Multirobot Systems

The consensus problem and applications

The consensus problem

Master Program in Robotics, Graphics and Computer Vision
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

In this lecture

- The consensus problem
- Laplacian based solution: Consensus protocol in continuous-time
- The consensus protocol in discrete-time:
  - Solutions based on weight matrices (Metropolis, degrees..)
  - Solution based on the Perron Matrix
- An example (Metrolopolis weights)
- Applications and variations of the consensus problem
- A naïve formation control method

The consensus problem

- One of the most fundamental problem in multi-robots (and multi-agents) literature

- ***The consensus problem*: the goal and the rules**
  - Consider N robots with internal **state** $x_i \in \mathbb{R}$
  - Consider an internal **dynamics** for the state evolution. Here, single integrator:

$$\dot{x}_i = u_i$$

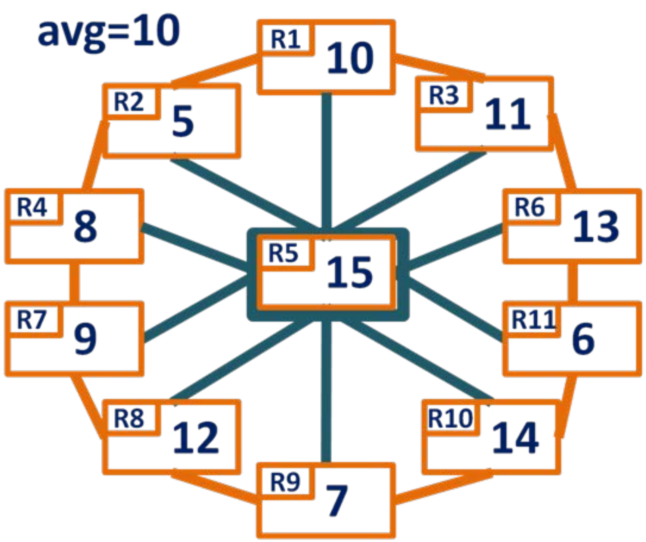
  - Consider an interaction **graph** between robots
  - Problem: design the control inputs u_i
  - so that all the sates **agree** on the same common value (unspecified, unknown, often the **average of x_i(0)**)

$$\lim_{t \to \infty} x_i(t) = \bar{x} \quad , \forall\, i = 1, \cdots, N$$

  - by making use of only information from **neighbors** (decentralized)

The consensus problem. Any ideas?

- Several possibilities, some of them very intuitive

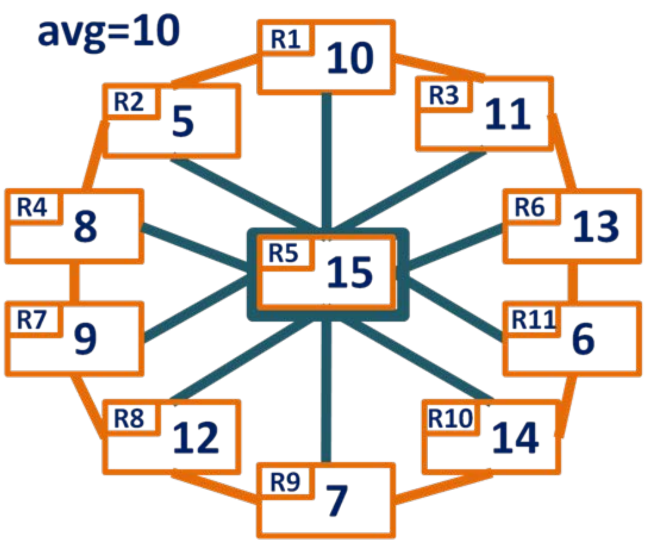


- Computational / storage / communication costs? (per iteration)
- Time until a robot gets the **average** value?
- What if the graph changes along time?
- Key idea of the consensus protocol (next): **distributed**, **scalable**

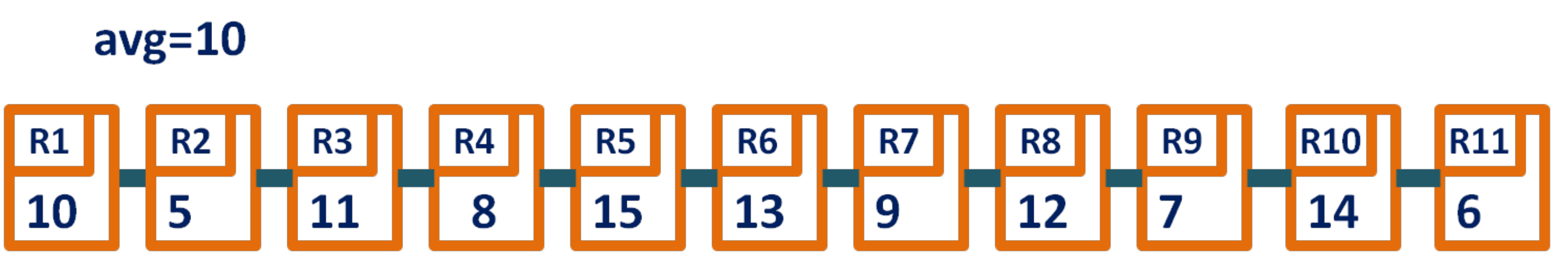
Images: created by the lecturers of the course

The consensus problem. Any ideas?

