EPFL

Disal

# Distributed Intelligent Systems – W6:

# Collective Movements in Multi-Robot Systems

# Outline

- Flocking for Multi-Robot Systems
  - Differences between digital and physical world
  - Examples

- Formations for Multi-Robot Systems
  - Behavior-based control strategies
  - Graph-based (consensus-based) control strategies
  - Examples

# Applications of Flocking/Formation

In some applications such as:

- lawn-mowing,
- vacuum cleaning,
- security patrolling,
- coverage and mapping,
- search and exploration in hazardous environment, etc.

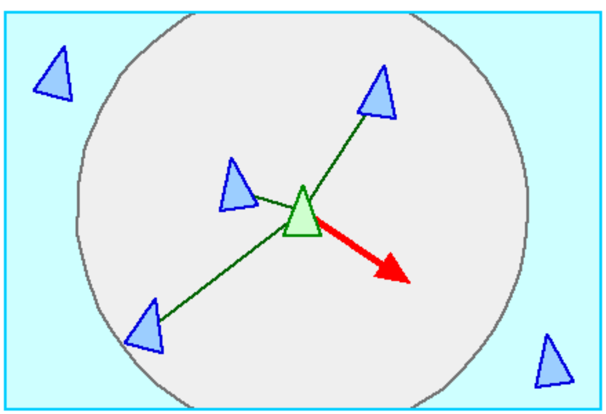
it is desired that the robots to stay together while navigating in the environment as a group.

# 2D Flocking in Real Robots

# From W5: Boids' Flocking Rules

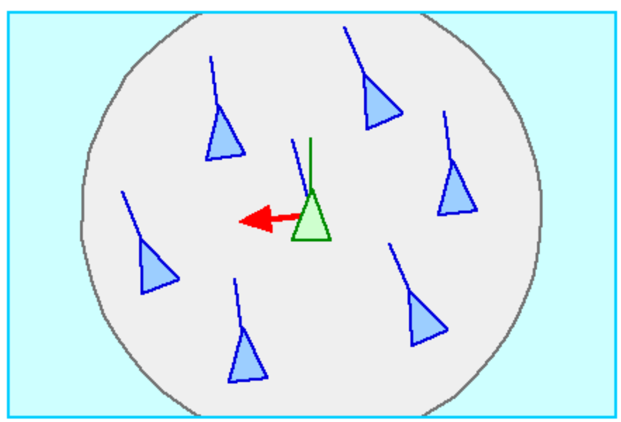
1. Separation: avoid collisions with nearby flockmates
2. Alignment: attempt to match velocity (speed and direction) with nearby flockmates
3. Cohesion: attempt to stay close to nearby flockmates

**Position control**



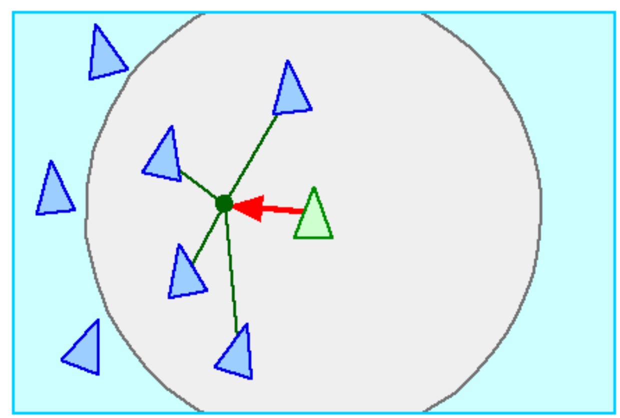
separation

**Velocity control**



alignment

**Position control**

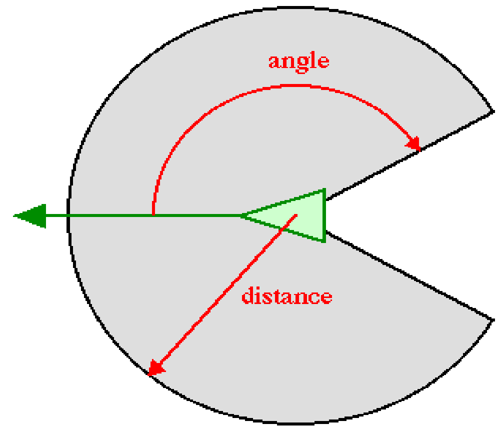


cohesion

# From W5: Boids' Sensory System

**An idealized system (but distributed and local!):**

- Local, almost omni-directional sensory system
- Perfect relative range and bearing system: no occlusion, no noise, all teammates perfectly identified within the range of detection
- Immediate response: one perception-to-action loop (no sensory, computational capacity considered)
- Homogeneous system (all boids have exactly the same sensory system)
- "Natural" nonlinearities: negative exponential of the distance (linear response also tested: bouncy, cartoony)




Neighborhood (2D version)

# A Real On-Board Sensory System for Flocking

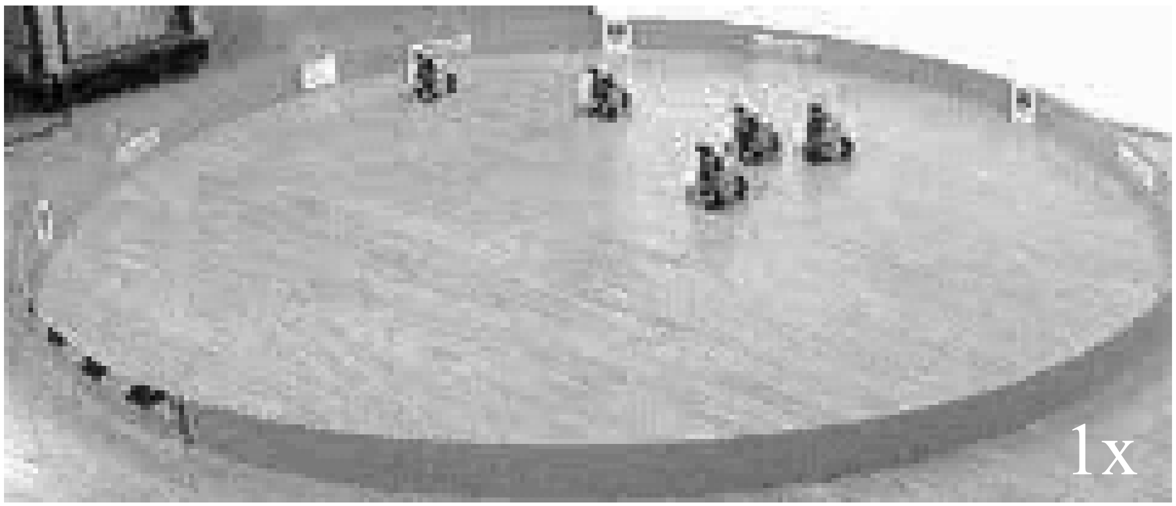
**In general, for real robots:**

- Noise in the range and bearing measurement, communication
- Homogeneous system impossible: even from manufacturing point of view small discrepancies -> calibration might be the key for an efficient system
- Immediate response impossible: computational and sensory capacity limited!
- Identifier for each teammate possible but scalability issues
- Non holonomicity of the vehicles

**More specifically, for local range and bearing systems:**

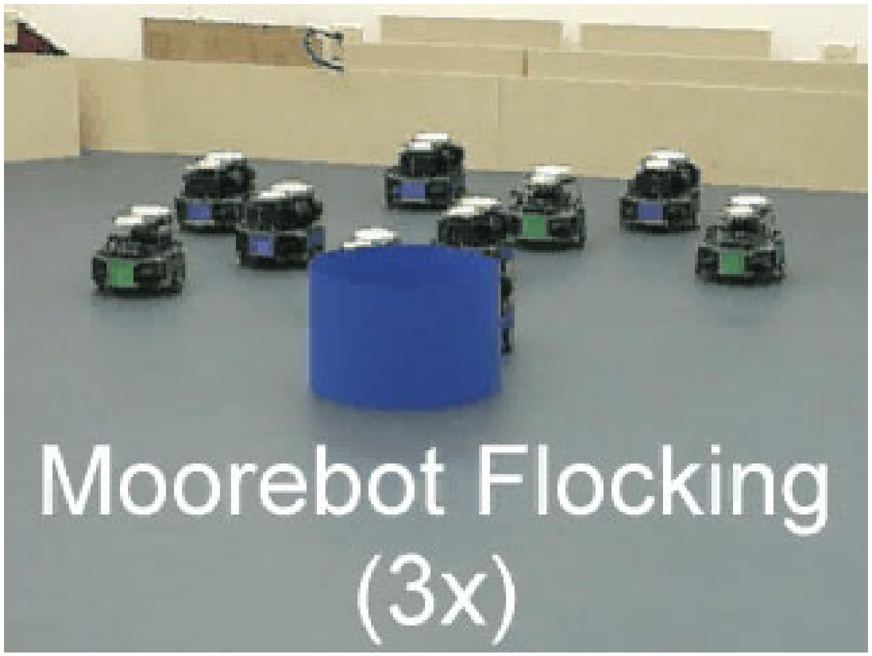
- Depending on the system used for range and bearing: occlusion possible (line of sight)!
- Nonlinearities determined by the underlying technology: might need to compensate with control for obtaining the desired effect!
- Second order variables (velocity) estimated with 2 first order measures (position) but takes time (the noisier the signal the more filtering needed, the longer the time)!

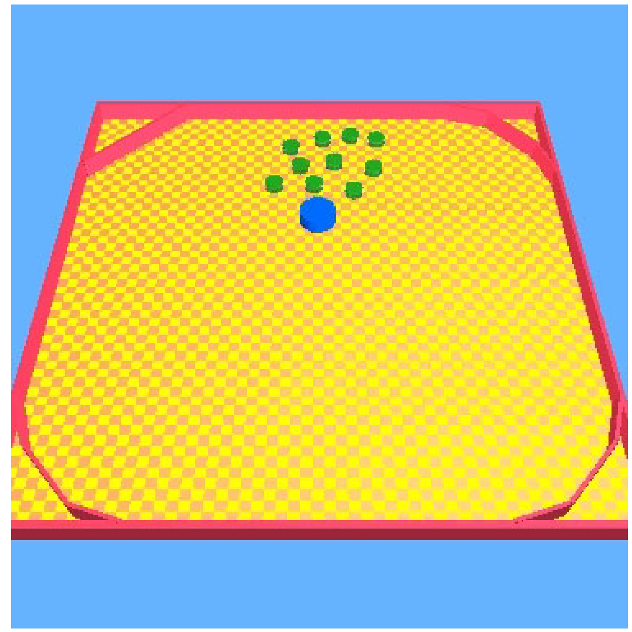
# Ex. 1: Kelly's Flocking (1996)




- Separation and cohesion only (alignment not applied)
- Migration urge/script replaced by leadership
- All on-board (IR system for local communication, range and bearing, fast 10 Hz)

# Ex. 2: Hayes's Flocking (2002)



- Separation, cohesion, and alignment
- Range & bearing using off-board system (overhead camera and LAN radio channel)

# 2D Formations – Behavior-Based Control

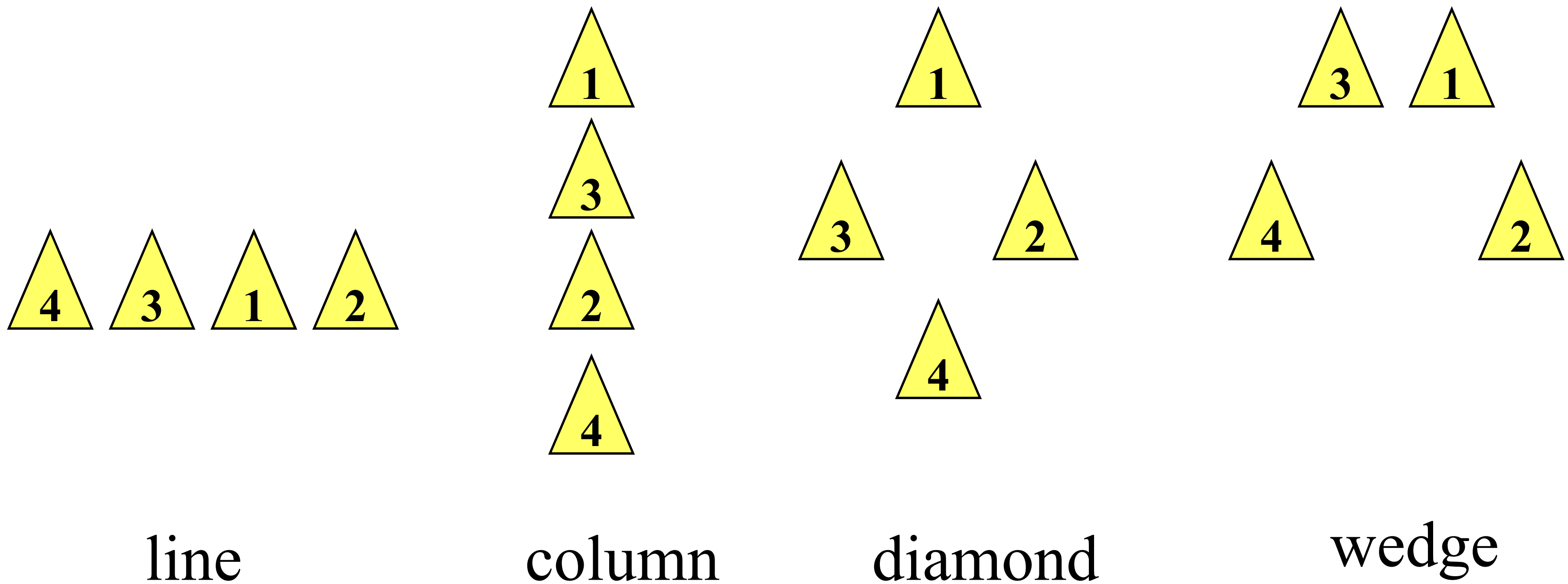
# Balch & Arkin, 1998

- Absolute coordinate system assumed (GPS, dead reckoning) but positional error considered
- Fully networked system but transmission delays considered (and formation traveling speed adapted …)
- Different platforms (lab robots, UGVs)
- Motor-schema-based formation control (move-to-goal, avoid-static-obstacle, avoid-robot, and maintain-formation)

