Mobile Robot Systems

Lecture 7: Multi-Robot Systems - Collective Movement

Dr. Amanda Prorok

asp45@cam.ac.uk

www.proroklab.org

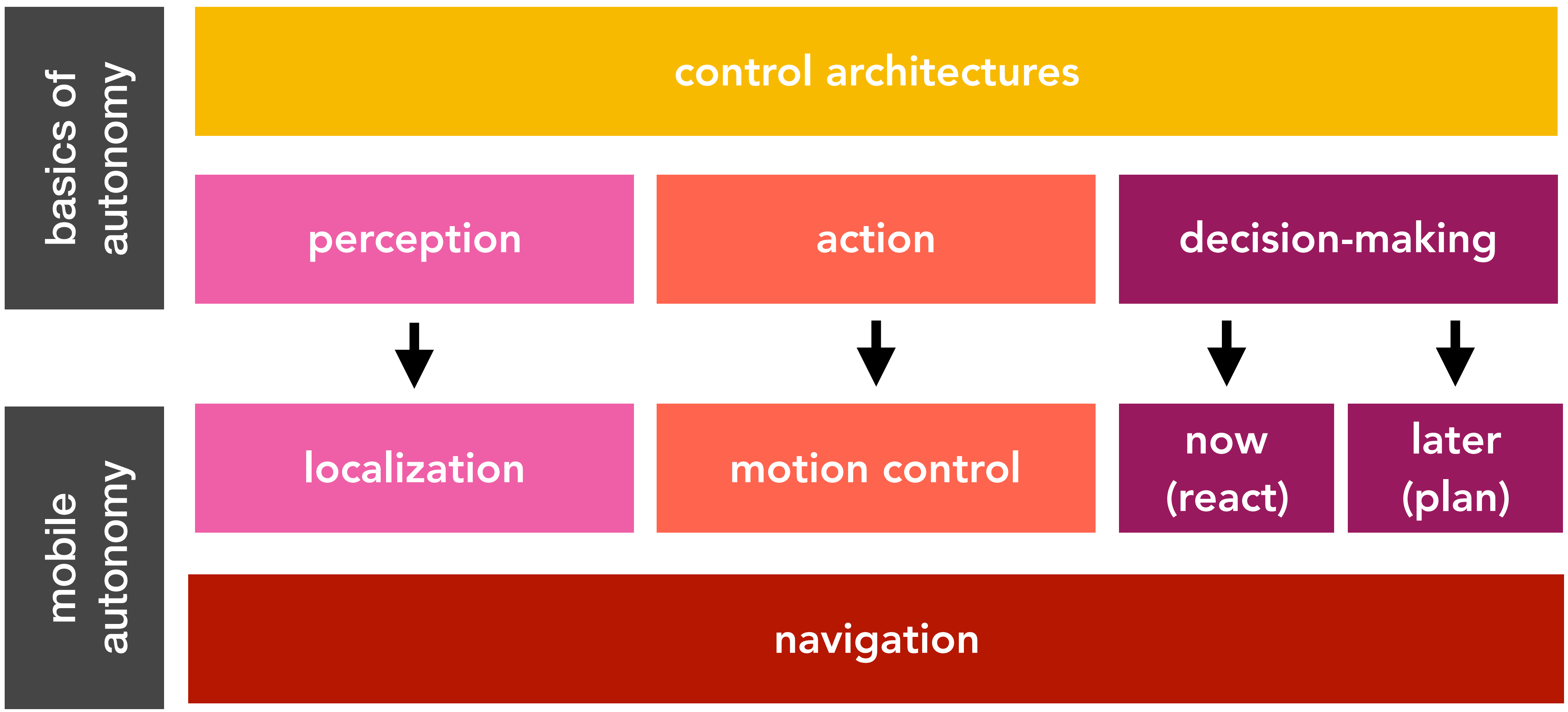
UNIVERSITY OF CAMBRIDGE

In this Lecture

- Introduction to multi-robot systems
- Taxonomy
- Collective movement
    - Flocking (2 example methods)
    - Formations (2 example methods)