- Several possibilities, some of them very intuitive



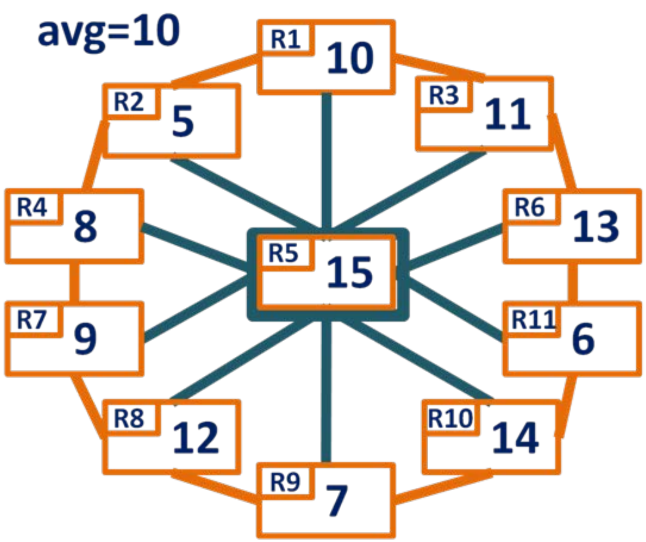
- R5 is the leader or root, compiling all info, making computation, sending the value to all the nodes
- Build trees, perform partial computations
- Only for fixed graphs. Switching graphs?



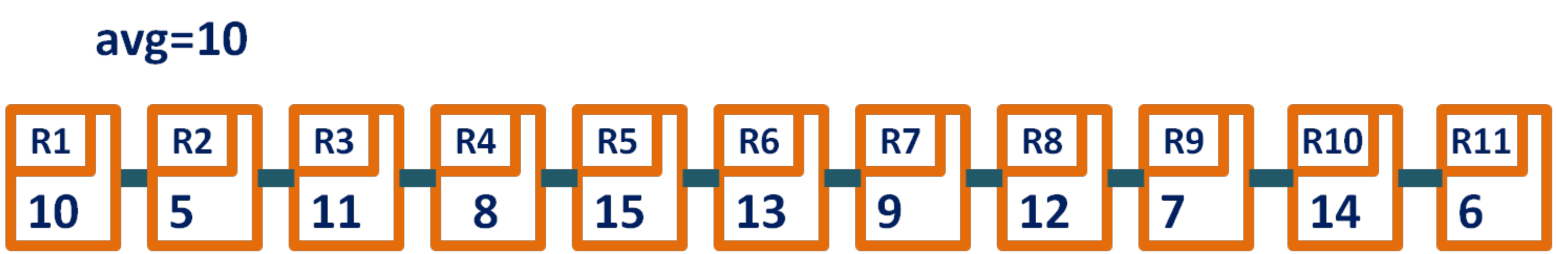
Images: created by the lecturers of the course

The consensus problem. Any ideas?

- Several possibilities, some of them very intuitive



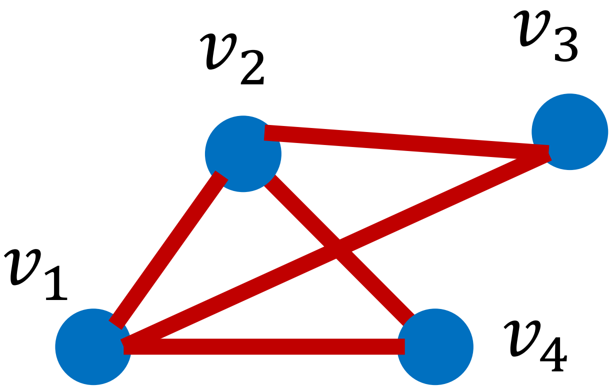
- Switching graphs?
- Keep a local storage of all the values discovered so far
- When meeting a node: compile the unknown values
- Costs depend on N
- The consensus problem: solutions with constant memory costs! **Scalability**



Images: created by the lecturers of the course

Several solutions. The most popular ones:

- **Laplacian** based: Let the control input $\mathbf{u_i}$ be the sum of all the differences of the neighbors states relative to the sate of the agent



$\dot{x}_i = u_i$ with:

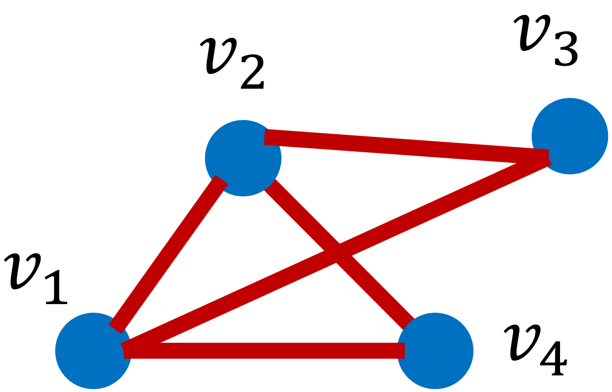
$$u_1 = (x_2 - x_1) + (x_3 - x_1) + (x_4 - x_1)$$
$$u_2 = (x_1 - x_2) + (x_3 - x_2) + (x_4 - x_2)$$
$$u_3 = (x_2 - x_3) + (x_1 - x_3)$$
$$u_4 = (x_2 - x_4) + (x_1 - x_4)$$

$$\dot{x}_i = u_i = \sum_{j \in N_i} (x_j - x_i)$$

Images: created by the lecturers of the course

Several solutions. The most popular ones:

- **Laplacian** based: Let the control input $\mathbf{u_i}$ be the sum of all the differences of the neighbors states relative to the sate of the agent



$\dot{x}_i = u_i$ with:

$$\dot{x}_i = u_i = \sum_{j \in N_i} (x_j - x_i)$$

- In compact form:

$$\dot{\boldsymbol{x}} = \boldsymbol{u} = -L\boldsymbol{x}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{u} = -\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix} \boldsymbol{x}$$

- Results for **undirected** graphs: Asymptotic convergence to the **average** of the initial robot states if the graph is **connected**

Images: created by the lecturers of the course

Several solutions. The most popular ones:

- **The consensus protocol in discrete time**: Iteratively, each robot:

$$x_i\ (k+1) = W_{ii}(k)x_i(k) + \sum_{j \in N_i(k)} W_{ij}(k)x_j(k)$$

- (Metropolis weights:)

$$W_{ij}(k) = \begin{cases} \dfrac{1}{1+max\{d_i(k), d_j(k)\}}, if\ (v_i, v_j) \in E(k) \\ 1 - \sum_{j' \in N_i(k)} W_{ij'}(k)\ , if\ i = j \\ 0, otherwise \end{cases}$$

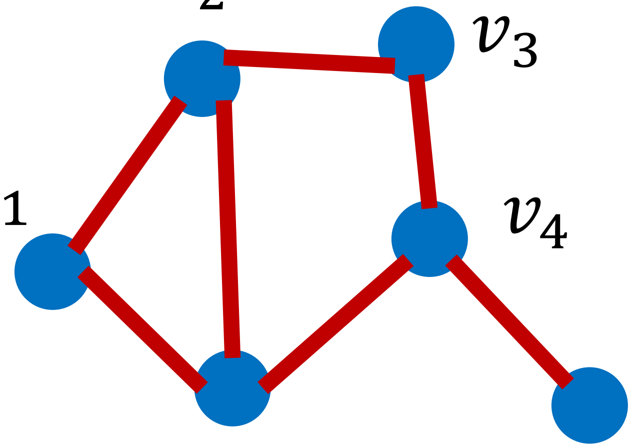
- (Laplacian -> Perron matrix:)

$$x_i\ (k+1) = x_{ii}(k) + \alpha \sum_{j \in N_i(k)} \left(x_j(k) - x_i(k)\right)$$

with α positive 0 < α < 1/(2N)

- Results for **undirected** graphs: Asymptotic convergence to the **average** of the initial robot states if the graph is **connected**

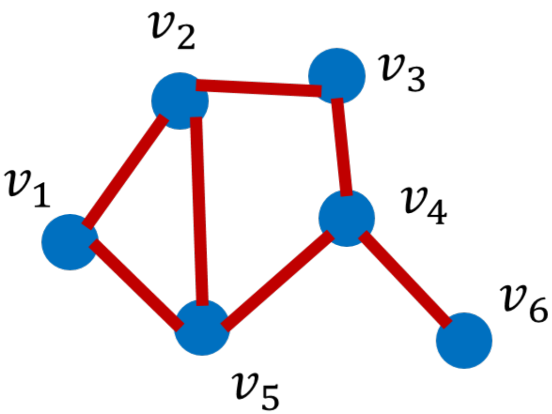
An example: Metropolis weights



$$W_{ij}(k) = \begin{cases} \frac{1}{1 + max\{d_i(k), d_j(k)\}}, if\ (v_i, v_j) \in E(k) \\ 1 - \sum_{j' \in N_i(k)} W_{ij'}(k)\ , if\ i = j \\ 0, otherwise \end{cases}$$