# Formation Taxonomy

[Balch & Arkin, *IEEE TRA*, 1998]

- Based on the formation shape:

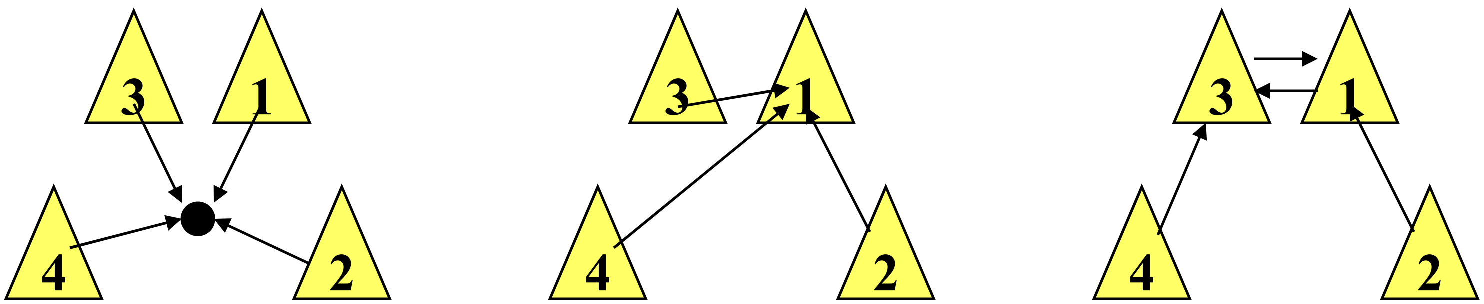



Note the vehicle ID!

# Formation Taxonomy

[Balch & Arkin, *IEEE TRA*, 1998]

- Based on the reference structure (ex. on wedge):



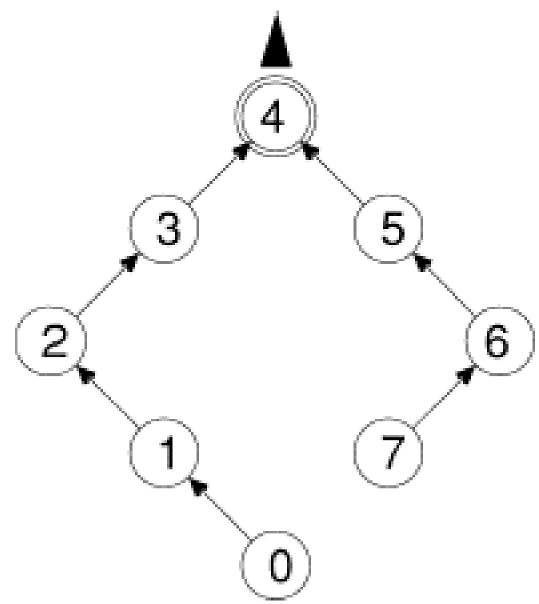

Leader-referenced

Unit-center-referenced

Neighbor-referenced

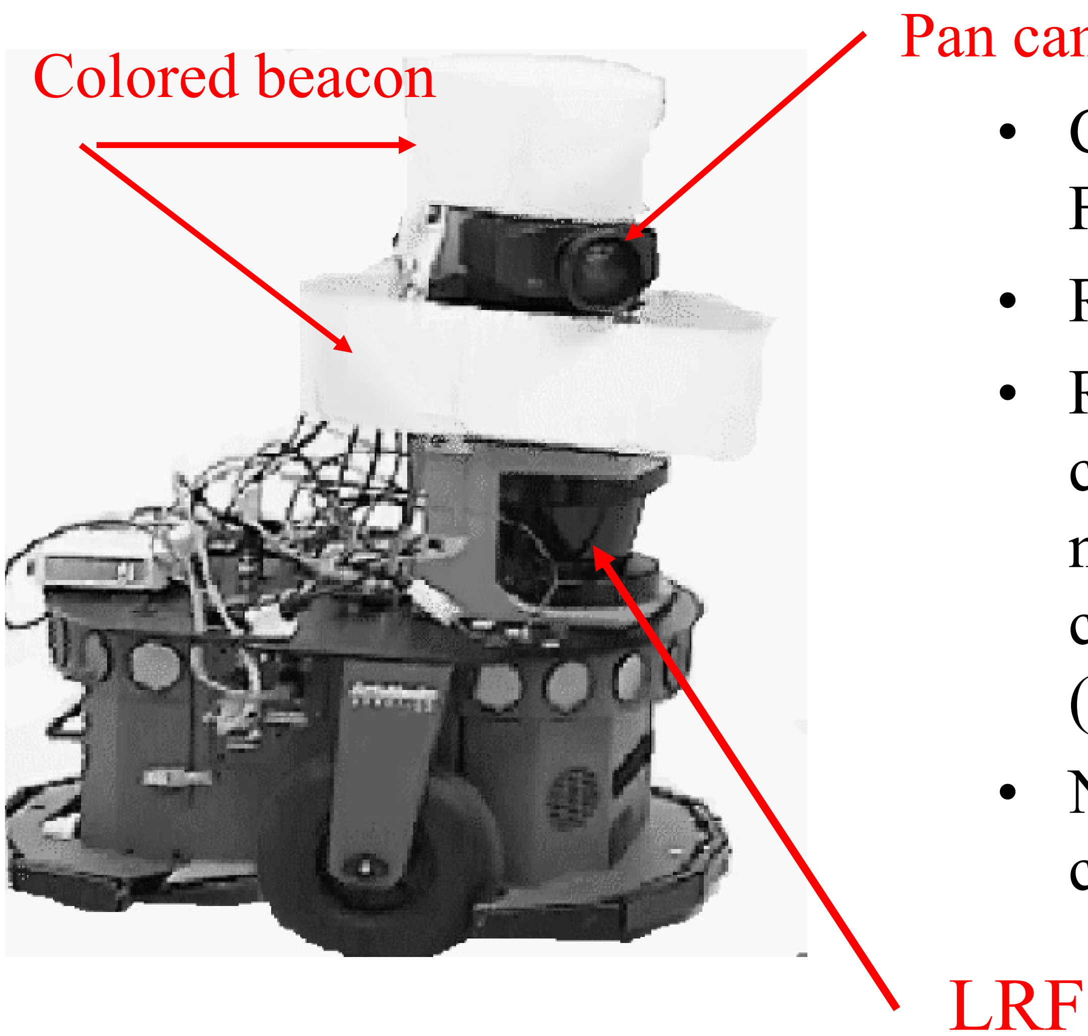
# Fredslund & Matarić (2002)

- Neighbor-referenced architecture based on a on-board relative positioning; single leader always
- Leader/formation speed: 2 cm/s
- Tested on 4 different formations (line, column, wedge, diamond) + switching between them
- Each robot has an ID and a global network can be formed, ID are broadcasted regularly
- As a function of the formation + order in the chain (ID-based rules), a relative range and bearing to another robot is calculated



# Fredslund & Matarić (2002)

## Hardware for inter-robot relative positioning




- Combined use of Laser Range Finder (LRF) and pan camera
- Relative range: LRF
- Relative angle: through the camera pan angle; neighboring robot kept in the center of view of the camera (also for a robustness sake)
- Neighboring robot ID: color code on visual beacon

# Fredslund & Matarić (USC, 2002)


**Laser Range Finder + vision**

**Highlights**

# Pugh et al. (2009) – Relative Localization Technology

See also Week 4 slides

Performance summary:

- Range: 3.5 m
- Update frequency 25 Hz with 10 neighboring robots (or 250 Hz with 1)
- Accuracy range: <7% (MAX), generally decrease 1/d
- Accuracy bearing: < 9º (RMS)
- Line-Of-Sight method
- Can also be used for 20 kbit/s IR com channel
- Measure range & bearing can be coupled with standard RF channel (e.g. 802.11) for heading assessment


[Pugh et al., *IEEE Trans. on Mechatronics*, 2009]

# Pugh et al (2009) – Formation Taxonomy

- Neighbor-referenced control using an on-board relative positioning system
- Approach: potential field control (similar to Balch 1998)
- Formations can be divided into two categories:
  - Location-based (position-based): robot group must maintain fixed location between teammates – robot headings don't matter
  - Heading-based (pose-based): robots must maintain fixed location and headings relative to teammates; subcategory: leader heading-based, where only the pose of leader is taken as reference

# Pugh et al (EPFL, 2009) – Formation Localization Modes

- **Mode 1**: No relative positioning – robots follow pre-programmed course with no closed-loop feedback
- **Mode 2**: Relative positioning – robots observe teammates with relative positioning module and attempt to maintain proper locations
- **Mode 3**: Relative positioning with communication – robots observe and share information with leader robot using relative positioning and wireless radio

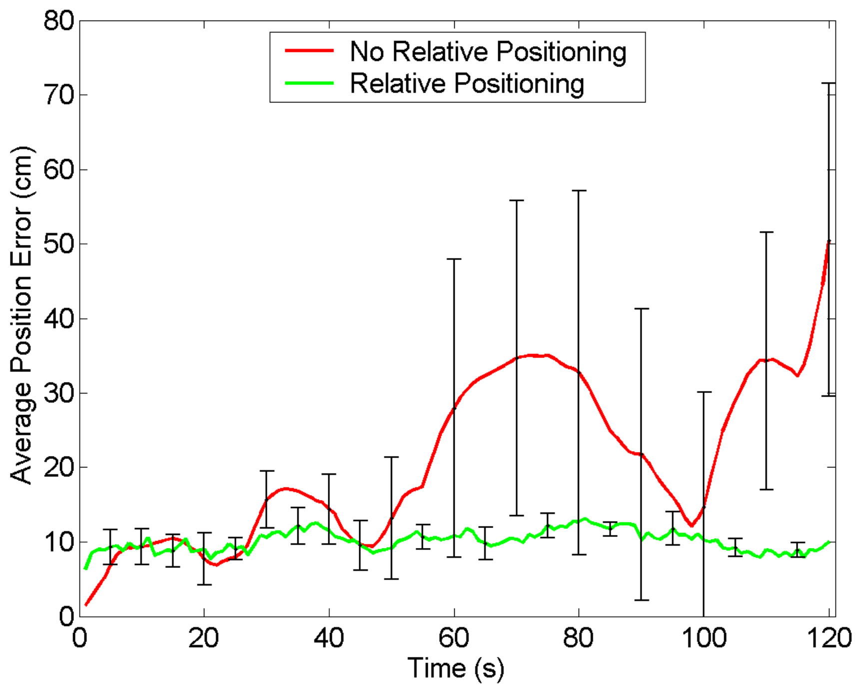
Notes:

- Mode 2 well-suited for location-based formation, Mode 3 well-suited for leader heading-based formation
- Pose-based formations in general can also be obtained with local localization systems that also deliver full pose of the neighbor without communication (e.g., multi-markers or shape detection + vision)

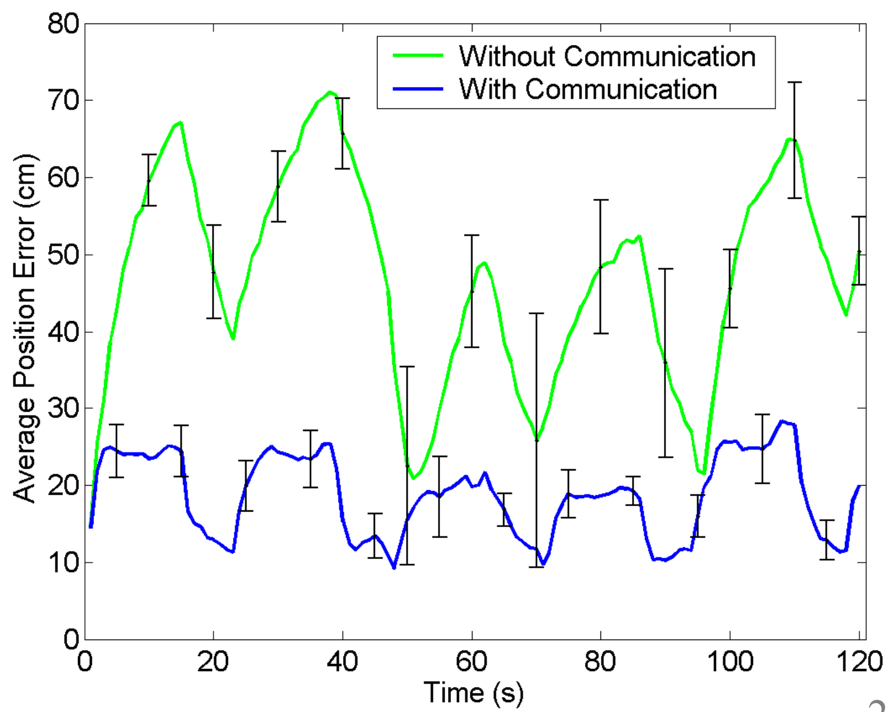
# Pugh et al (2009) - Sample Results

- Diamond formation movement (figure-eight pattern) with four robots
- Robot speed = 10 cm/s, update rate = 10-15 Hz
- Metric: average position error for the 4 robots respect to the prescribed diamond shape measured with an overhead camera system
- Results averaged over 10 runs, error bars indicate standard deviation