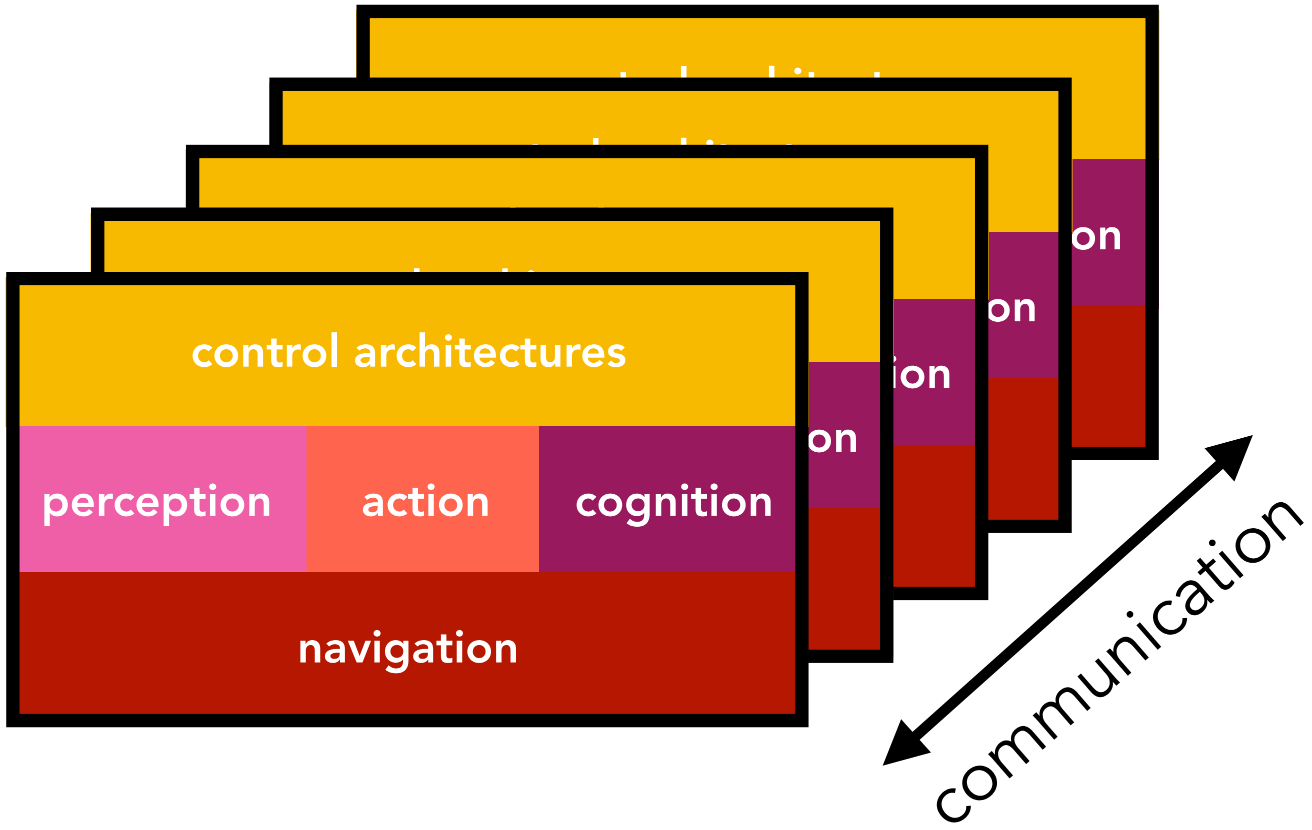
What This Course is About

- Design of this course + focus on **autonomous mobile robots**



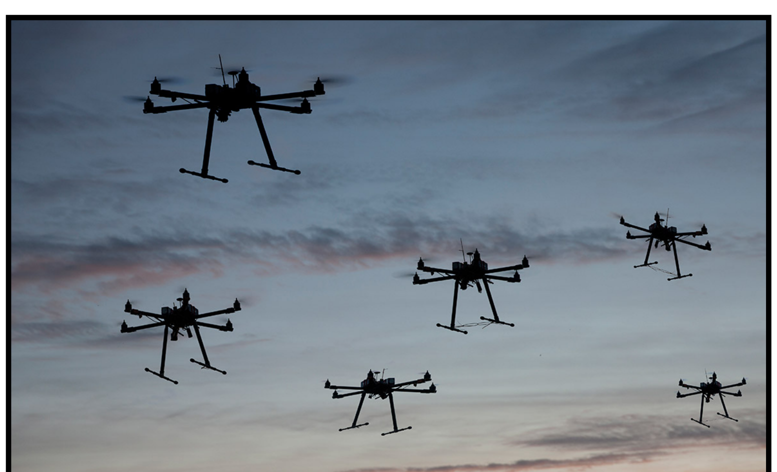
What This Course is About

- Design of this course + focus on mobile robots
- **Multiple** mobile robots → **multi-robot systems**
- Higher-order goals
- Coordination facilitated through communication

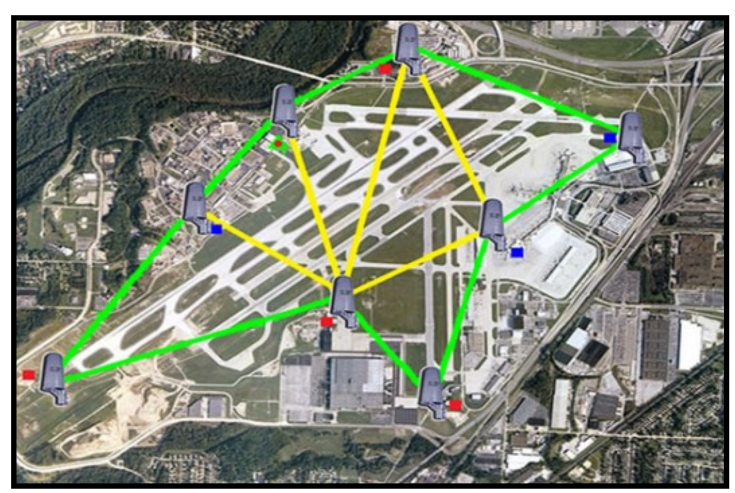


Multi-Robot Systems

- Terms used: robot swarms / robot teams / robot networks
- Why?
  - Distributed nature of many problems
  - Overall performance greater than sum of individual efforts
  - Redundancy and robustness
- Numerous commercial, civil, military applications
- Additional challenge: robot **coordination**



search & rescue



surveillance / monitoring

product pickup / delivery

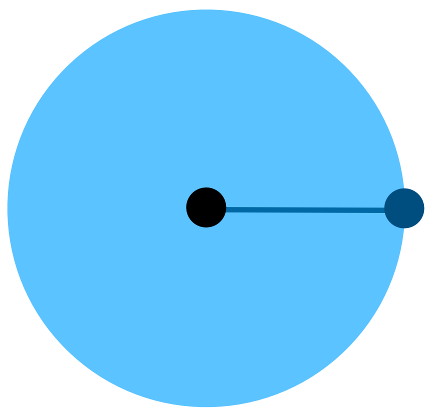
D’Andrea et al., ETHZ

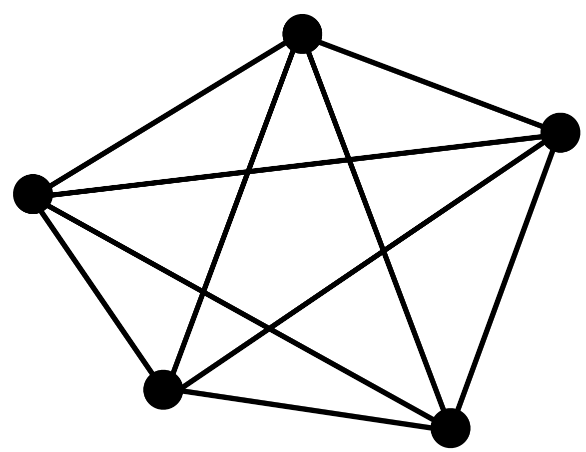
Taxonomy

- **Architecture**: centralized vs. decentralized
  - Centralized: one control/estimation unit communicates with all robots to issue commands; requires synchronized, reliable communication channels; single-point failures
  - Decentralized: scalable, robust to failure; often asynchronous; sub-optimal performance (w.r.t centralized)
- **Communication**: explicit vs. implicit
  - Implicit: observable states (e.g., in the environment); information exchanged through common observations
  - Explicit: unobservable states; need to be communicated explicitly
- **Heterogeneity**: homogenenous vs. heterogeneous
  - Robot teams can leverage inter-robot complementarities

