t)) & K_I L(\mathcal{G}(t)) \\ -K_I L(\mathcal{G}(t)) & 0 \end{bmatrix}\begin{bmatrix} z(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} \gamma I \\ 0 \end{bmatrix} u(t)$$

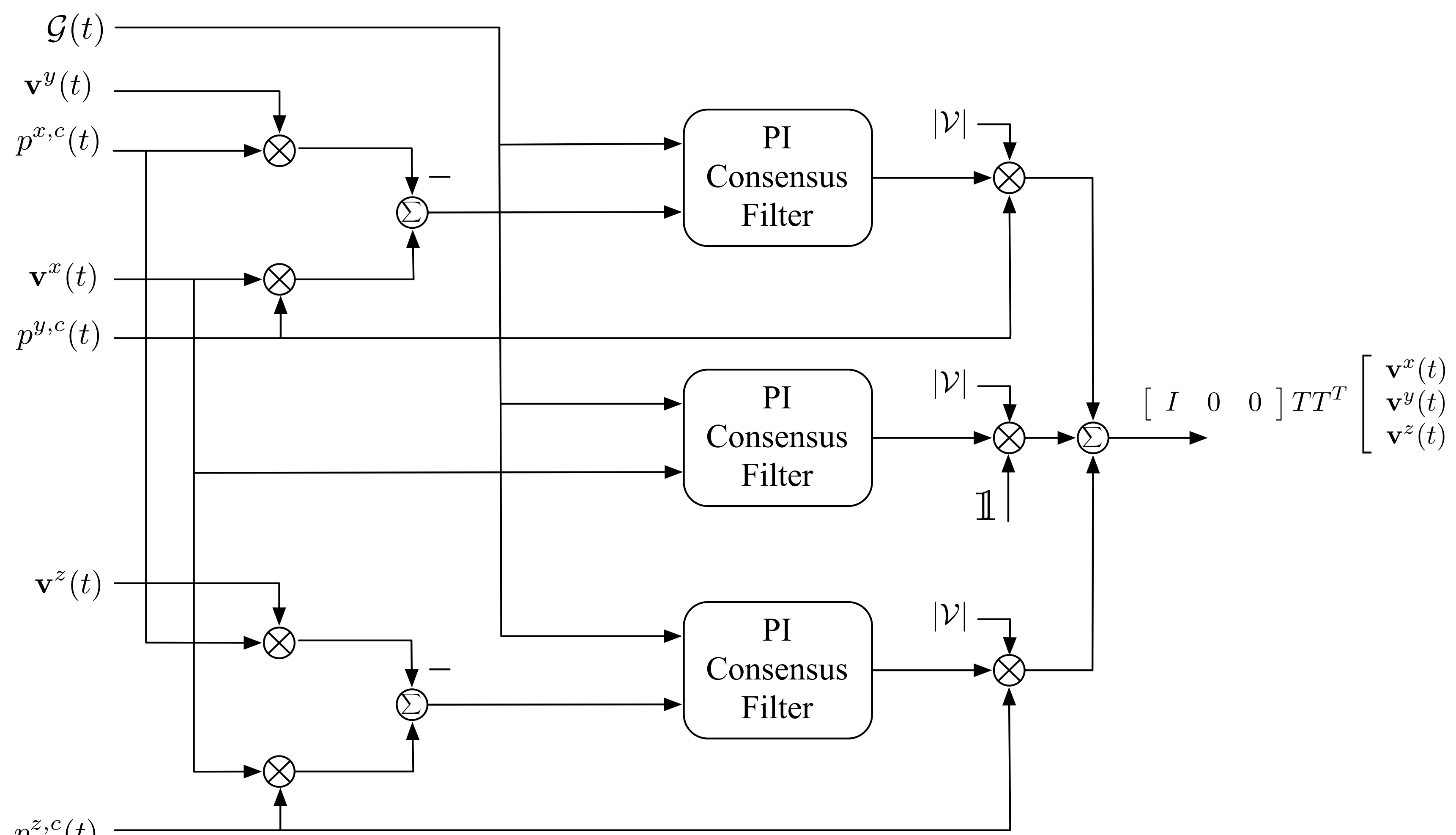
$$y(t) = \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} z(t) \\ w(t) \end{bmatrix}.$$