Robot	Initial state	Neighbors	Degre e	Weights (Metropolis)
i=1	x1(0)=5	N1={2,5}	d1=2	W12=1/4, W15=1/4, W11=0.5
i=2	x2(0)=20	N2={1,3,5}	d2=3	W21=1/4, W23=1/4, W25=1/4, W22=0.25
i=3	x3(0)=12	N3={2,4}	d3=2	W32=1/4, W34=1/4, W33= 0.5
i=4	x4(0)=2	N4={3,5,6}	d4=3	W43=1/4, W45=1/4, W46=1/4, W44=0.25
…	…	…	…	…

An example: Metropolis weights



Consensus algorithm run at every iteration by the robots (using the Metropolis weights)

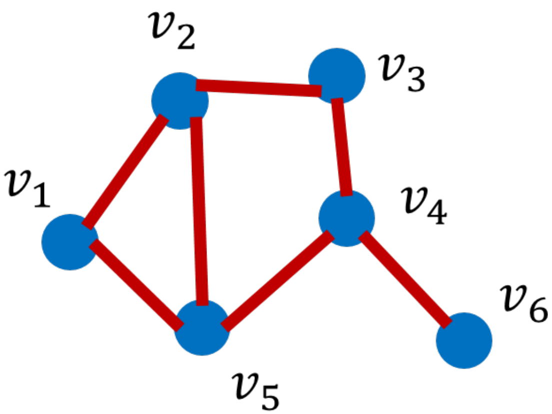
$$x_i\ (k+1) = W_{ii}(k)x_i(k) + \sum_{j \in N_i(k)} W_{ij}(k)x_j(k)$$

Robot i=1 (step t)
Send x1(t) to neighbors N1={2,5}
Receive x2(t) and x5(t) from neighbors
Update
x1(t+1)=0.5 * x1(t) + 0.25*x2(t) + 0.25*x5(t)

Robot i=6 (step t)
Send x6(t) to neighbor N6={4}
Receive x4(t) from neighbor
Update
x6(t+1)=0.75 * x6(t) + 0.25*x4(t)

Robot	Initial state	Neighbors	Degree	Weights (Metropolis)
i=1	x1(0)=5	N1={2,5}	d1=2	W12=1/4, W15=1/4, W11=0.5
i=6	X6(0)=22	N6={4}	d6=1	W64=1/4, W66=0.75

An example: Metropolis weights

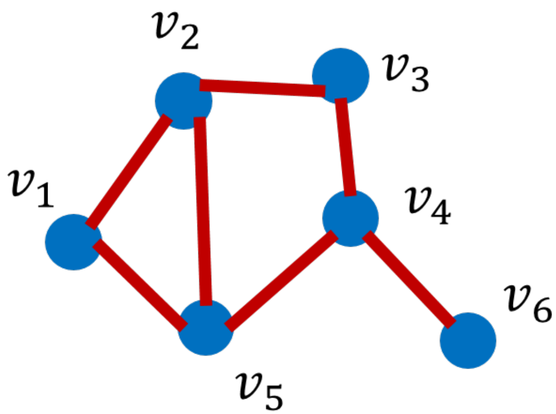


As more iterations are run …

$$x_i\ (k+1) = W_{ii}(k)x_i(k) + \sum_{j \in N_i(k)} W_{ij}(k)x_j(k)$$

Robot	State (t=0)	t=1	t=2	t=3	t=4	t=5	t=6	…	t=10
i=1	x1(0)=5	8.25	8.5	8.8	9.1	9.4	9.6	…	10.2
i=2	x2(0)=20	10	9.3	9.3	9.5	9.7	9.9		10.3
i=3	x3(0)=12	11.5	10.7	10.5	10.5	10.5	10.5		10.6
i=4	x4(0)=2	9.75	11.4	11.5	11.5	11.3	11.2		10.9
i=5	x5(0)=3	7.5	8.9	9.5	9.8	10	10.1		10.4
i=6	X6(0)=22	17	15.2	14.2	13.6	13	12.6	…	11.5
avg(t)	10.7	10.7	10.7	10.7	10.7	10.7	10.7	…	10.7

An example: Metropolis weights



As more iterations are run …

$$x_i\ (k+1) = W_{ii}(k)x_i(k) + \sum_{j \in N_i(k)} W_{ij}(k)x_j(k)$$

Robot	State (t=0)	t=1	…	t=10
i=1	x1(0)=5	8.25	…	10.2
i=2	x2(0)=20	10		10.3
i=3	x3(0)=12	11.5		10.6
i=4	x4(0)=2	9.75		10.9
i=5	x5(0)=3	7.5		10.4
i=6	X6(0)=22	17	…	11.5
avg(t)	10.7	10.7	…	10.7

Consensus vs. flooding (tree building + propagation)

Memory storage required?
n increases and.. ?
Switching topology?