Communication Topologies

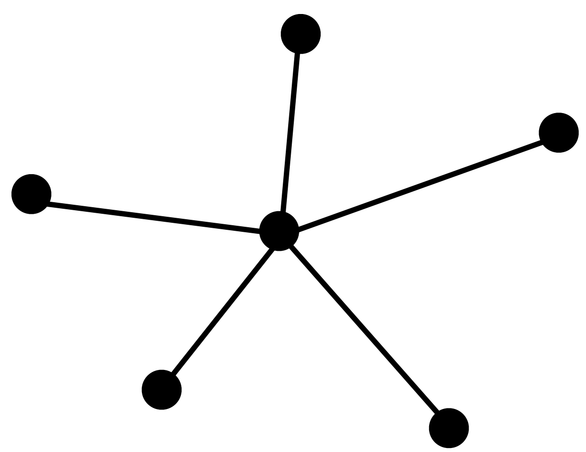
- Robot configurations / topologies are often defined by the maximum range of the available communication module.
- A disc model can be used to represent the communication range (very crude approximation)





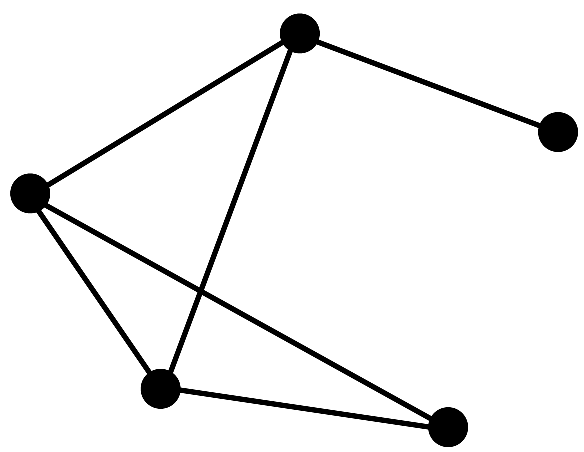
fully connected

centralized / decentralized coordination



star topology

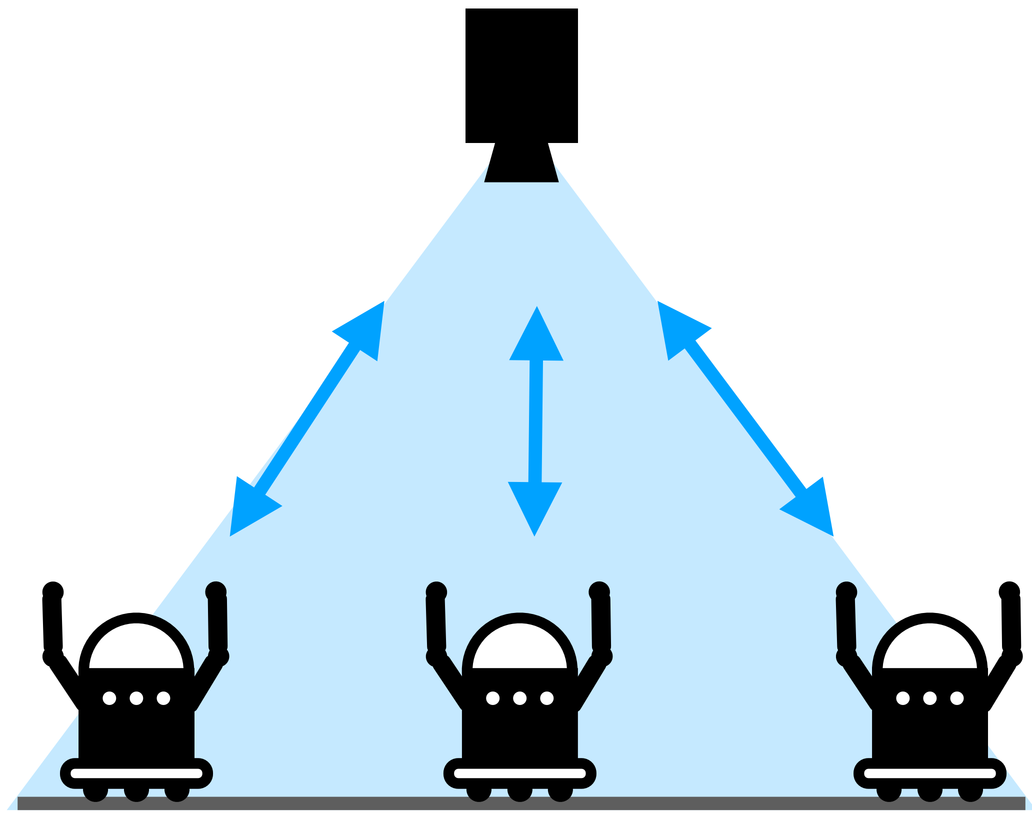
centralized / decentralized coordination



random mesh

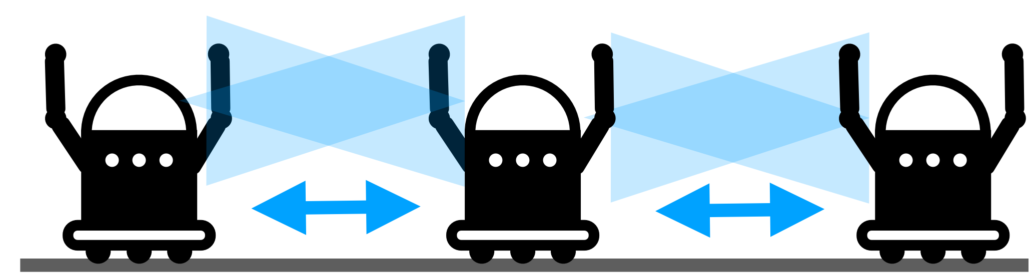
decentralized coordination

Centralization vs Decentralization



centralized

- Centralized control. The controller computes actions based on knowledge of the global state
- Centralized estimation. The unit fuses partial information.



decentralized

- Decentralized control. A robot's control input is based on interactions with its neighbors.
- Decentralized estimation. The robot's estimate is based on relative observations.