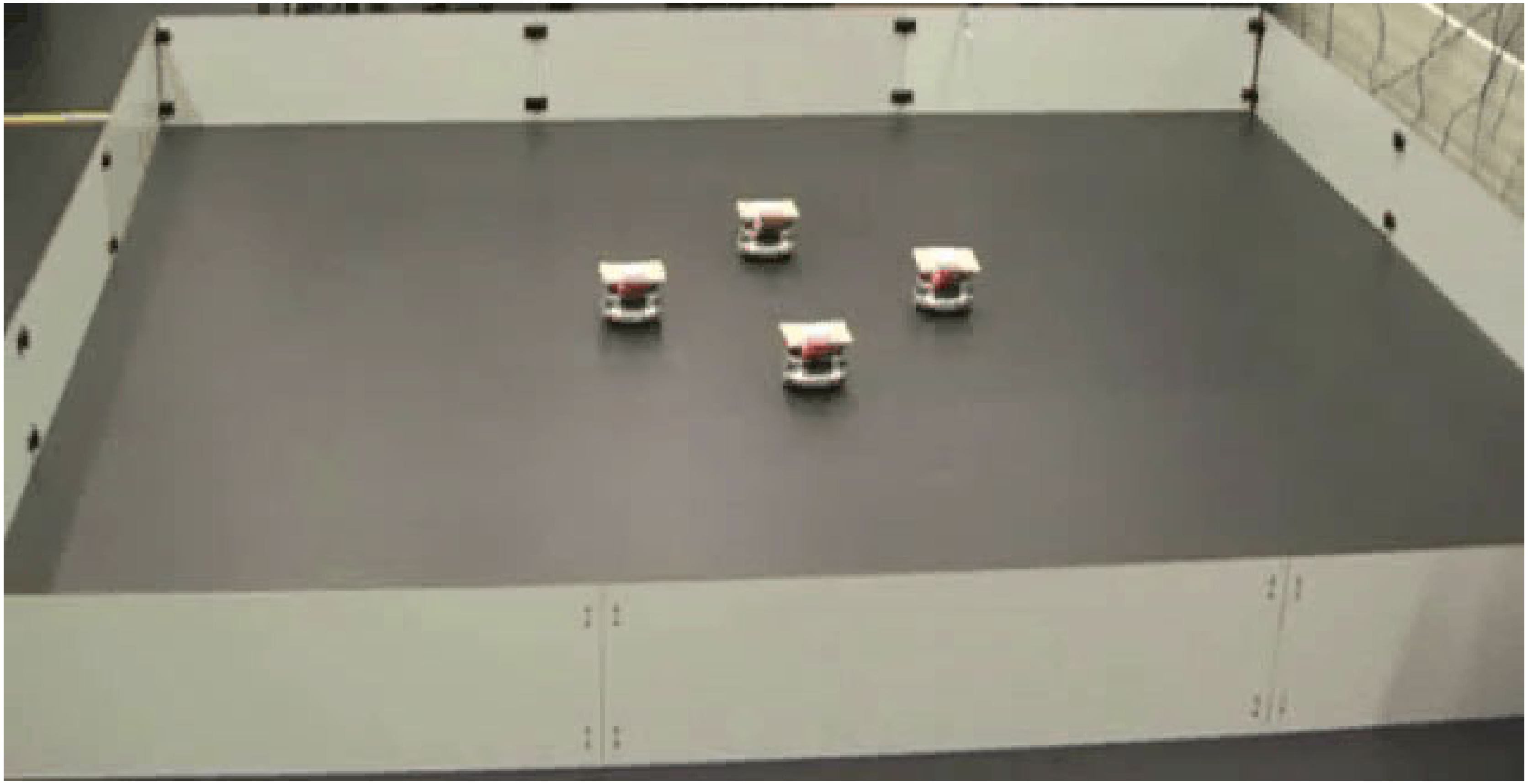
## Neighbor-Based Formation




## Heading-Based Formation

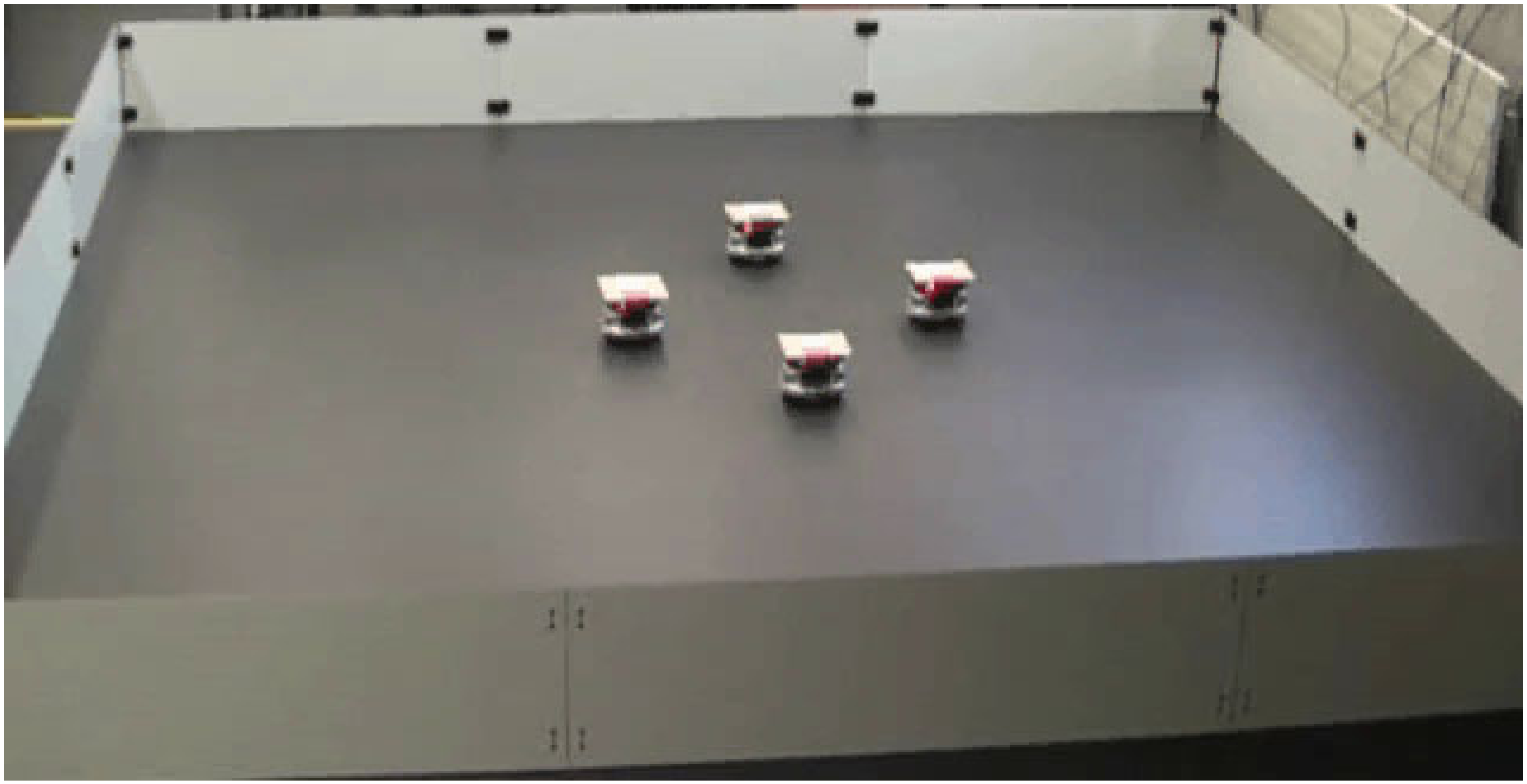


# Diamond Formation: Reactive Control



Location-based, 5x speed-up
[Pugh et al., 2009]

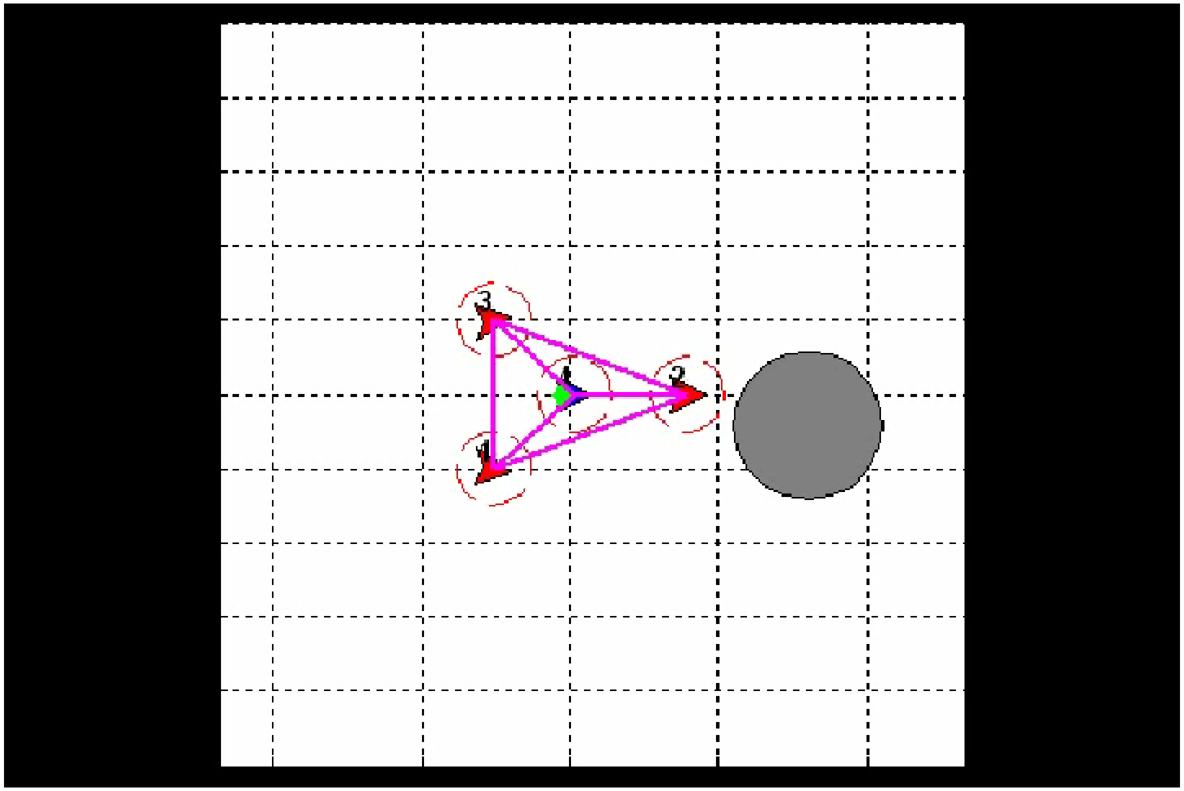
# Diamond Formation: Reactive Control

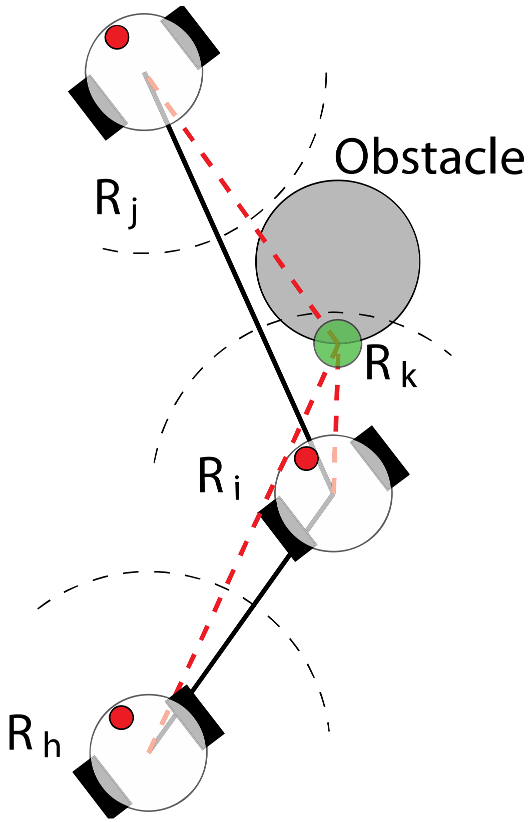


Heading-based, 5x speed-up
[Pugh et al, 2009]

# 2D Formations – Graph-Based (Consensus-Based) Control

# Motivation





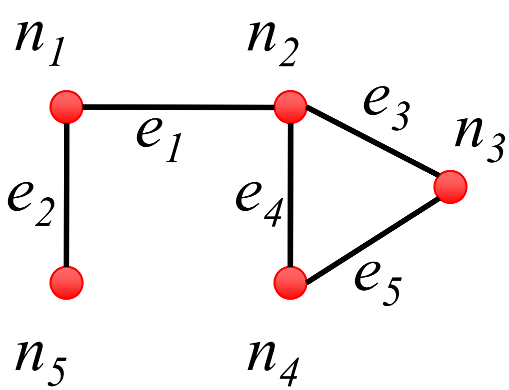

- Graph-theory to reconfigure, avoid obstacles, control cohesion or formation, …

# Definitions

Graph: $G = (V, E)$

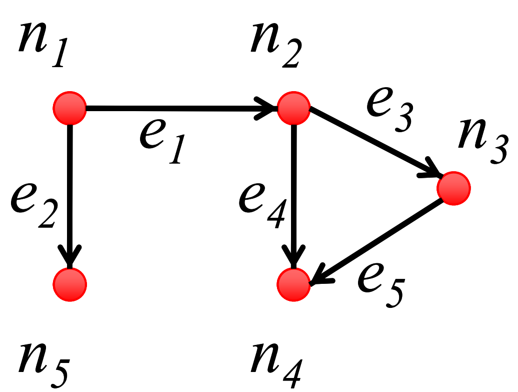
Vertex Set: $V = \{n_1, \ldots, n_N\}$

Edge Set: $E = \{(n_i, n_j) \in V \times V | n_i \neq n_j\}$




Graphs can be oriented (directed), but we will assume unoriented (undirected) graphs in this lecture.