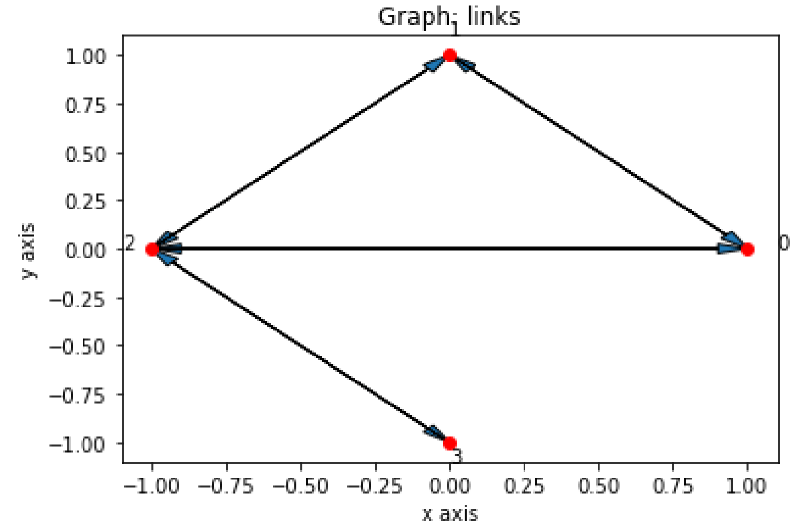
An example: Simulating consensus with matrices

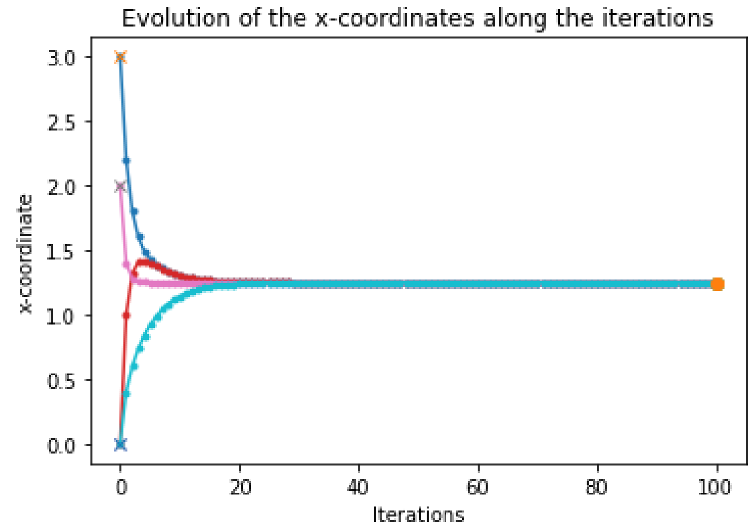
- **The compact form**: Based on matrices
- Allows a fast check of how the states of **all** the robots will evolve
- Implementation (Laplacian & Perron matrix method:)
  - Define the Graph (nodes and edges)
  - Compute the Adjacency matrix A
  - Compute the Degree matrix D
  - Compute the Laplacian matrix

    $L = D - A$
  - Compute the Perron matrix:

    $W = I - \alpha L$
  - Select the initial states vector x
  - Iteratively,

    $x = W x$
  - Store and plot the results





Images: created by the lecturers of the course

Why is the consensus problem interesting?

Rendezvous (consensus-based)

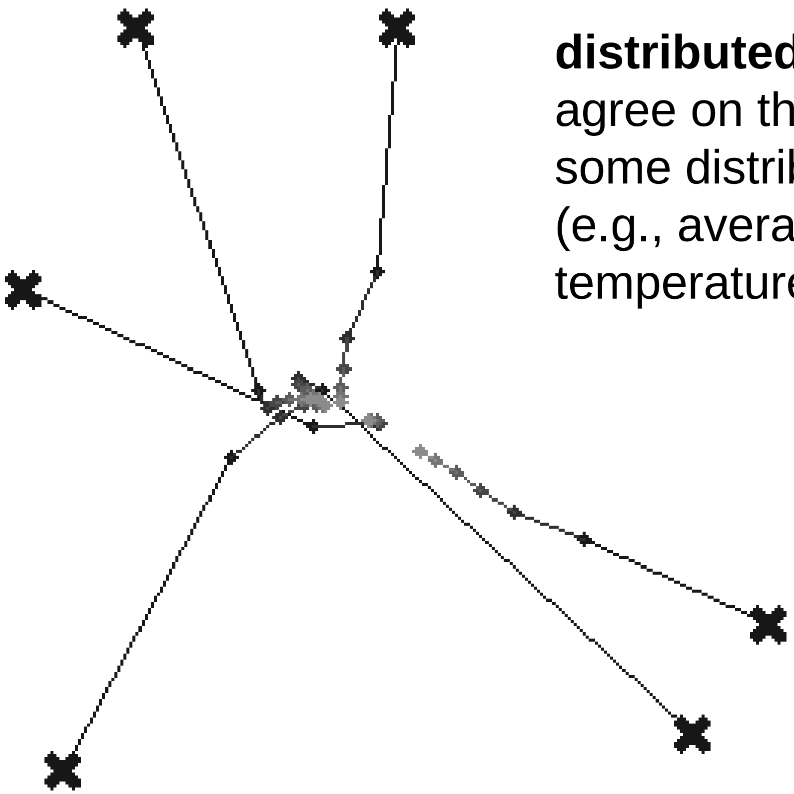
meet at a common point (uniform the positions)