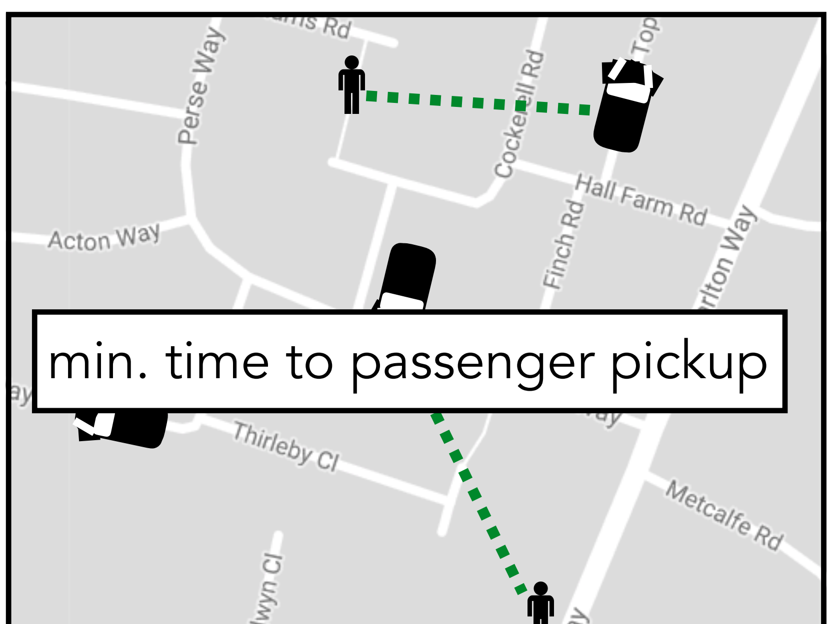
Centralization vs Decentralization

min. time to product dispatch

automated warehouses

max. area coverage / min. time to target

search & rescue / surveillance



min. time to passenger pickup

automated mobility-on-demand

max. throughput / min. collision probability

connected autonomous vehicles

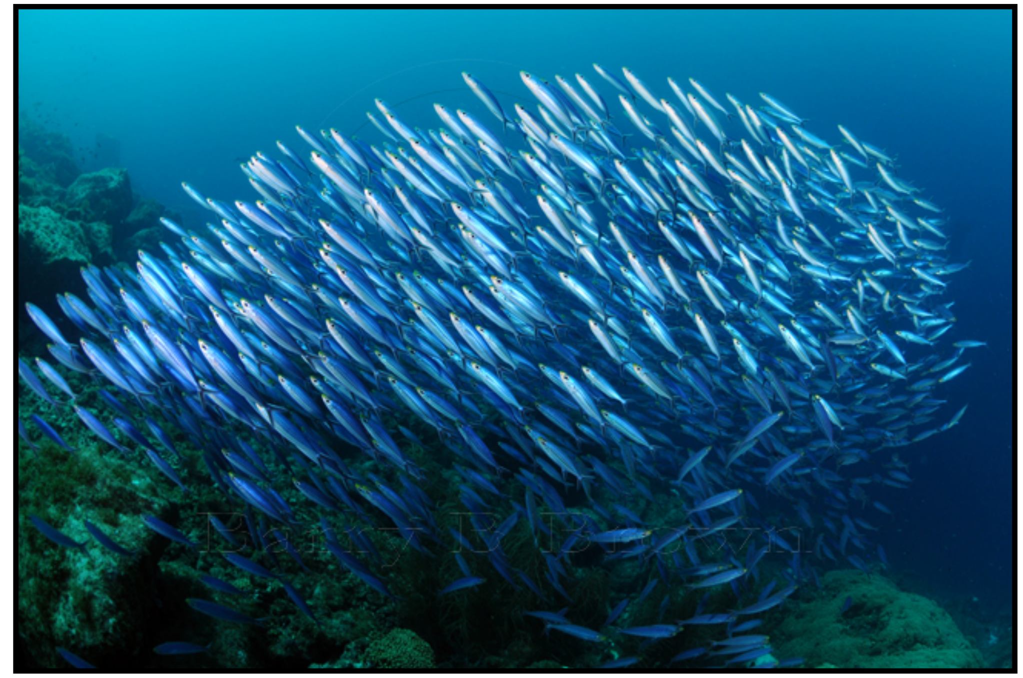
Decentralization

- Goal: Achieve similar (or same) performance as would be achievable with an ideal, centralized system.
- Challenges:
  - Communication: delays and overhead
  - Input: asynchronous; with rumor propagation
  - Sub-optimality with respect to the centralized solution
- Advantages:
  - No single-point failure
  - Can converge to optimum as time progresses
  - 'Any-comm' algorithms exist (graceful degradation under failing comms)
  - 'Any-time' algorithms exist (continuous improvement of solution)

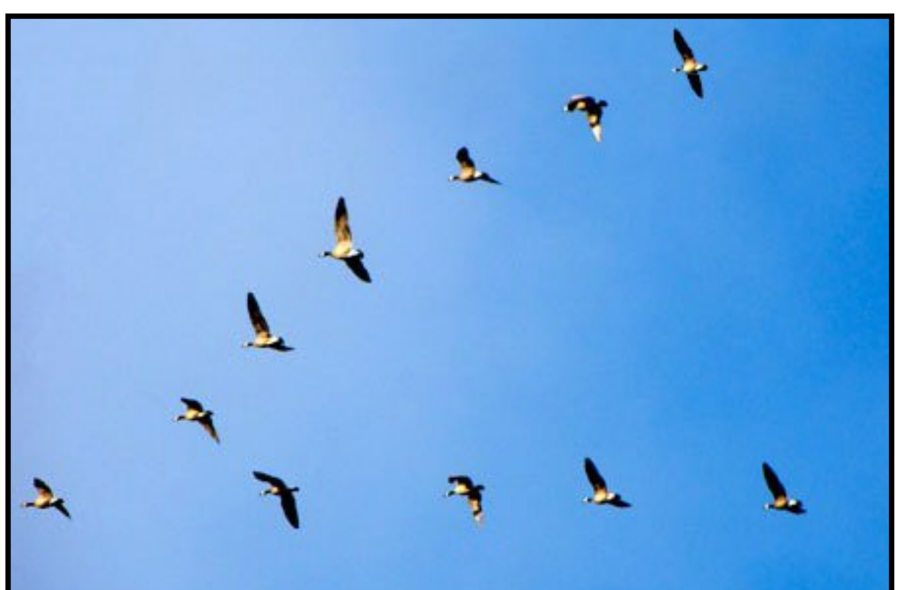
Collective Movement

In nature:

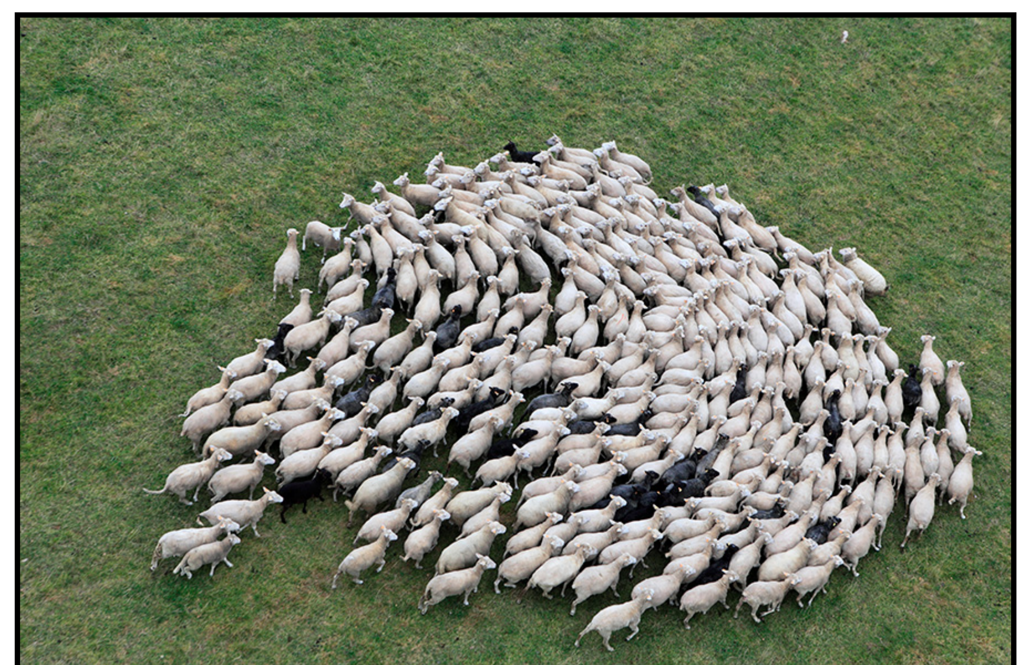
flock of birds



school of fish



flock of geese



herd of mammals

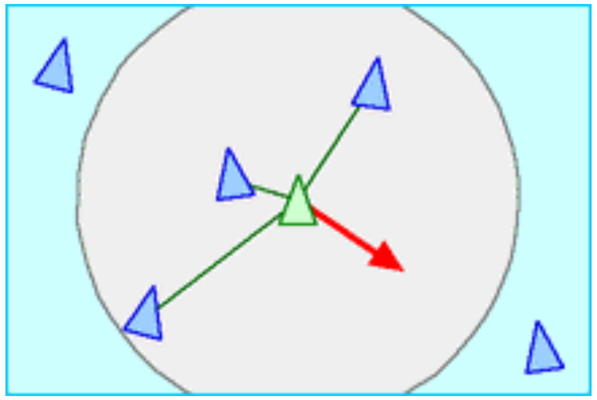
Collective Movement

- Collective movement in natural societies:
  - Properties: no collisions; no apparent leader; tolerance of loss or gain of group member; coalescing and splitting; reactivity to obstacles; different species have different flocking characteristics
  - Benefits: energy saving (e.g., geese extend flight range by 70%); signs of better navigation accuracy
- Engineered flocking - decentralized:
  - Reynolds’ virtual agents (Boids)
  - Graph-based distributed control for spatial consensus
- Engineered flocking - centralized:
  - E.g.: Controls for each robot computed off-board, in the cloud

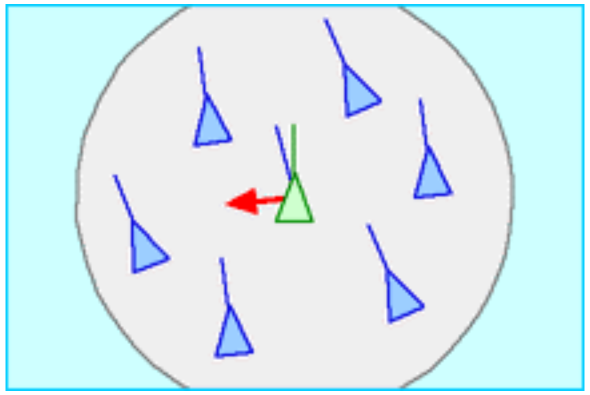
Intel drone swarm

Flocking with Boids

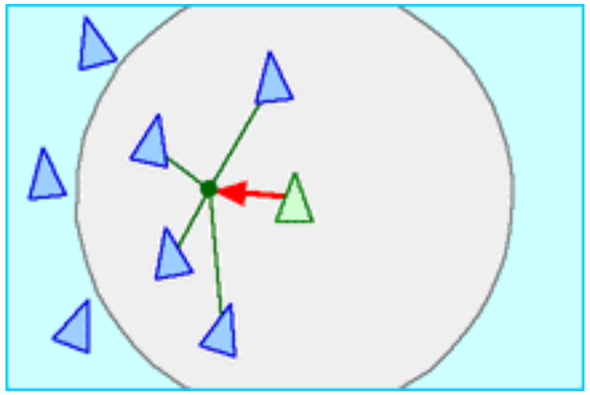
- In 1986, Craig Reynolds (computer animator) wanted to create a computationally efficient method to animate flocks
- Goal: $O(N)$; current best was $O(N^2)$



separation



alignment

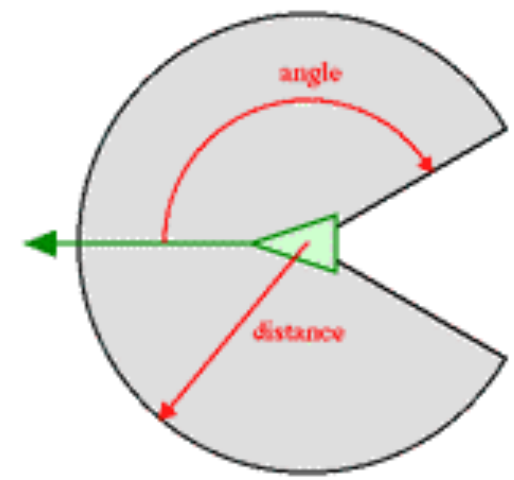


cohesion

- A boid reacts only to its neighbors
- Neighborhood defined by distance and angle (region of influence)
- Each boid follows 3 steering rules based on positions and velocities of neighbors. Recipe: compute 3 components, then combine to form motion (vector)