- dynamic consensus filter
- tunable gains
- tracks average of time-varying signal

Distributed Power Iteration

$$\dot{\hat{\mathbf{v}}}(t) = -\left(k_1 T T^T + k_2 \mathcal{R} + k_3 \left(\frac{\hat{\mathbf{v}}(t)^T \hat{\mathbf{v}}(t)}{3|\mathcal{V}|} - 1\right) I\right) \hat{\mathbf{v}}(t)$$

③ $$TT^T = \begin{bmatrix} \mathbb{1}\mathbb{1}^T + p^{y,c}(p^{y,c})^T + p^{z,c}(p^{z,c})^T & -p^{y,c}(p^{x,c})^T & -p^{z,c}(p^{x,c})^T \\ -p^{x,c}(p^{y,c})^T & \mathbb{1}\mathbb{1}^T + p^{x,c}(p^{x,c})^T + p^{z,c}(p^{z,c})^T & -p^{z,c}(p^{y,c})^T \\ -p^{x,c}(p^{z,c})^T & -p^{y,c}(p^{z,c})^T & \mathbb{1}\mathbb{1}^T + p^{x,c}(p^{x,c})^T + p^{y,c}(p^{y,c})^T \end{bmatrix}$$



Distributed Power Iteration

$$\dot{\hat{\mathbf{v}}}(t) = -\left(k_1 TT^T + k_2\mathcal{R} + k_3\left(\frac{\hat{\mathbf{v}}(t)^T\hat{\mathbf{v}}(t)}{3|\mathcal{V}|} - 1\right)I\right)\hat{\mathbf{v}}(t)$$

Corollary V.4. *Let* $\overline{\mathbf{v}}_i^2(t)$ *denote the output of the PI consensus filter for estimating the quantity* $Avg(\hat{\mathbf{v}}(t) \circ \hat{\mathbf{v}}(t))$ *for agent* i*. Then agent* i*'s estimate of the rigidity eigenvalue,* $\hat{\lambda}_7^i$*, can be obtained as*

$$\hat{\lambda}_7^i = \frac{k_3}{k_2}\left(1 - \overline{\mathbf{v}}_i^2(t)\right).$$

Putting it all together...

power iteration

$$\dot{\mathbf{v}}_i^x = -k_1|\mathcal{V}|\left(\overline{\mathbf{v}}_i^x + z_i^{xy}(t)\hat{p}_{i,c}^y + z_i^{xz}\hat{p}_{i,c}^z(t)\right) - k_2 \sum_{j\in\mathcal{N}_i(t)} W_{ij}\left(\hat{\mathbf{v}}_i^x(t) - \hat{\mathbf{v}}_j^x\right) - k_3\left(\overline{\mathbf{v}}_i^x - 1\right)\hat{\mathbf{v}}_i^x$$

frame estimation

$$\dot{\hat{p}}_{i,c} = \sum_{j\in\mathcal{N}_i(t)} \left(\|\hat{p}_{j,c}-\hat{p}_{i,c}\|^2 - \ell_{ij}^2\right)(\hat{p}_{j,c}-\hat{p}_{i,c}) - \delta_{ii_c}\hat{p}_{i,c} - \delta_{i\iota}\left(\hat{p}_{\iota,c} - (p_\iota - p_{i_c})\right) - \delta_{i\kappa}\left(\hat{p}_{\kappa,c} - (p_\kappa - p_{i_c})\right)$$