# Definitions




$$I = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

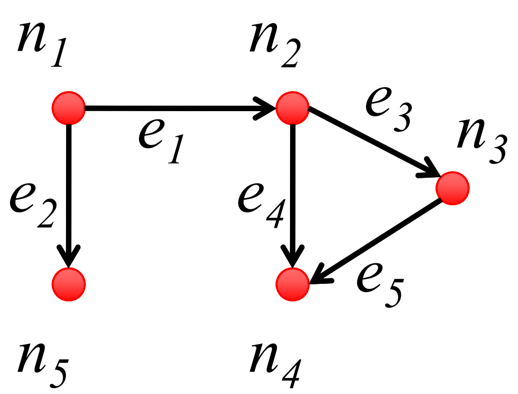
Incidence Matrix:

Define $\mathcal{I} \in \mathbb{R}^{\|\mathcal{V}\| \times \|\mathcal{E}\|}$ as:

$$\mathcal{I}(i,j) = \begin{cases} -1, & \text{if } e_j \text{ leaves } n_i \\ 1, & \text{if } e_j \text{ enters } n_i \\ 0, & otherwise \end{cases}$$

If the graph is unoriented, we can **arbitrarily choose an orientation** for any edge.

# Definitions




Weight Matrix:

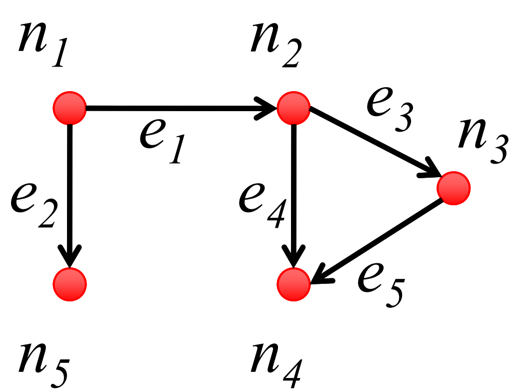
Define $\mathcal{W} \in \mathbb{R}^{\|\mathcal{E}\| \times \|\mathcal{E}\|}$ as:

$$\mathcal{W}(i,j) = \begin{cases} w_i, & \text{if } i = j \\ 0, & otherwise \end{cases}$$

$$W = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_5 \end{bmatrix}$$

$w_i$ represents the weight associated with the edge $e_i$. "The bigger the weight the more important the edge becomes."

# Definitions




Laplacian Matrix:

Define $\mathcal{L} \in \mathbb{R}^{\|\mathcal{V}\| \times \|\mathcal{V}\|}$ as:

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T$$

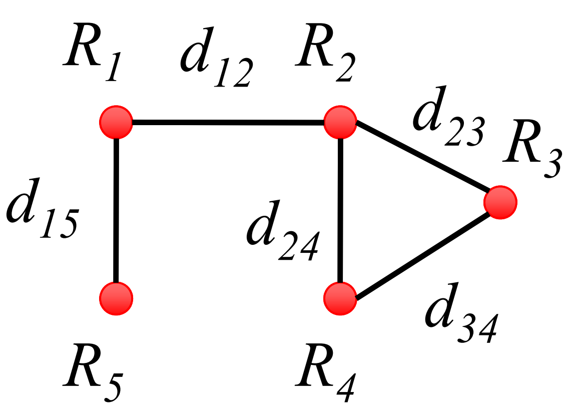
$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

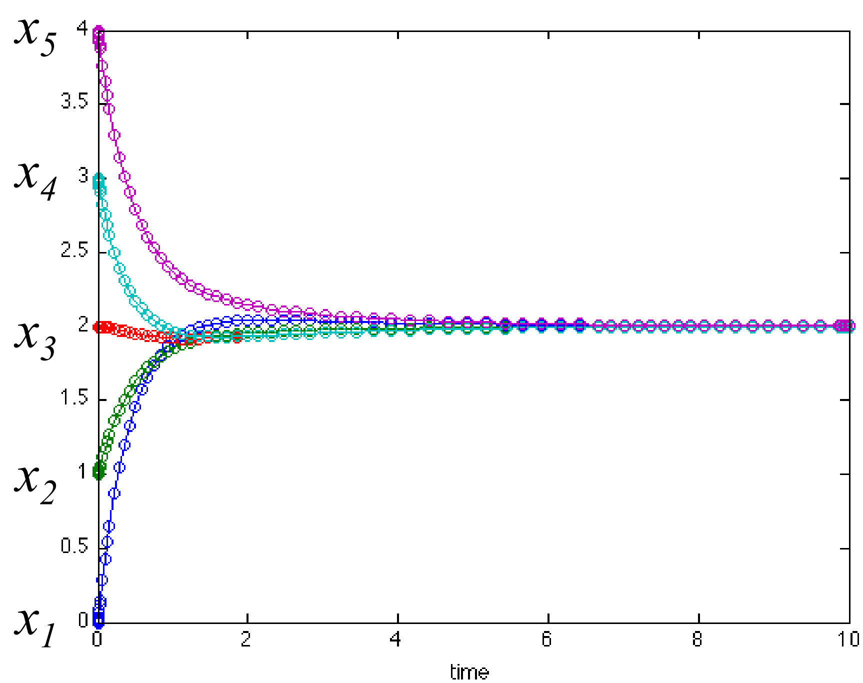
If $w_i \neq 1$, for any $i$, then the Laplacian matrix is called the weighted Laplacian matrix.

# The *Rendezvous* Problem in 1D

- Each node is given a state $x_i$, the goal is to make $x_1 = x_2 = \ldots = x_N$ as time tends to infinity.
- Final consensus value not pre-established but consensus framework (e.g., variable type, range) is shared and defined a priori
- Imagine 5 robots $R_i$ moving on 1D measuring distances $d_{ij}$

$$\boxed{\lim_{t \to \infty} x_i(t) = x^*, \forall i}$$



# Solving the *Rendezvous* Problem

- One way to solve the rendez-vous problem is to use the Laplacian matrix:

$$\dot{x}(t) = -\mathcal{L}x(t)$$

- This is equivalent to the following formulation (assume 1D moving robots):

$$\dot{x}_i = \sum_{\mathbf{R}_j \in \mathcal{N}_i} w_{ij}(x_j - x_i)$$

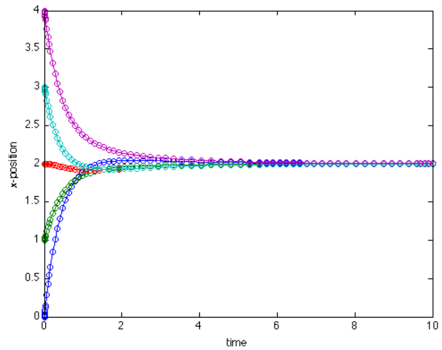
$\mathbf{R_j}$ = robot j
$\mathbf{N_i}$ = neighborhood of robot i

- If $w_{ij} > 0$, graph connected, rendezvous is guaranteed

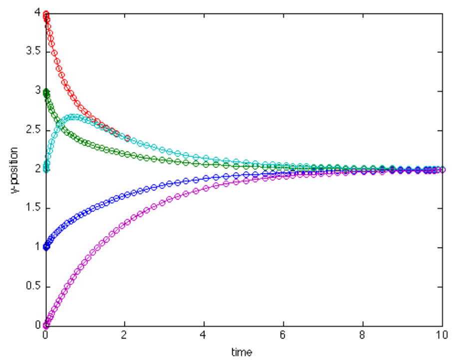
# *Rendezvous* Problem in 2D

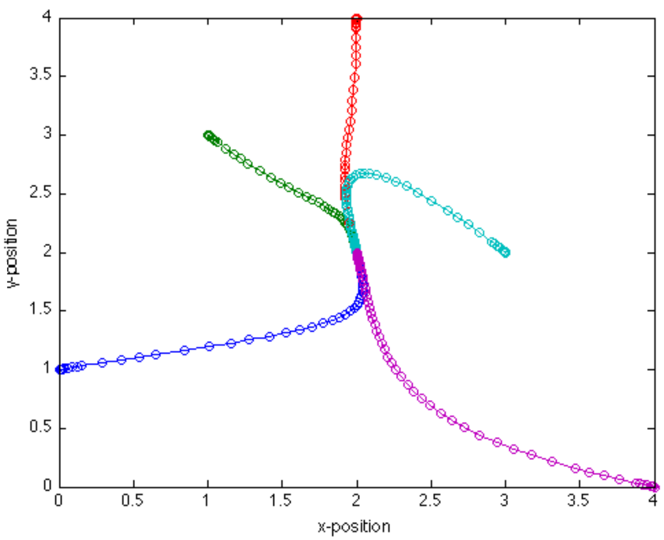
- We simply solve the rendez-vous problem for each dimension separately.

$$\dot{x}(t) = -\mathcal{L}x(t)$$

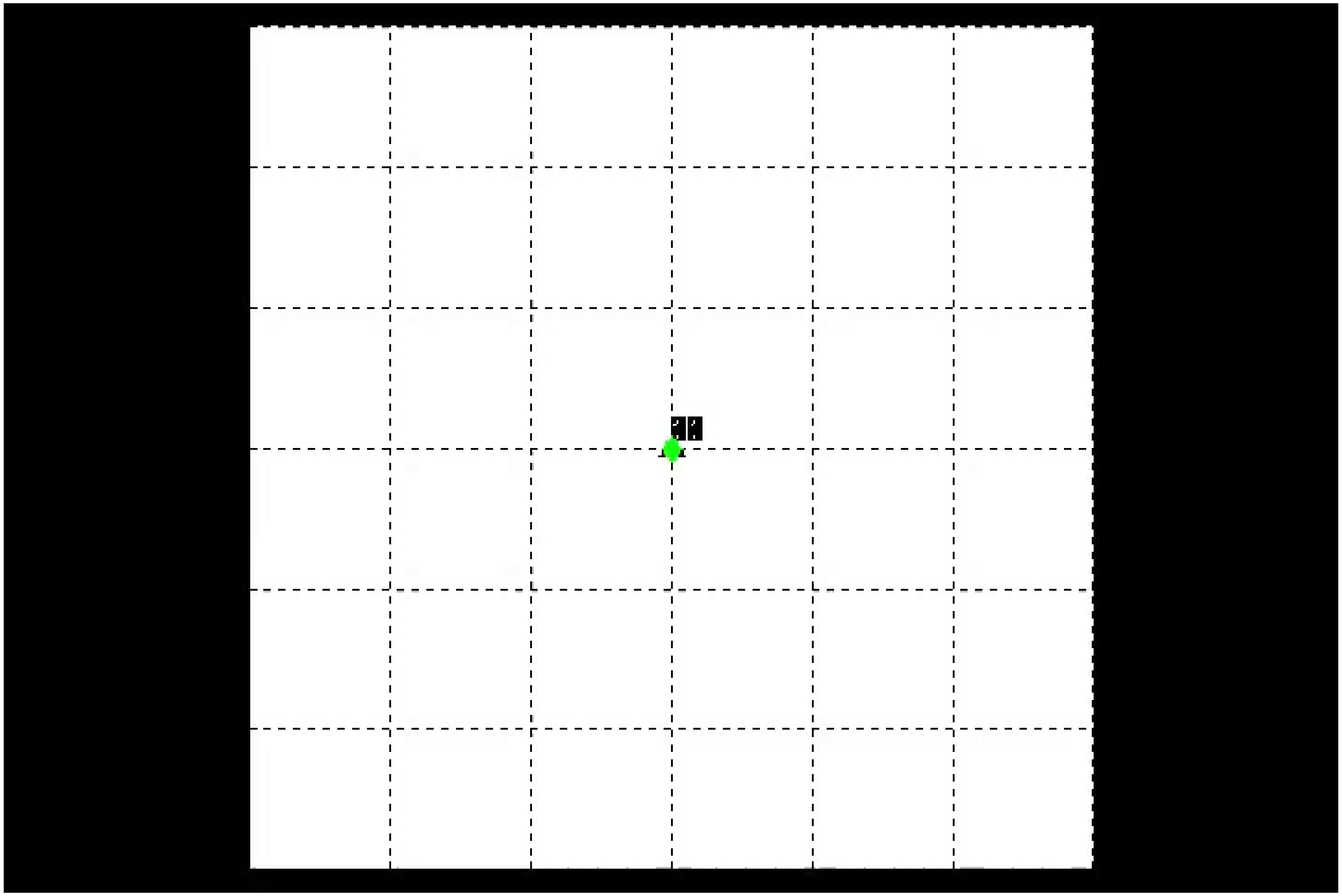



$$\dot{y}(t) = -\mathcal{L}y(t)$$



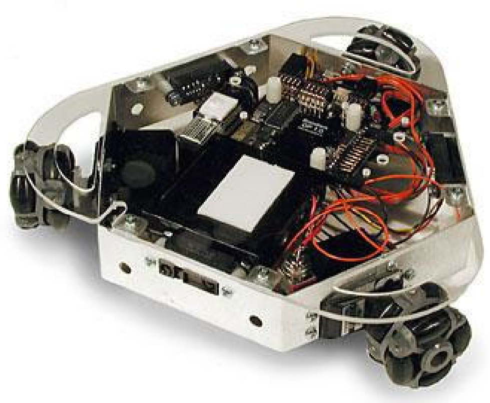



# *Rendezvous* Problem in 2D

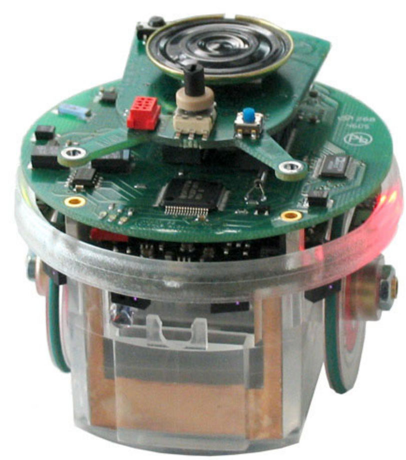


# Holonomic Robots

- Holonomic: total number of degree of freedom = number of controllable degree of freedom.
- From the point of view of mobility: a mobile robot is holonomic if it can move in any direction at any point in time.