Average on x-coordinate
Average on y-coordinate
Robots move to position

$$\left(x_i\ (k+1), y_i\ (k+1)\right)$$

Rendezvous at the centroid

¿One leader?
Robots follow the leader



distributed estimation: agree on the estimation of some distributed quantity (e.g., average temperature)

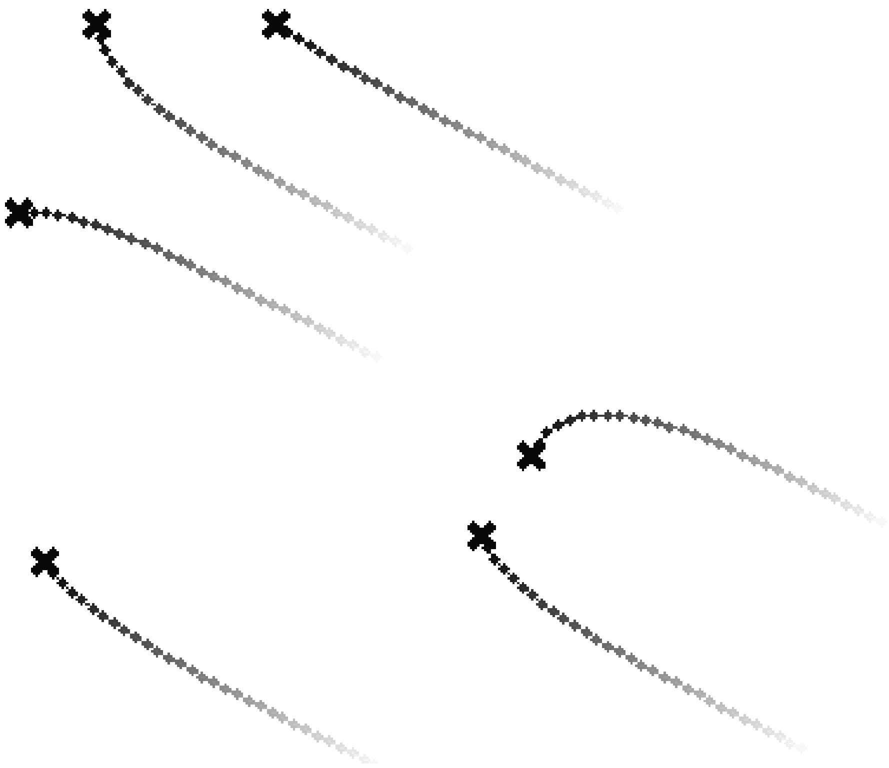
Images: created by the lecturers of the course

Why is the consensus problem interesting?

Images: created by the lecturers of the course

Flocking (consensus-based)

alignment: point in the same direction (uniform the angles)



$$x_i \ (k+1) = x_i \ (k) + v_i \ (k) \, T$$

Speed with constant modulus and orientation given by the **averaged orientation**

Alternative: average on speed modulus and on orientation

¿One leader?

Robots flock accordingly

T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995

Why is the consensus problem interesting?