PI-consensus I

$$\begin{cases} \dot{\overline{\mathbf{v}}}_i^x = \gamma\left(\hat{\mathbf{v}}_i^x - \overline{\mathbf{v}}_i^x\right) - K_P \sum_{j\in\mathcal{N}_i} \left(\overline{\mathbf{v}}_i^x - \overline{\mathbf{v}}_j^x(t)\right) + K_I \sum_{j\in\mathcal{N}_i(t)} \left(\overline{w}_i^x - \overline{w}_j^x\right) \\ \dot{\overline{w}}_i^x = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(\overline{\mathbf{v}}_i^x - \overline{\mathbf{v}}_j^x\right) \end{cases}$$

PI-consensus II

$$\begin{cases} \dot{\overline{\mathbf{v}}}_i^{2x} = \gamma\left((\hat{\mathbf{v}}_i^x)^2 - \overline{\mathbf{v}}_i^{2x}\right) - K_P \sum_{j\in\mathcal{N}_i(t)} \left(\overline{\mathbf{v}}_i^{2x} - \overline{\mathbf{v}}_j^{2x}\right) + K_I \sum_{j\in\mathcal{N}_i(t)} \left(\overline{w}_i^{2x} - \overline{w}_j^{2x}\right) \\ \dot{\overline{w}}_i^{2x} = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(\overline{\mathbf{v}}_i^{2x} - \overline{\mathbf{v}}_j^{2x}\right) \end{cases}$$