Holonomic



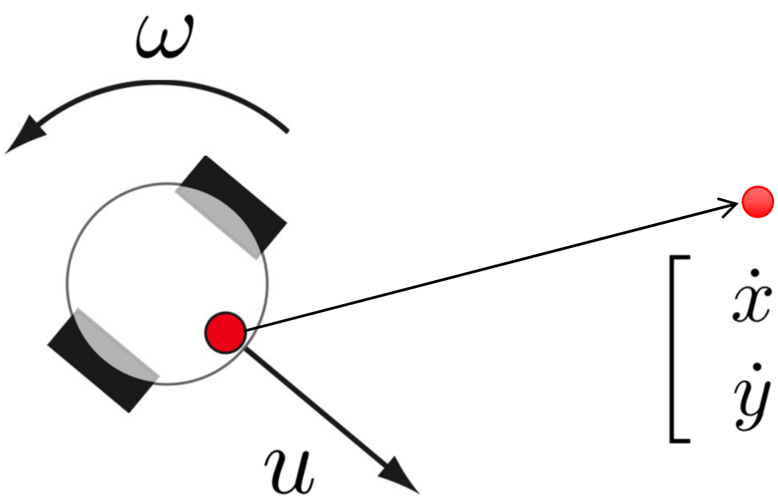
Non-holonomic

# Some Considerations

- The *Laplacian method* gives the direction vector at each point in time.
- If we have **holonomic** robots we can simply go in that direction.
- If we don't… we will need to **transform** the direction vector in something useable by the robots given their mobility constraints (captured in their kinematic model)

# Transformation

- From total degrees of freedom (DOFs) to controllable DOFs (and eventually to actuator control via the inverse kinematic model)
- Note: rendez-vous is not supposed to find consensus on the full pose, only position.



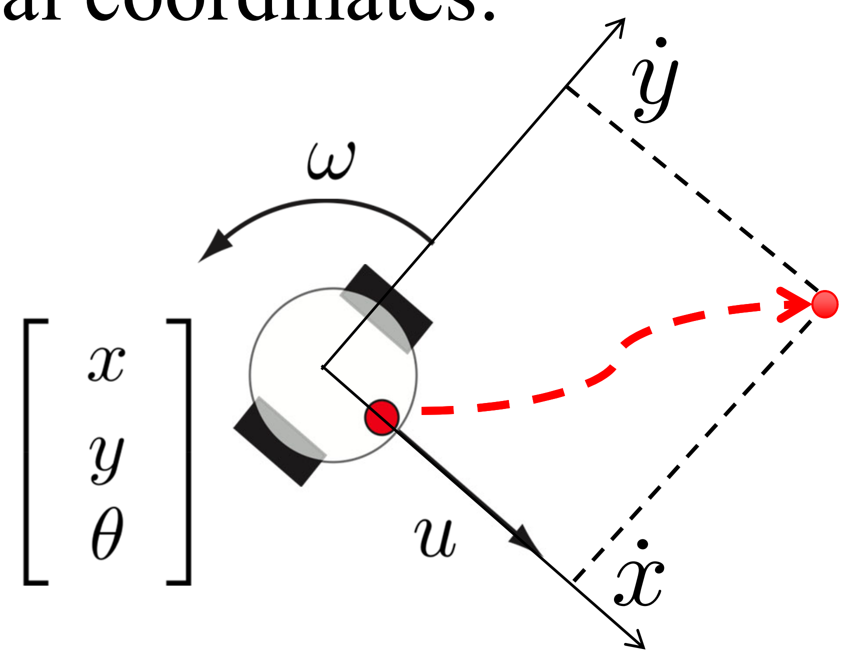

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

It's all about finding the right function *f* such that:

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = f(\dot{x}, \dot{y})$$

$$f(\dot{x}, \dot{y})$$

- We want a function that makes the robot move from its current position to its position plus the derivative of the position.
- First, let's transform the global coordinates to local coordinates:



$$f(\dot{x}, \dot{y})$$

- Then, the following transformation achieves the requirements:

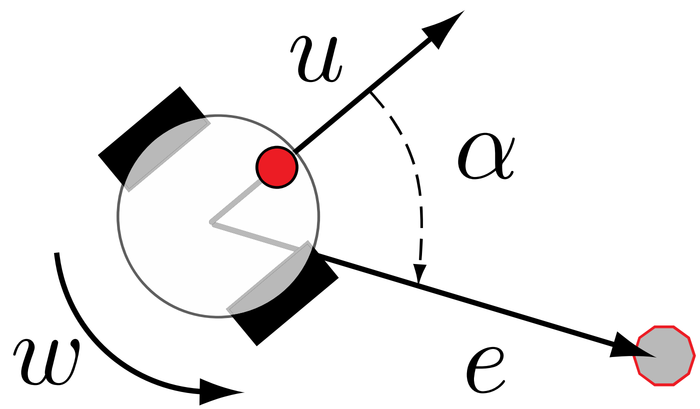
$$\begin{cases} u = \underbrace{K_u}_{\text{Constants}} \cdot \overbrace{\sqrt{\dot{x}^2 + \dot{y}^2}}^{\text{Distance to the goal}} \cdot \overbrace{\cos(\operatorname{atan2}(\dot{y}, \dot{x}))}^{\text{Motion direction}} \\ \omega = K_\omega \cdot \underbrace{\operatorname{atan2}(\dot{y}, \dot{x})}_{\text{Direction to the goal}} \end{cases}$$

- The motion is directed toward the goal and its speed is proportional to the distance to that goal.

**Proportional (P) controller**

# Non-Holonomicity

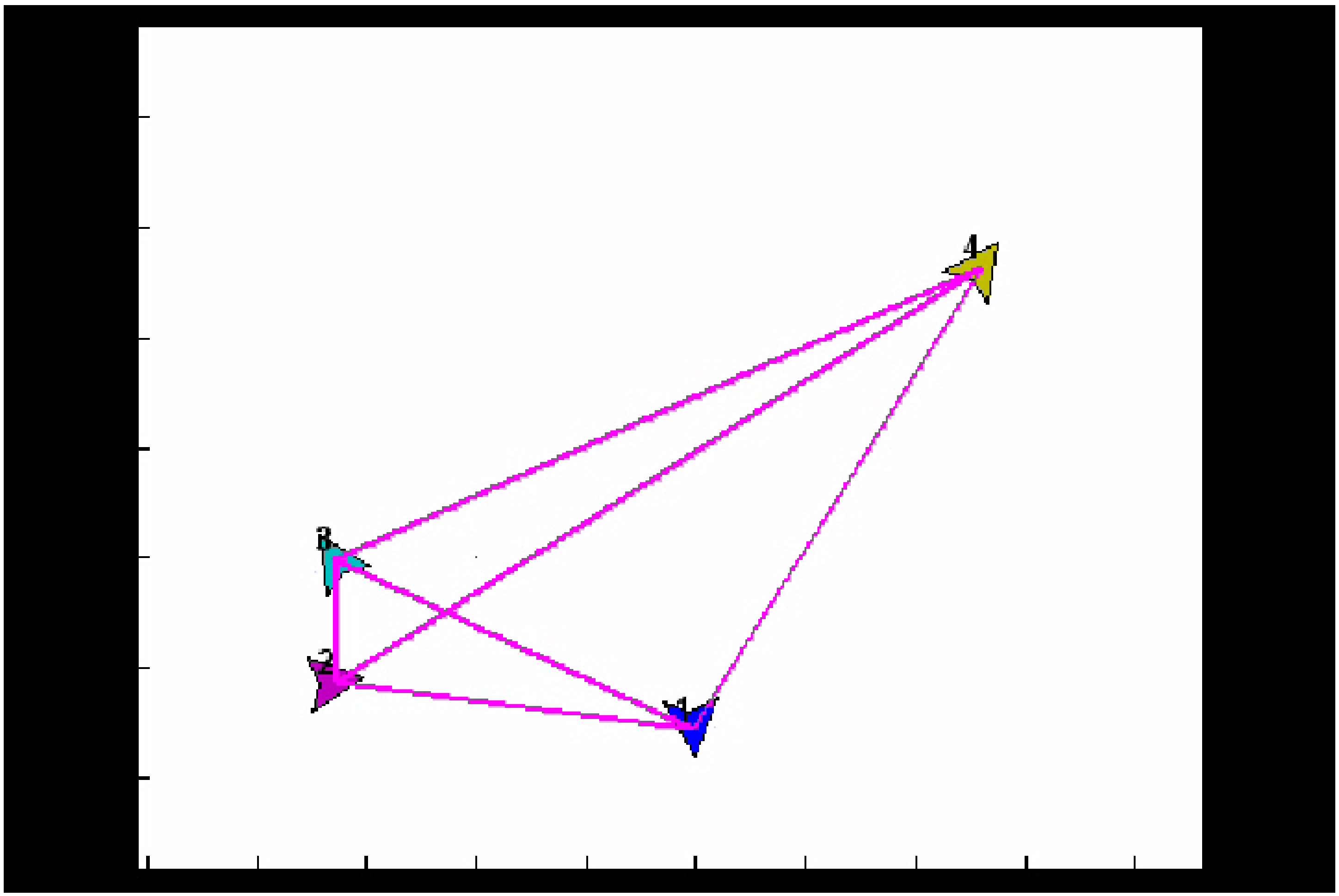
- We can also use relative range and bearing:



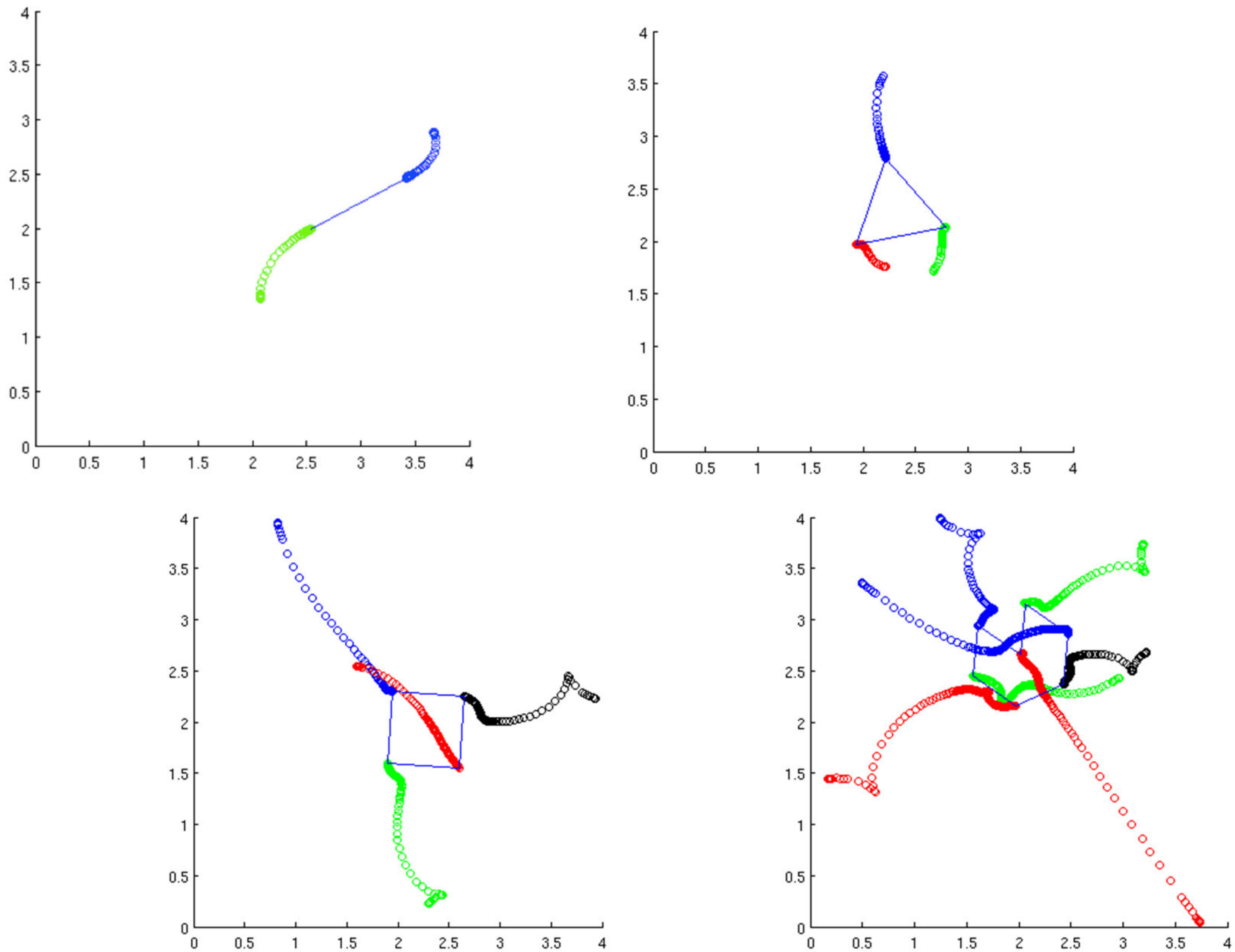

- Then:

$$u = K_u e \cos \alpha$$
$$w = K_w \alpha$$

# Non-Holonomicity



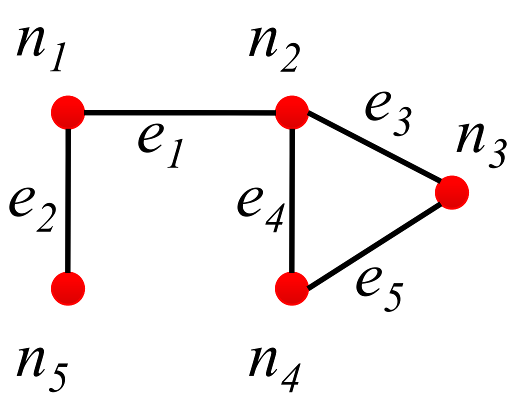
# Reconfiguring

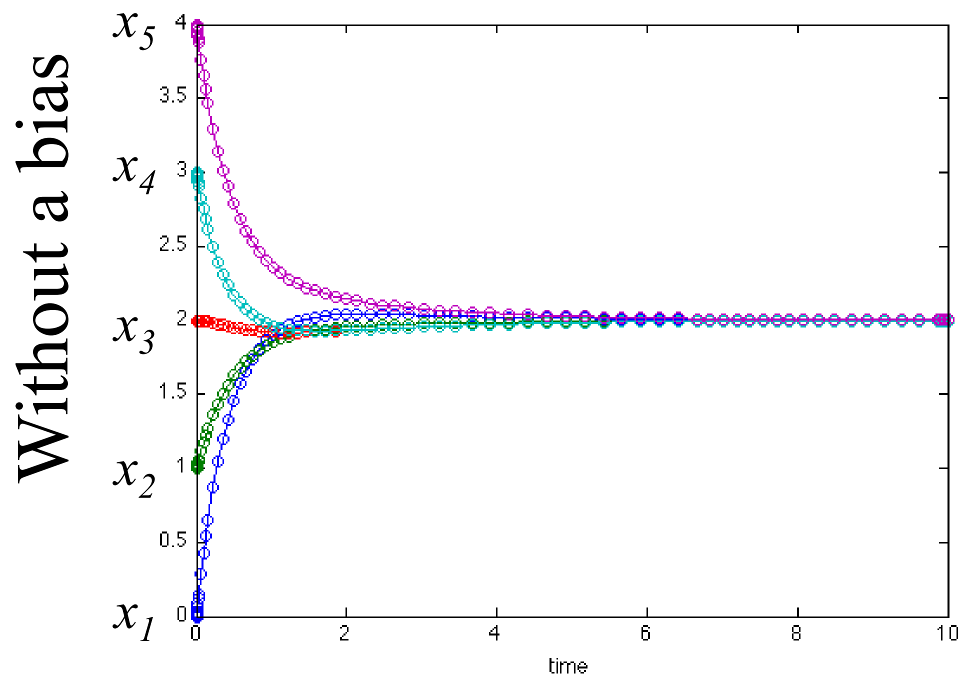


# Configurations Using a Bias

- By adding a bias vector, we can modify the state (or assumed position):

$$\dot{x} = -\mathcal{L}(x(t) - \mathcal{B})$$

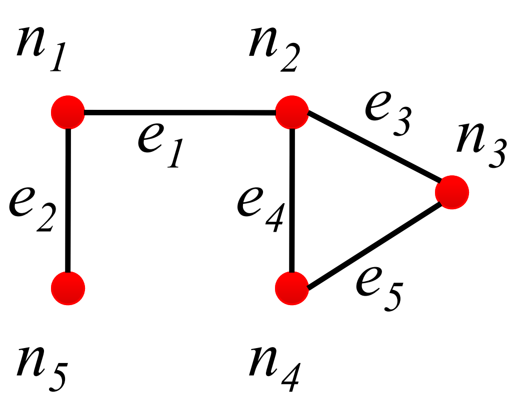


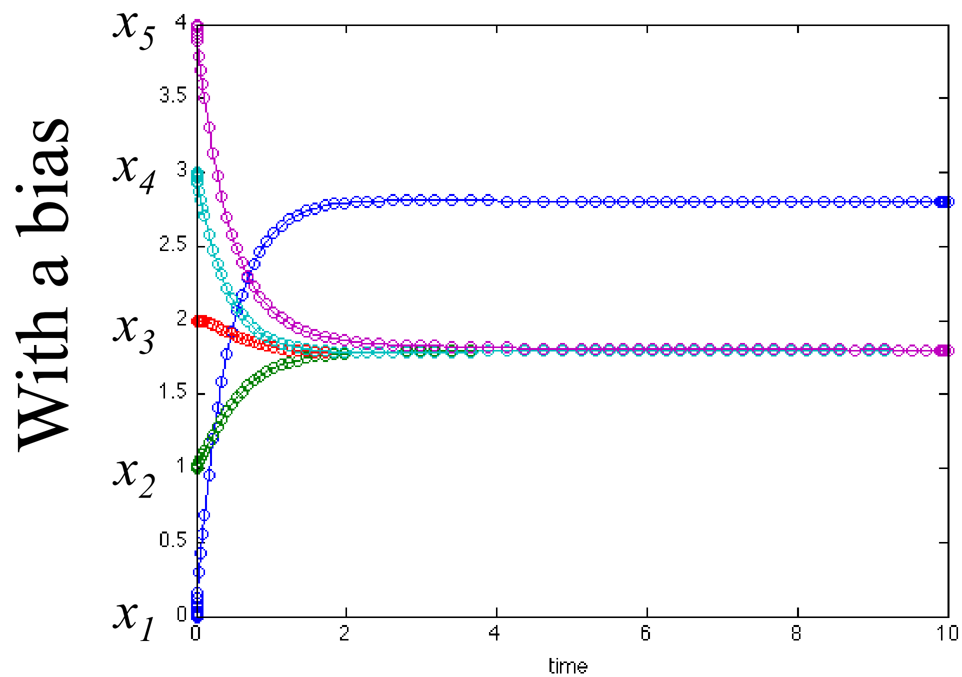



# Configurations Using a Bias

- By adding a bias vector, we can modify the state (or assumed position):

$$\dot{x} = -\mathcal{L}(x(t) - \mathcal{B})$$

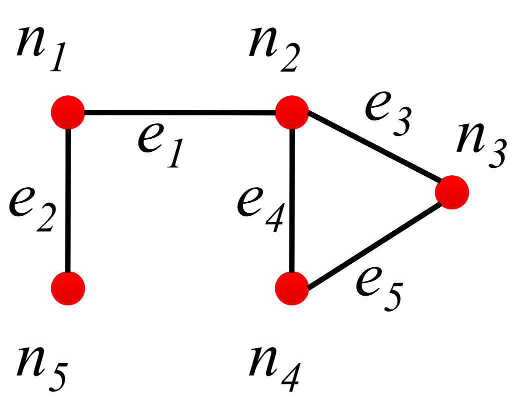


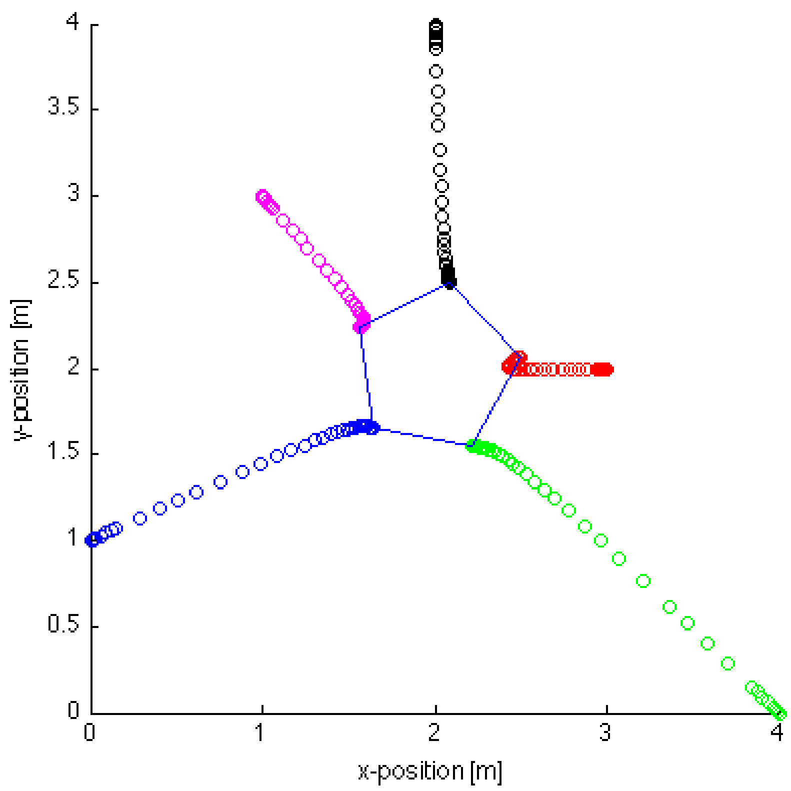



# Configurations Using a Bias

- By adding a bias vector, we can modify the state (or assumed position):

$$\dot{x} = -\mathcal{L}(x(t) - \mathcal{B})$$

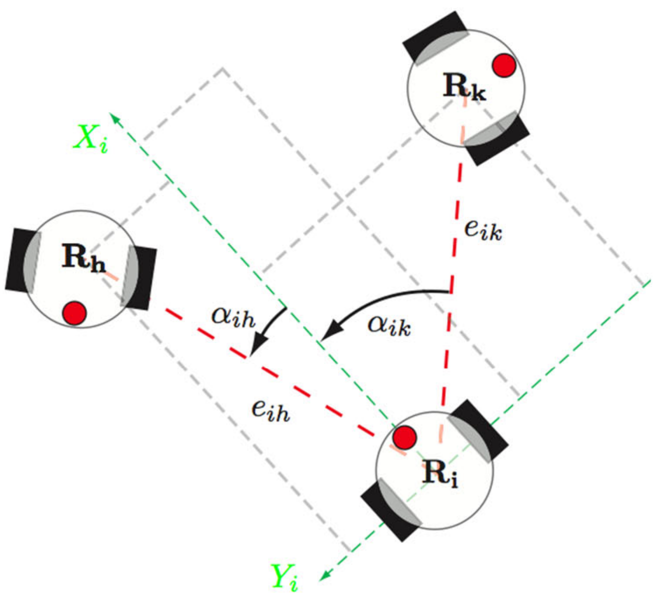


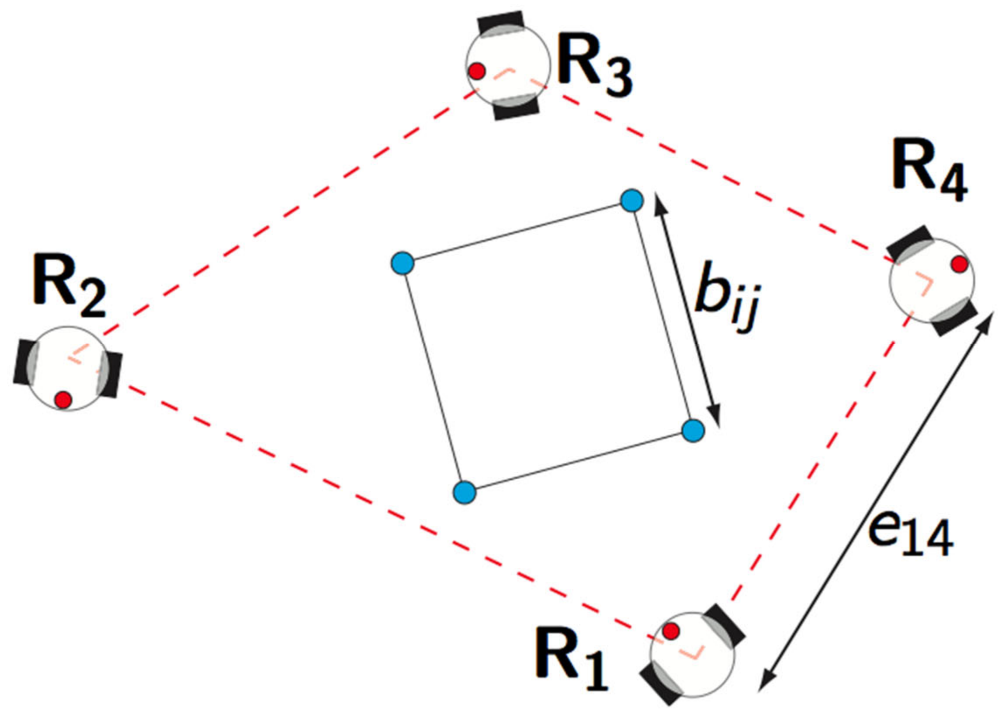
# Decentralized Version

- Each robot solves the Laplacian equation taking as *x* and *y* the relative coordinates of the other robots.