Flocking with Boids

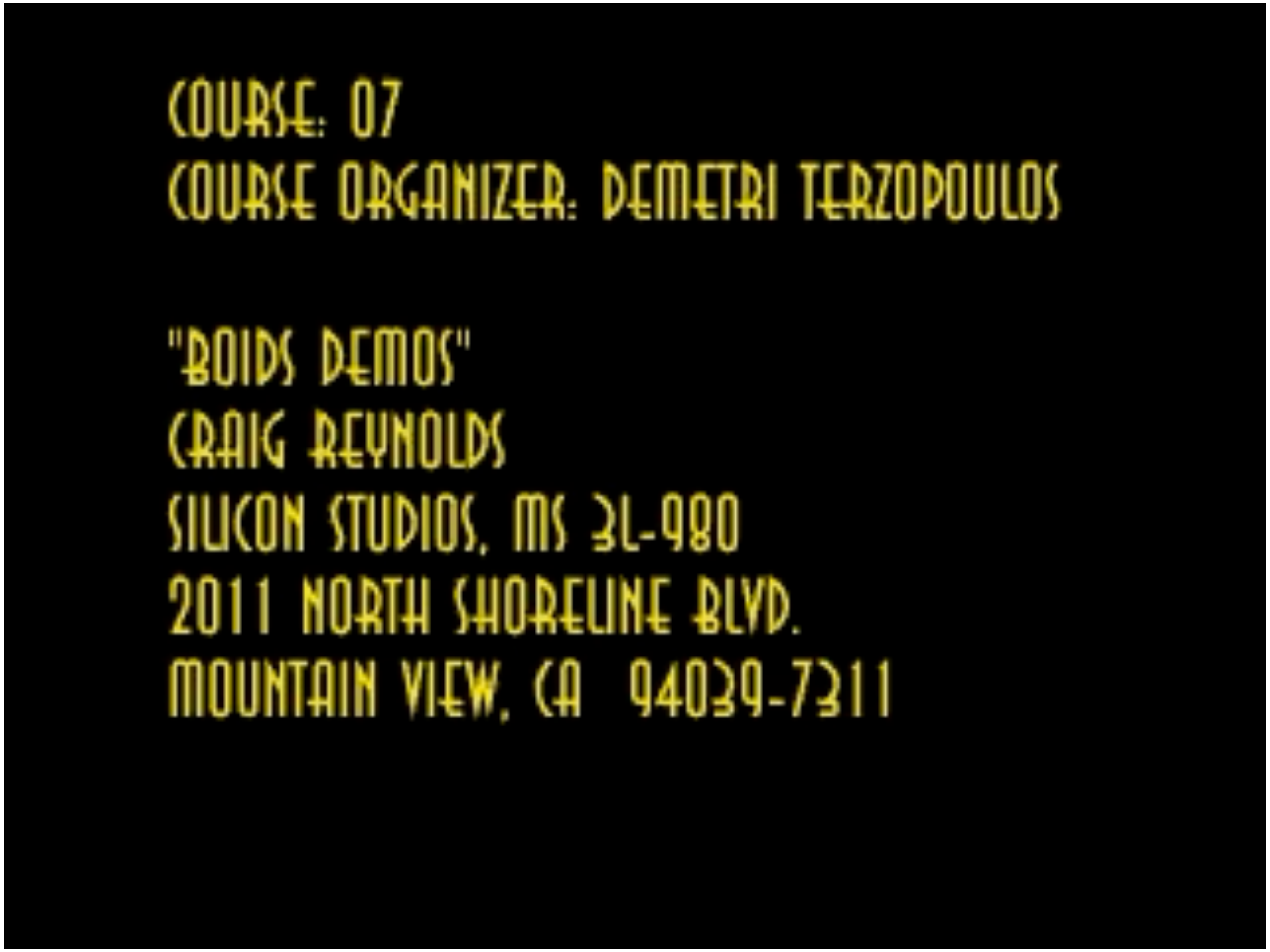
- Sensory system: idealized, but local:
  - almost omni-directional
  - no delays (in sensing)
  - no noise (in range and bearing)



2D representation of boid neighborhood

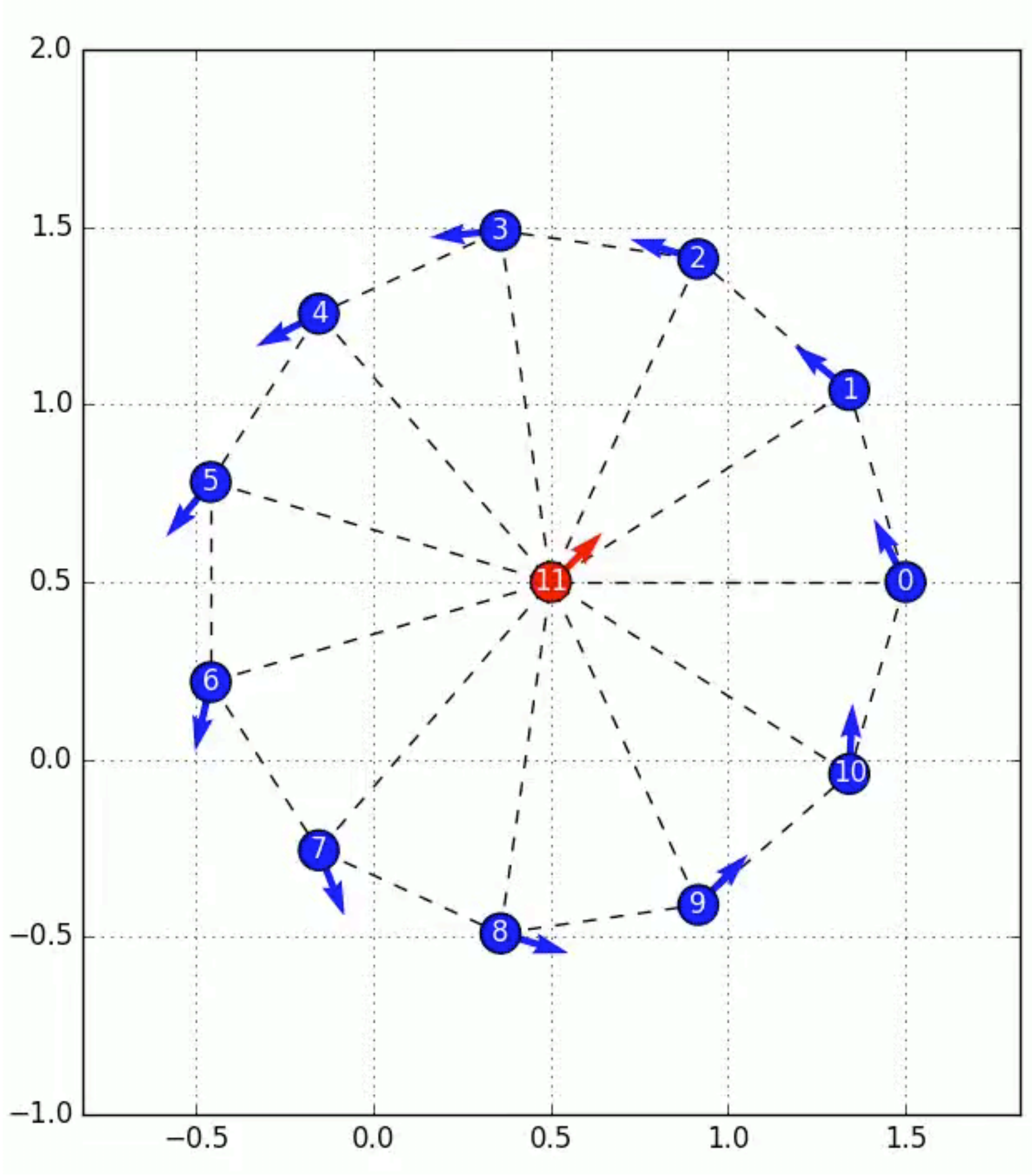
- Behavior-based with priorities (cf Brooks):
  - Low priority acceleration request towards a point or in a direction (to direct flock)
  - Highest priority to obstacle avoidance ('steer-to-avoid' with a different sensory system)