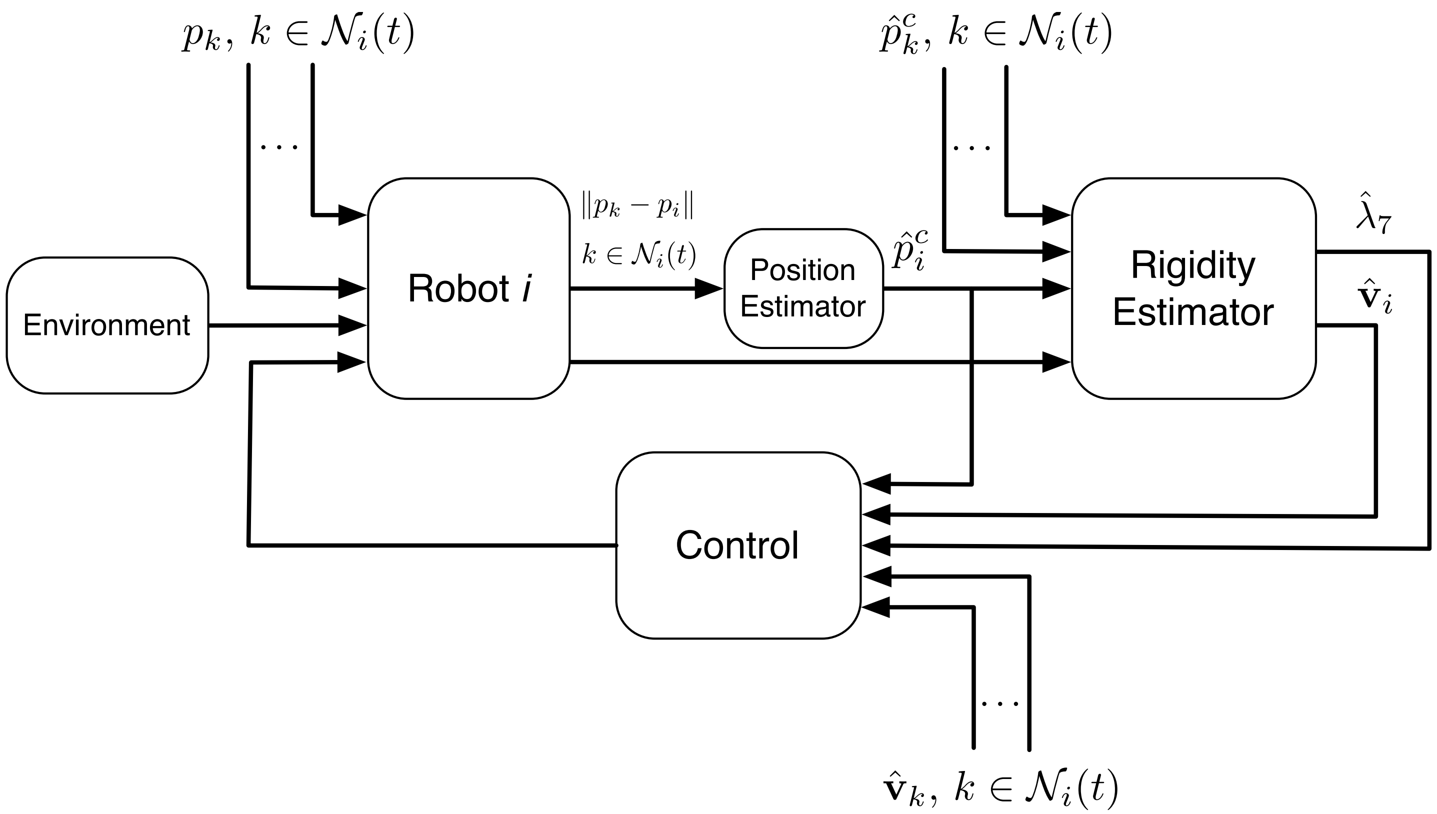
PI-consensus III

$$\begin{cases} \dot{z}_i^{xy} = \gamma\left((\hat{p}^y \circ \hat{\mathbf{v}}^x - \hat{p}^x \circ \hat{\mathbf{v}}^y) - z_i^{xy}\right) - K_P \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xy} - z_j^{xy}\right) + K_I \sum_{j\in\mathcal{N}_i(t)} \left(w_i^{xy}(t) - w_j^{xy}\right) \\ \dot{w}_i^{xy} = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xy} - z_j^{xy}\right) \end{cases}$$

PI-consensus IV

$$\begin{cases} \dot{z}_i^{xz} = \gamma\left((\hat{p}^z \circ \hat{\mathbf{v}}^x - \hat{p}^x \circ \hat{\mathbf{v}}^z) - z_i^{xz}\right) - K_P \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xy} - z_j^{xy}\right) + K_I \sum_{j\in\mathcal{N}_i(t)} \left(w_i^{xy} - w_j^{xy}\right) \\ \dot{w}_i^{xz} = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xz} - z_j^{xz}\right) \end{cases}$$