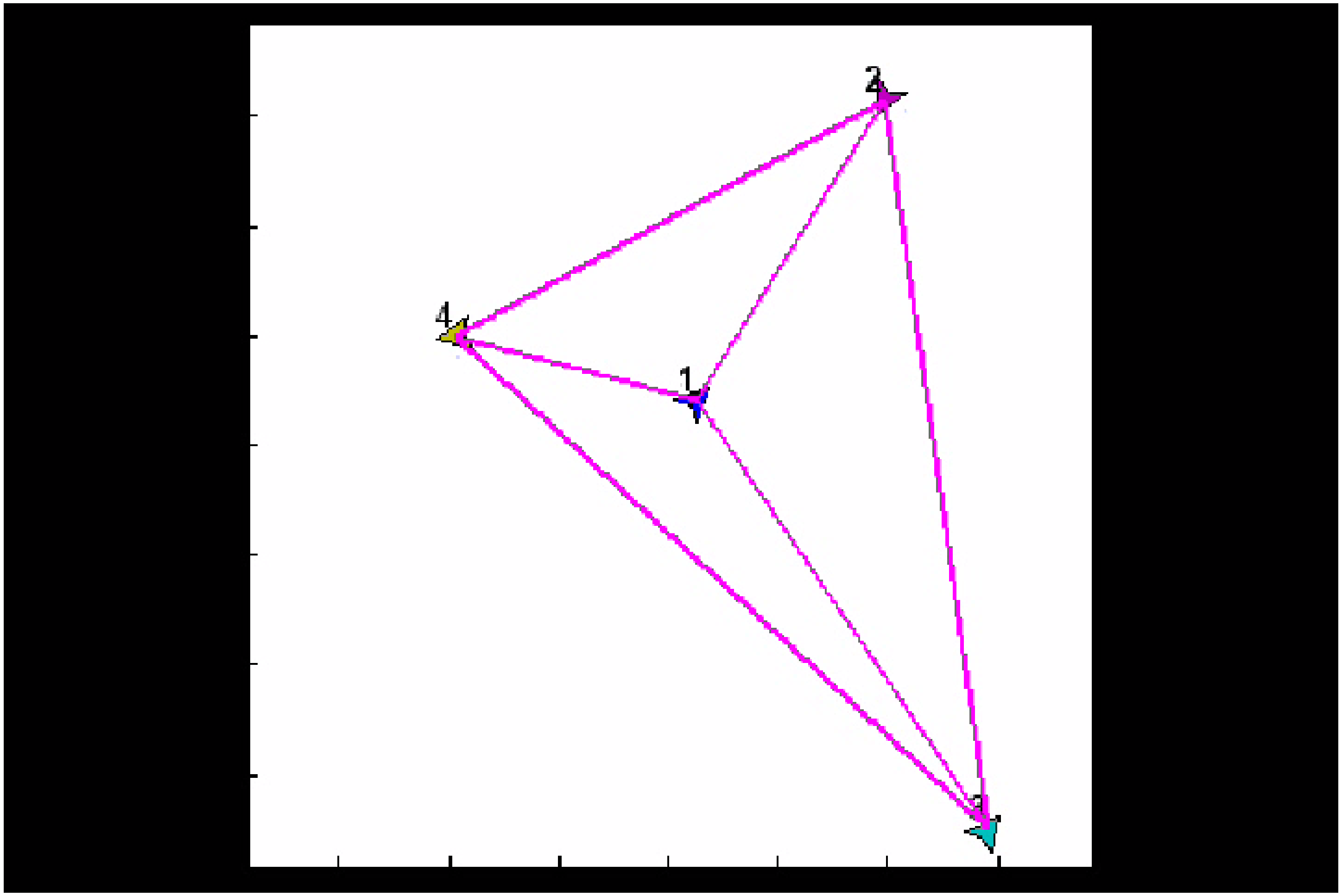
Bias



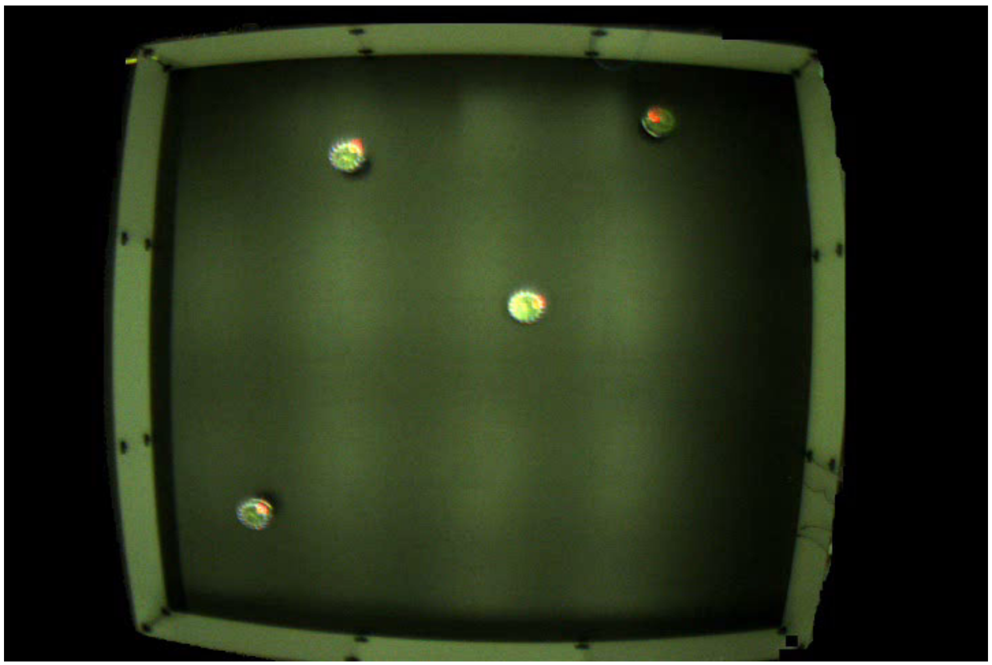

Sensing

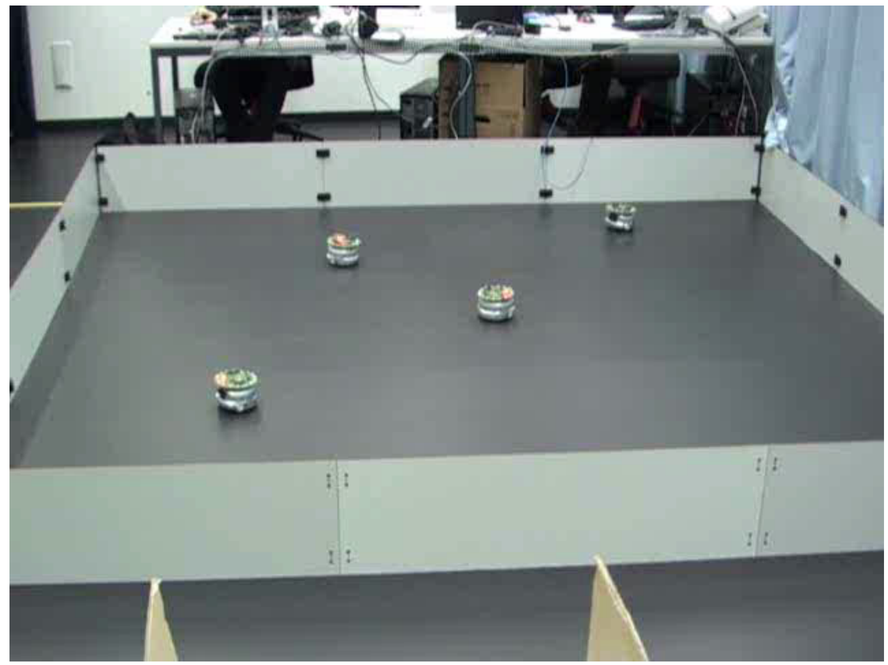


# Example of Reconfiguration



# Real Robots





[Falconi et al., Robocomm 2009; ICRA 2010]

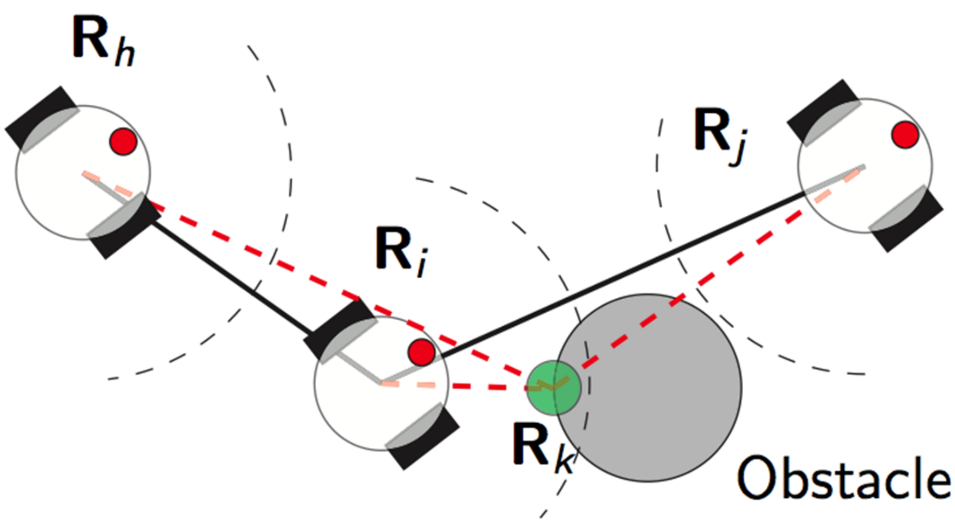
# What's Next?

- Current status: non-holonomic robots are able to reconfigure in any shape
- Can we perform **obstacle avoidance** or/and **cohesion control** using the same ideas?

Yes

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T$$

# Obstacle avoidance

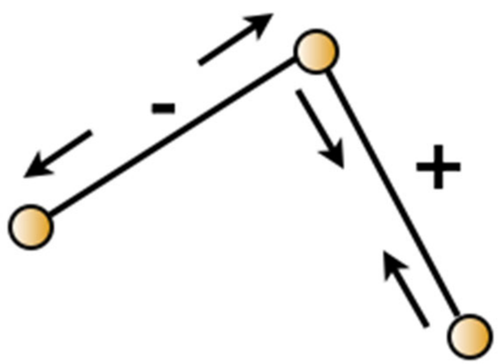



- When an obstacle is detected by a robot, its position is propagated to other robots.
- Each robot updates its neighbors list if necessary by adding a repulsive agent.

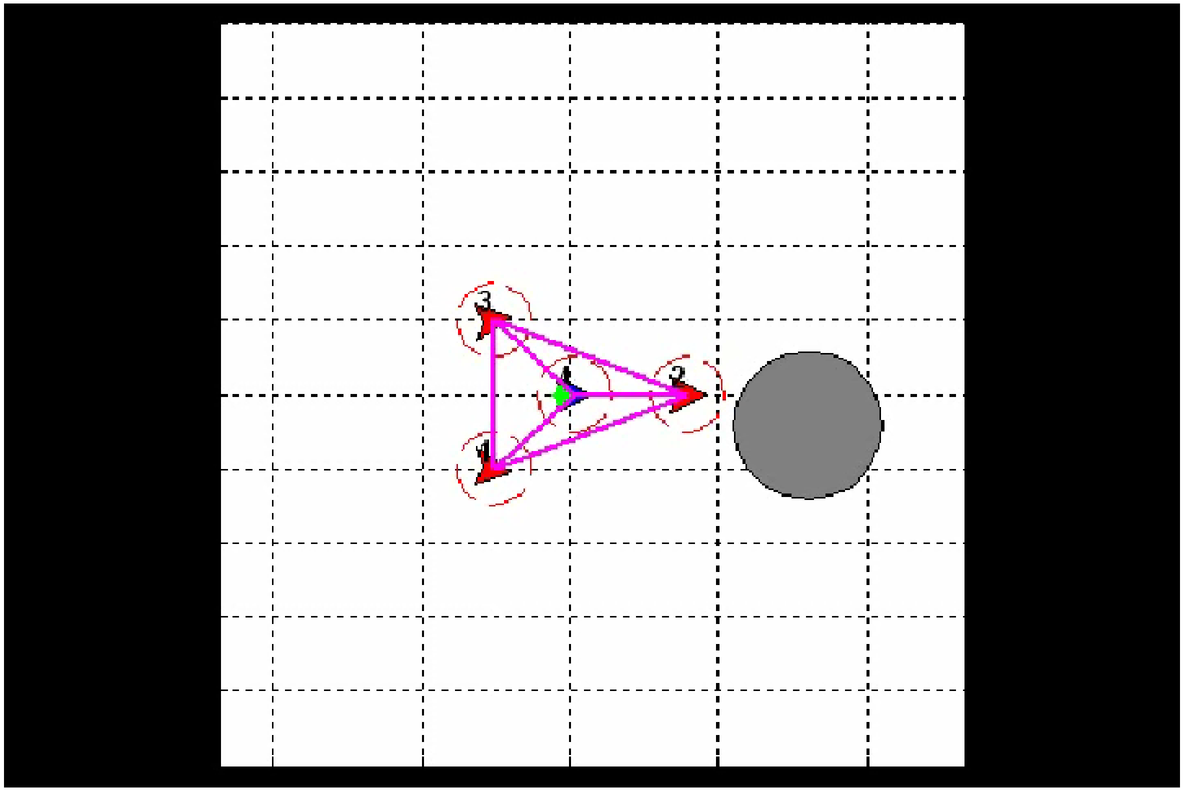
# Obstacle Avoidance

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T$$

- Positive weights will attract vehicles together.
- Negative weights will create a repulsion mechanism.
- This can be used for obstacles or other robots