Circuit pursuit

Orbit motions

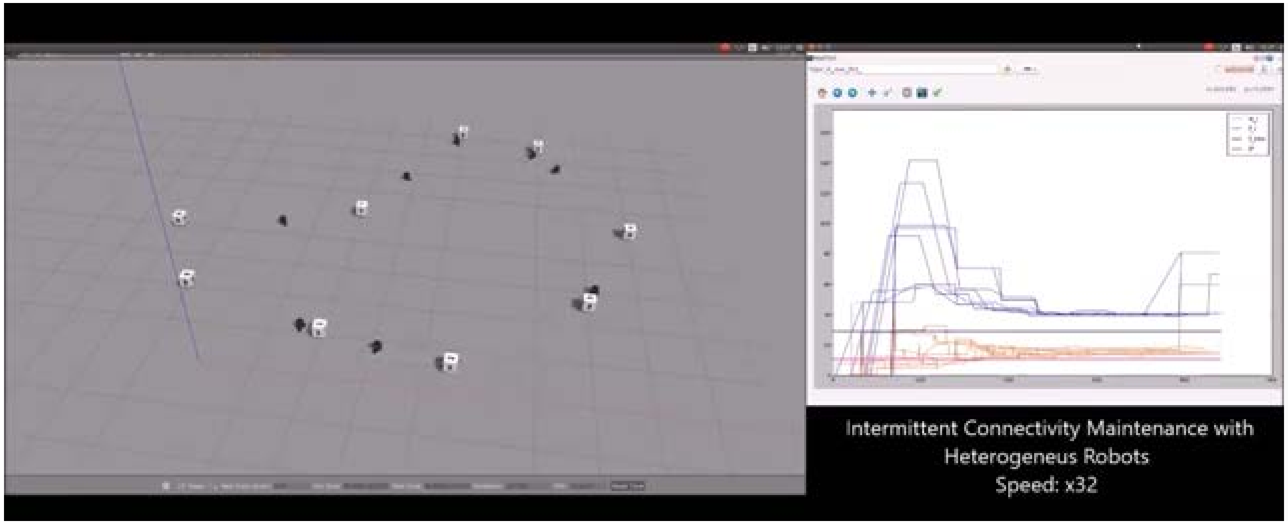
Deployment on a ring

Target enclosing

Intermittent connectivity

https://www.youtube.com/playlist?list=PLmyvo-kjDwz30b_i8vW6gW0NeC3NHBvo6

Containment control



Images: created by the lecturers of the course

R. Aragues, D. V. Dimarogonas (2019). *Intermittent Connectivity Maintenance with Heterogeneous Robots using a Beads–on–a–Ring Strategy*. American Control Conference (ACC), 2019, Philadelphia, PA, USA, pp. 120–126 + Extension Pablo Guallar & C. Sagues

Mei, J., Ren, W., & Ma, G. (2012). Distributed containment control for Lagrangian networks with parametric uncertainties under a directed graph. *Automatica*, *48*(4), 653-659.

Why is the consensus problem interesting?

Formation control

In this part of the course: linear (naïve) consensus-based version

https://youtu.be/AxT-fFcGQoA

Cooperative transport

https://www.youtube.com/watch?v=kxRu426UVdM

Drones in formation going through a narrow passage

https://www.youtube.com/watch?v=YQIMGV5vtd4&t

K. Fathian, S. Safaoui, T. H. Summers and N. R. Gans, "Robust Distributed Planar Formation Control for Higher Order Holonomic and Nonholonomic Agents," in IEEE Transactions on Robotics, doi: 10.1109/TRO.2020.3014022.

Alonso-Mora, J, Knepper, R, Siegwart, R, & Rus, D (2015). Local motion planning for collaborative multi-robot manipulation of deformable objects. *IEEE int. Conf. robotics automation,* pp. 5495-5502.

Formation control (Consensus-based)

Kaveh Fathian, Sleiman Safaoui, Tyler Summers, Nicholas Gans
University of Texas at Dallas

$$x_i\ (k+1) = W_{ii}(k)x_i(k) + \sum_{j\in N_i(k)} W_{ij}(k)x_j(k)$$

Rewritten

$$\boldsymbol{x_i\ (k+1) = x_i\ (k) + \sum_{j\in N_i} W_{ij}(x_j(k) - x_i(k))}$$

Now… to keep a fixed relative position between neighbors rij

$$\boldsymbol{x_i\ (k+1) = x_i\ (k) + \sum_{j\in N_i} W_{ij}(x_j(k) - x_i(k) - r_{ij})}$$

why it works? x_i(k) remains constant only if the desired relative positions are kept

Equivalently..

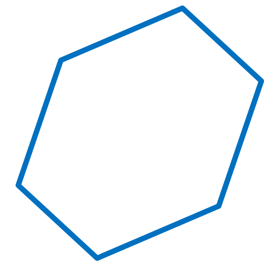
$$\boldsymbol{x_i\ (k+1) = x_i\ (k) + \sum_{j\in N_i} W_{ij}(x_j(k) - x_i(k)) + r_i} \qquad \boldsymbol{r_i = -\sum_{j\in N_i} W_{ij} r_{ij}}$$

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215-233.