Rigidity Maintenance Controller



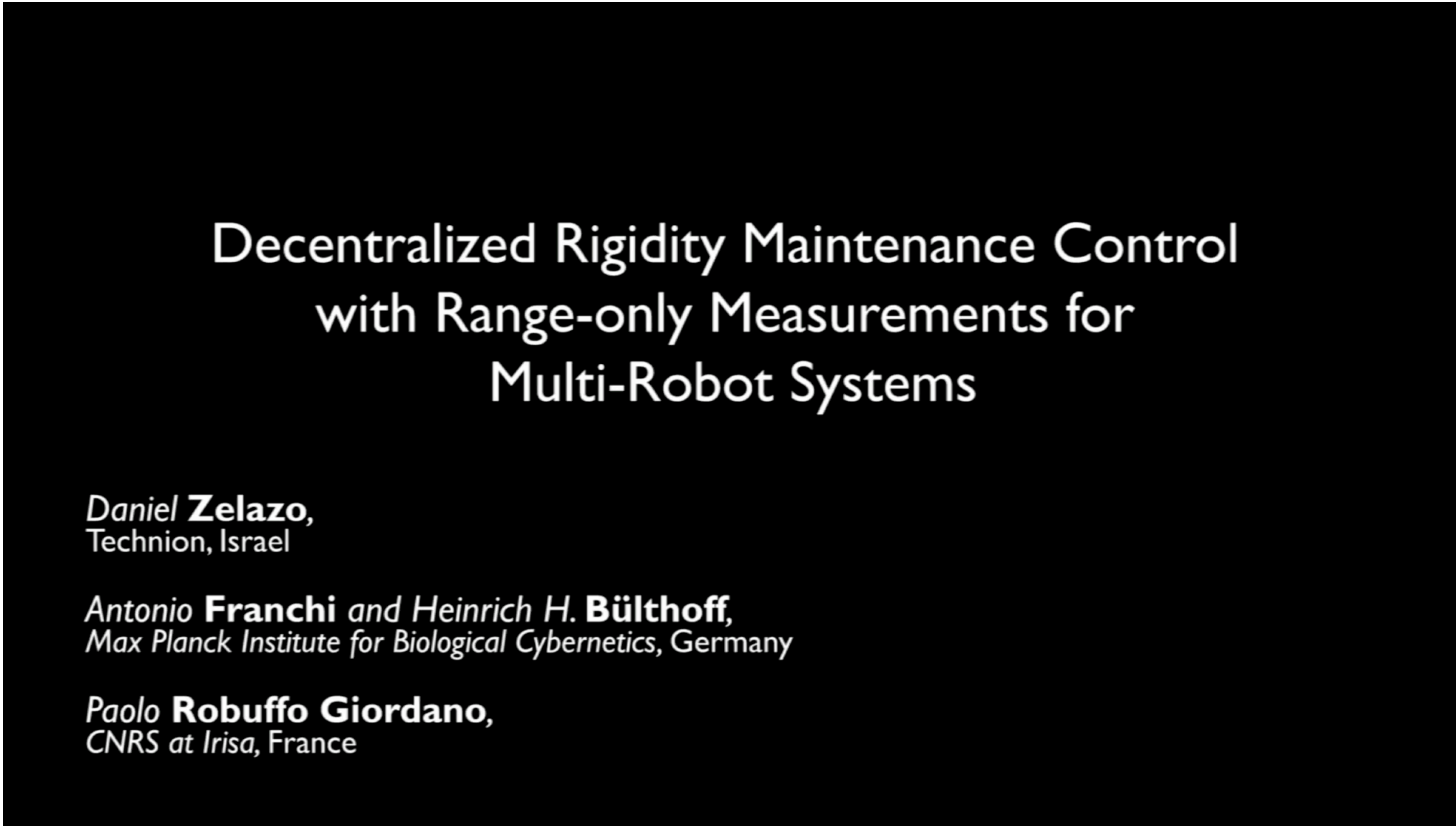
use output of rigidity estimator in control

$$\xi_i^x = -\frac{\partial V(\hat{\lambda}_7^i)}{\partial \lambda_7} \sum_{j \in \mathcal{N}_i} W_{ij} \Big(2(\hat{p}_{i,c}^x - \hat{p}_{j,c}^x)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)^2 +$$

$$2(\hat{p}_{i,c}^y - \hat{p}_{j,c}^y)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)(\hat{\mathbf{v}}_i^y - \hat{\mathbf{v}}_j^y) + 2(\hat{p}_{i,c}^z - \hat{p}_{j,c}^z)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)(\hat{\mathbf{v}}_i^z - \hat{\mathbf{v}}_j^z) \Big) +$$