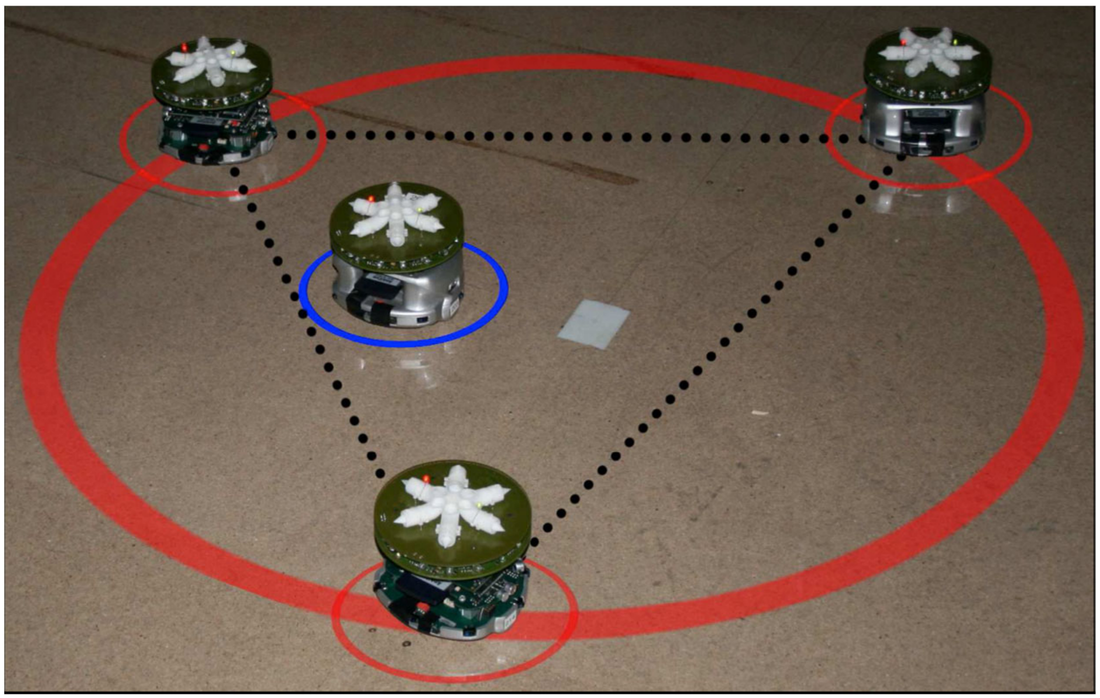


# Obstacle Avoidance



# What about Formation Control?

- Our goal is to enable a group of robots (the *followers*) to follow a robotic leader.



[Falconi et al., ICRA 2010]

# Formation Control

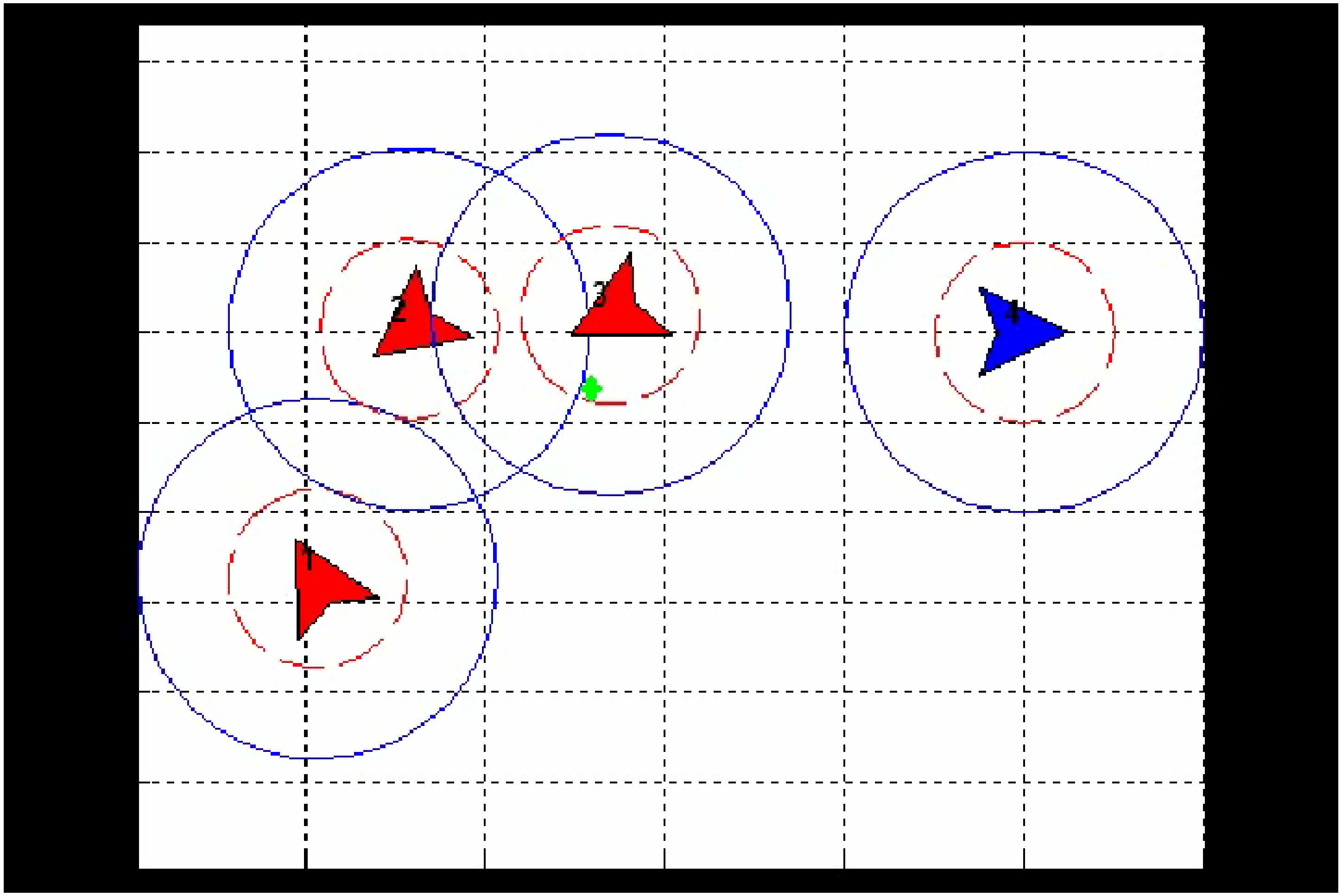
- Until now, we only changed the weights, but we can also modify the control law.
- If we have a single leader moving at a constant velocity, we can add an integral term:

$$u = K_u e \cos\alpha + K_I \int_0^t e \, \mathrm{d}t$$

$$w = K_w \alpha$$

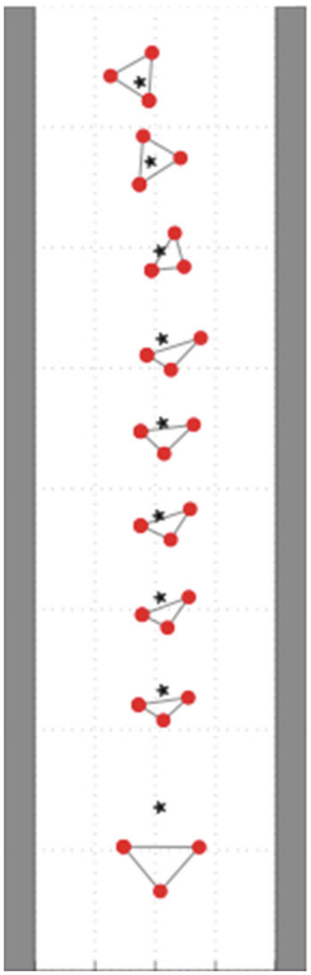
**Proportional, integral (PI) controller**

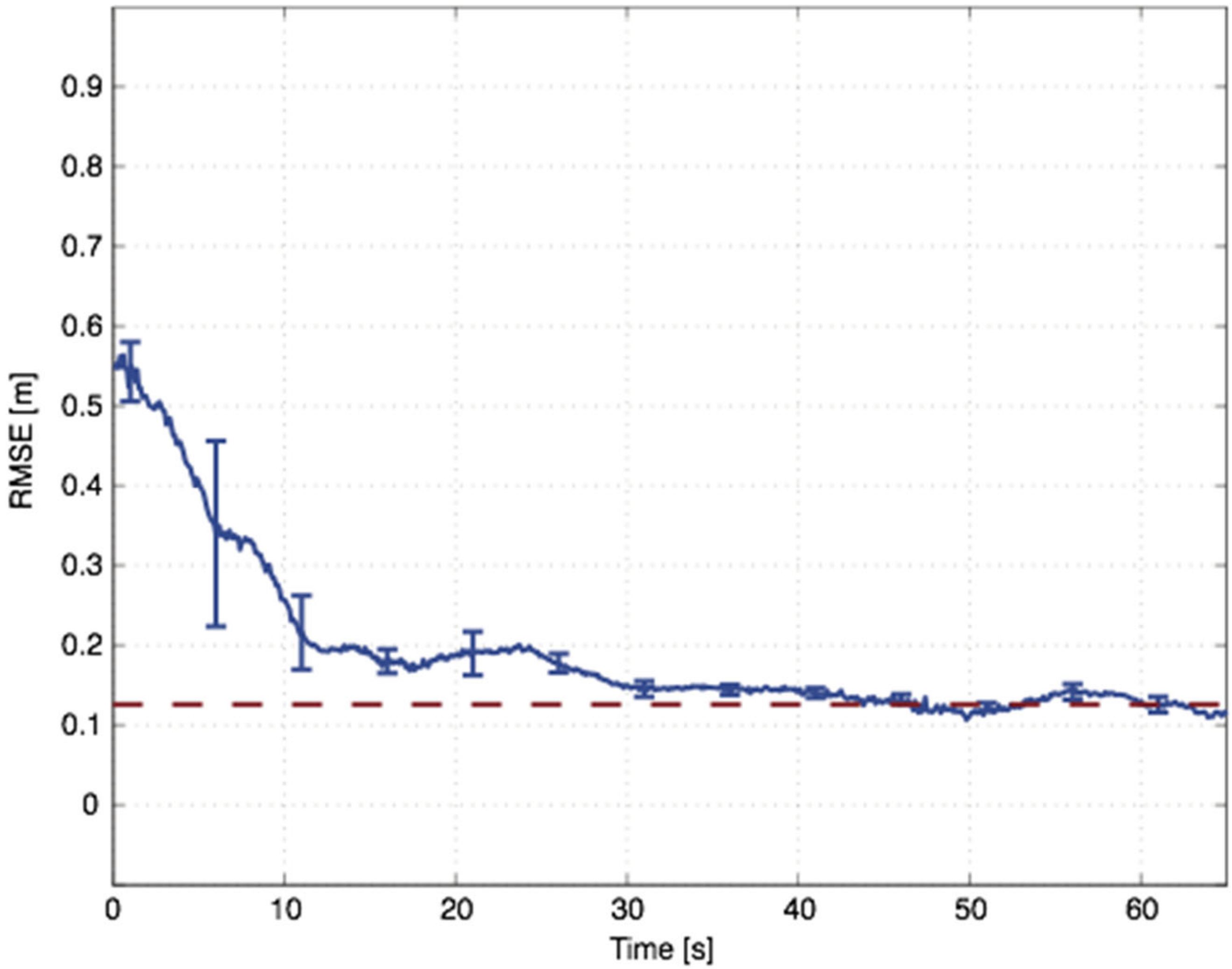
# Formation Control in Matlab



# Formation Control in Webots

# Formation Control in Real Robots





# Adding an Estimation Layer for Dealing with Noisy Localization

Heading-based, 4x speed-up
[Gowal and Martinoli, 2013]

# Conclusion

# Take Home Messages

- Flocking can be considered a loose formation
- Major differences between virtual and real agents in communication, sensing, actuation, and control
- Formations and flocking can be obtained in a number of ways, depending on the underlying inter-robot positioning technology and corresponding control rules
- Formation maintenance becomes more challenging when the full pose  (instead of only positions) has to be maintained
- Consensus-based control laws can be captured with graph-based formalism; this formalism is powerful and allow for decentralized control architectures while maintaining theoretically provable properties

# Additional Literature – Week 6

**Books**


- Ren W. and Beard R., “Distributed Consensus in Multi-vehicle Cooperative Control”, Springer, 2008.
- M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010


**Papers**


- Kelly I. D., Keating D. A., “Flocking by the Fusion of Sonar and Active Infrared Sensors on Physical Autonomous Mobile Robots”. *Proc. of the Third Conf. on Mechatronics and Machine Vision in Practice*, Gaimardes, Portugal, 1996, Vol. 1, pp. 1-4.
- Hayes A. T. and Dormiani-Tabatabaei P., “Self-Organized Flocking with Agent Failure: Off-Line Optimization and Demonstration with Real Robots”. *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation*, May 2002, Washington DC, USA, pp. 3900-3905.
- Vicsek T., Czirok A., Ben-Jacob E., Cohen I, and Schochet O, “Novel Type of Phase Transition in a System of Self-Driven Particles”. *Physical Review Letters*, **75**(6): 1226-1229, 1995.
- Jadbabaie Ali, Lin Jie, and Morse A. Stephen, “Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules”, *IEEE Trans. on Automatic Control*, **48**(6):988-1001, 2003
- R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. on Automatic Control*, **49**(9):1520–1533, 2004.
- Gowal S., Falconi R., and Martinoli A., “Local Graph-based Distributed Control for Safe Highway Platooning”. *Proc. of the 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* October 2010, Taipei, Taiwan, pp. 6070-6076.