Flocking with Boids



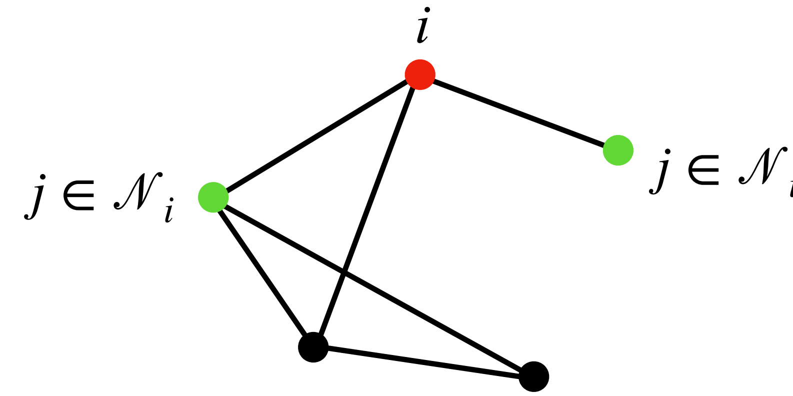
more info on http://www.red3d.com/cwr/boids/

Flocking with Consensus



1 leader robot; robots apply consensus algorithm to agree on heading

The Consensus Algorithm



- Aim of consensus:
  - Reach decentralized agreement
  - Purely based on local interactions
- Consensus
  - Based on a graph-topological definition of multi-robot system
  - Applications: motion coordination; cooperative estimation; synchronization
- Discrete time consensus update:

$$x_i[t+1] = \frac{1}{|\mathcal{N}_i| + 1}\left(x_i[t] + \sum_{j \in \mathcal{N}_i} x_j[t]\right)$$

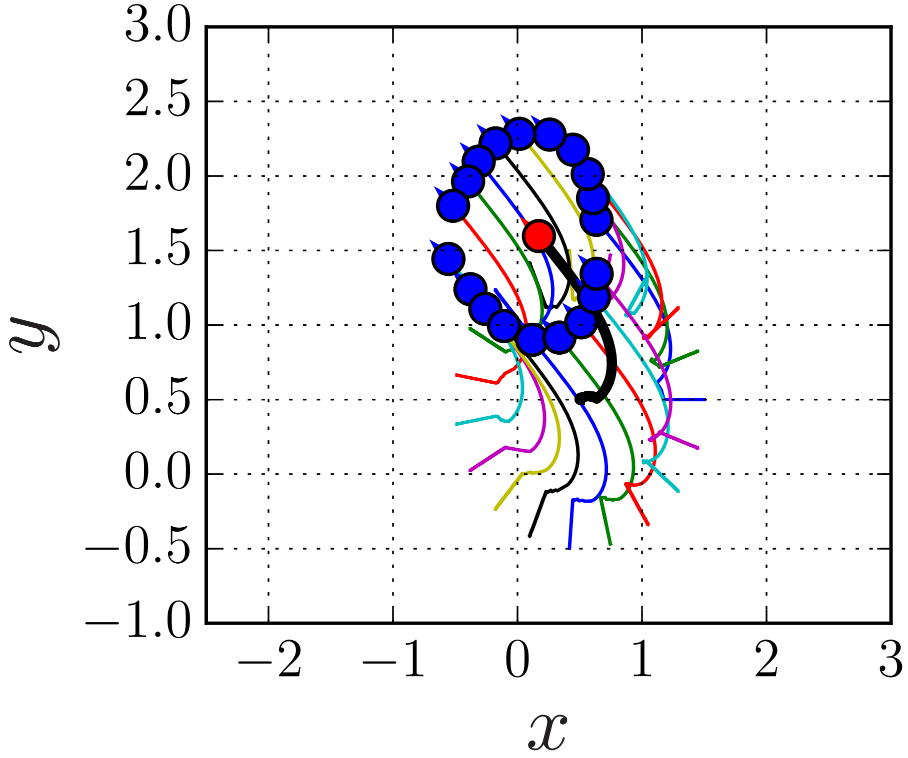
- Consensus outcome:
  - All robots converge to average of initial values (convergence rate is exponential):

$$t \to \infty, \qquad x_i[t] = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} x_i[0]$$