$$\frac{\partial W_{ij}}{\partial p_i^x} \hat{S}_{ij},$$

Experiment

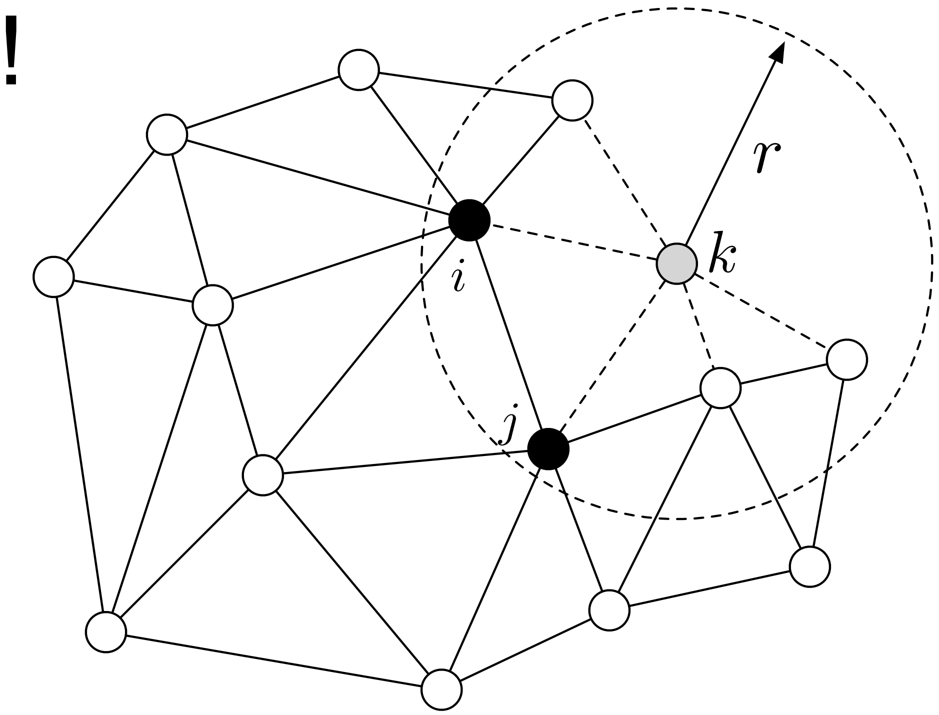

Decentralized Rigidity Maintenance Control
with Range-only Measurements for
Multi-Robot Systems
Daniel Zelazo,
Technion, Israel
Antonio Franchi and Heinrich H. Bülthoff,
Max Planck Institute for Biological Cybernetics, Germany
Paolo Robuffo Giordano,
CNRS at Irisa, France

Some unresolved points....

- Power Iteration method assumes *distinct eigenvalues*
  - proposed scheme can not guarantee that rigidity eigenvalue is unique
  - can lead to undesirable behaviors
- Formal stability proof for interconnection of all filters is missing
  - *ad hoc* implementation
  - *engineering art* to ensure each filter converges fast enough
  - alternative to power iteration method
- Need to relax requirement for "special agent"

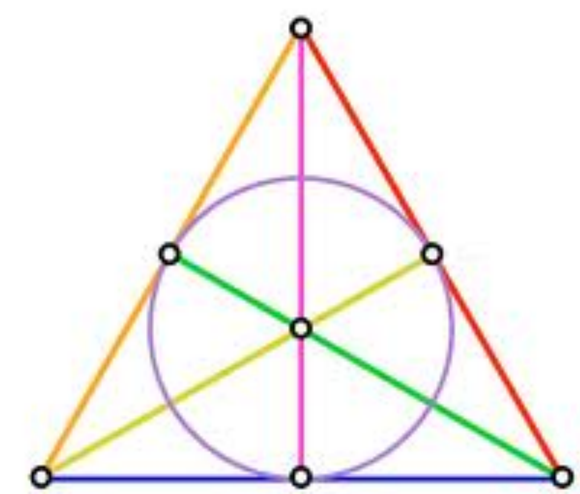
Outlook

Rigidity is an important architectural requirement for multi-agent systems!



- “bearing” rigidity
- full distributed implementations
- formation specification and trajectory tracking
- optimality
- rigidity matroids
- sub-modular optimization
- sensor fusion and localization
- ...

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$$



Acknowledgements

どもありがとうございます！

Dr. Paolo Robuffo Giordano

Dr. Antonio Franchi

Questions?