Formation control (Consensus-based). STEPS

Kaveh Fathian, Sleiman Safaoui, Tyler Summers, Nicholas Gans
University of Texas at Dallas

To choose a geometric pattern and assign robot identifiers



To choose a network topology for the robots

To compute the desired relative positions between neighbors $\boldsymbol{r_{ij}}$

To obtain the compact version $$\boldsymbol{r_i} = -\sum_{j \in N_i} \boldsymbol{W_{ij} r_{ij}}$$

To run the formation control iteration at every robot

$$\boldsymbol{x_i}\ (\boldsymbol{k}+\mathbf{1}) = \boldsymbol{x_i}\ (\boldsymbol{k}) + \sum_{j \in N_i} \boldsymbol{W_{ij}}(\boldsymbol{x_j}(\boldsymbol{k}) - \boldsymbol{x_i}(\boldsymbol{k}))\ + \boldsymbol{r_i}$$

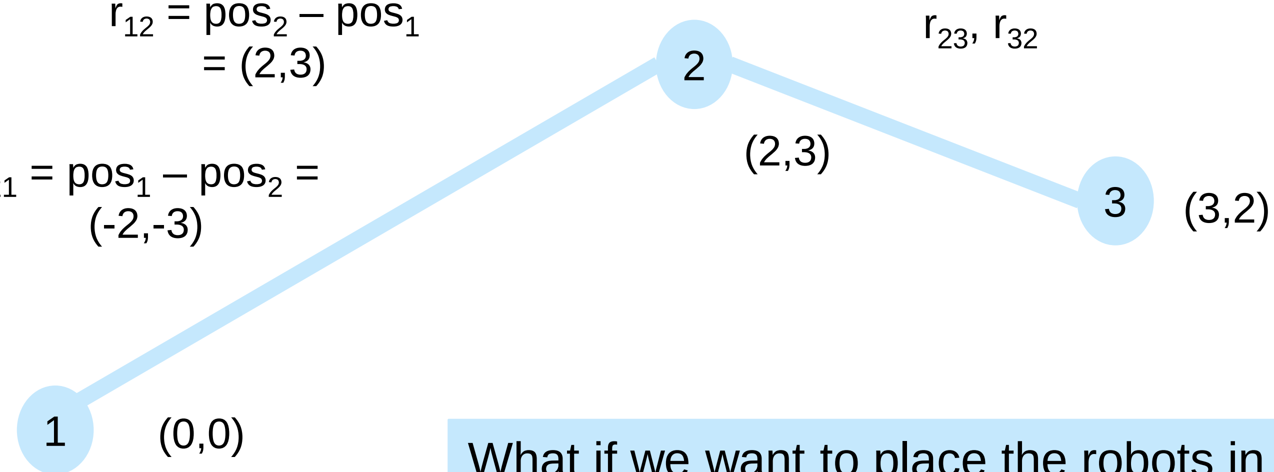
Kaveh Fathian, Sleiman Safaoui, Tyler Summers, Nicholas Gans
University of Texas at Dallas

Example: if the desired formation is as follows, then the desired relative positions would be:

$r_{12} = pos_2 - pos_1 = (2,3)$

$r_{21} = pos_1 - pos_2 = (-2,-3)$

r_{23}, r_{32}



What if we want to place the robots in a **vertical line** formation?
To make things easier, you can assume the robot identifiers are sorted in the line as well

In which sense this formation control approach is naïve?

- Relative measurements between i and j
  - sensors? assumptions?
- Local / global coordinate methods
- Range only / bearing only
- Sensing vs. Communication (undirected / directed graphs)
- What if the network depends on the distance between agents?
- Network connectivity imposition / multi-hop messages / combine with rendezvous / exchange goals
- Are all motions attainable? (omnidirectional / differential drive …)
- Collision avoidance

Some of these problems will be revisited later in the course

Cortés, J., & Egerstedt, M. (2017). Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, *10*(6), 495-503.

Main ideas in this lecture ?

- The consensus problem
- Consensus protocols in discrete time
- Applications: Rendezvous, flocking…
- A formation-control method
- Algorithm: What every robot runs
  - From a global point of view

    (fast simulations, check of properties)

Ok, but how does it work? What does it mean that the convergence is asymptotic? And if there are more robots, more links?

Next lectures…

- Implementation (compact, matrix form) of the consensus protocol
  - Experience tuning the parameters, including more or less links, establishing leaders
  - Obtain figures with the evolution of the robot states

- Advanced topics related to the consensus problem
  - Why it works: sketches of the proofs
  - Dynamic consensus
  - Gossip consensus

Bibliography

- Xiao, L., Boyd, S., & Lall, S. (2005, April). A scheme for robust distributed sensor fusion based on average consensus. In IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005. (pp. 63-70). IEEE.
- Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and Cooperation in Networked Multi-Agent Systems. Proceedings of the IEEE (95)1:215-233, 2007.
- Kia, S. S., Van Scoy, B., Cortes, J., Freeman, R. A., Lynch, K. M., & Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. IEEE Control Systems Magazine, 39(3), 40-72.

- Courses in other institutions covering similar topics:
  - "Control of Autonomous Multi-Agent Systems II", Dr. Antonio Franchi and Prof. Giuseppe Oriolo. Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma. http://www.diag.uniroma1.it/oriolo/cams_part2/
  - "Mobile Robot Systems", Dr. Amanda Prorok. University of Cambridge, Dep. Of Computer Science and Technology. https://www.cl.cam.ac.uk/teaching/1819/MobRobot/