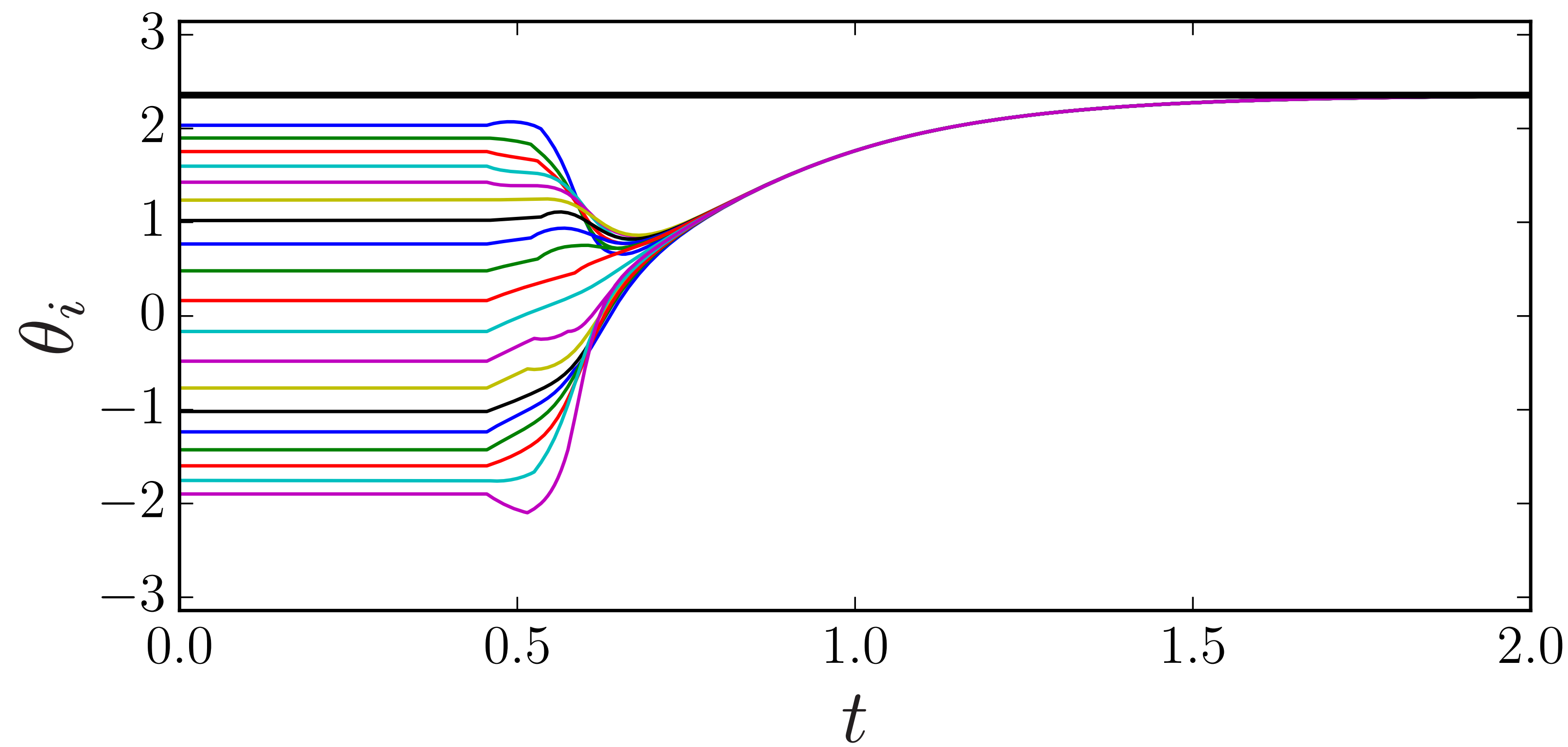
Flocking with Consensus

Holonomic robot: $\dot{\mathbf{x}} = \mathbf{u}$ with $\mathbf{x}_i = [x_i, y_i]$

Consensus on heading θ_i



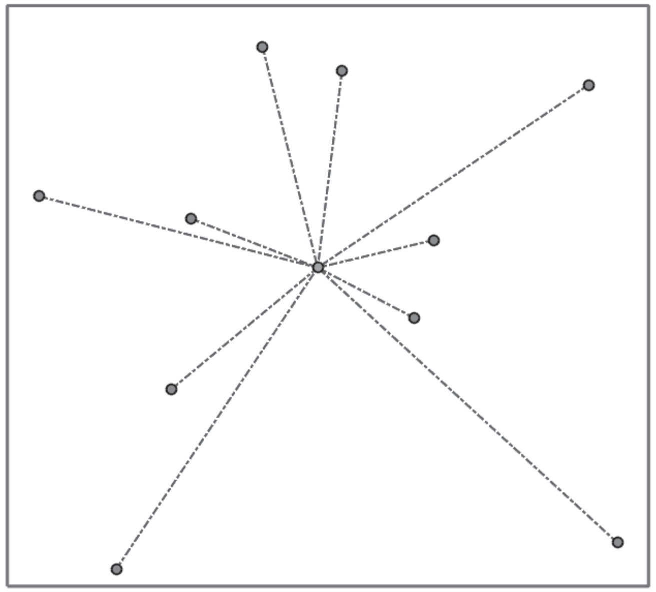
robot trajectories



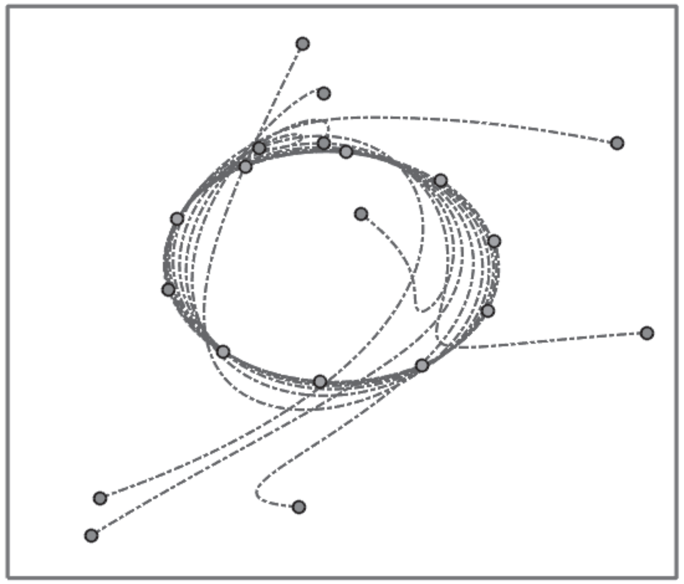
heading as a function of time

Note: Collision avoidance and connectivity maintenance are needed *in addition* to agreement on direction of motion.

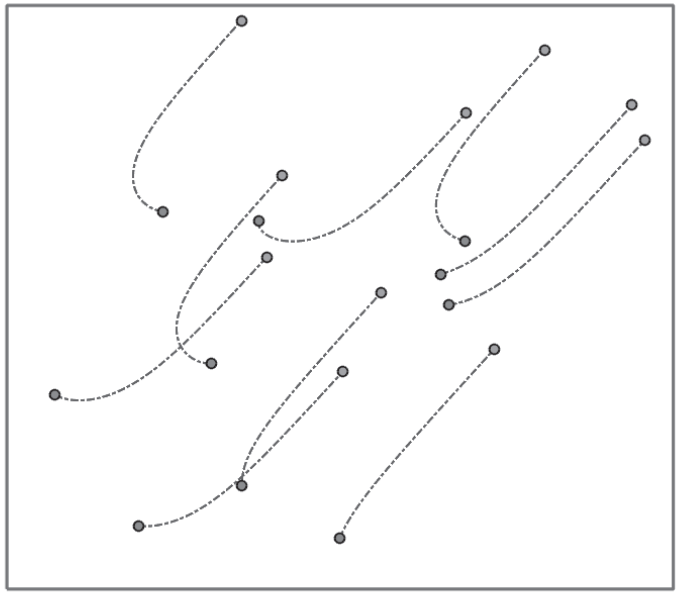
Other Consensus Applications



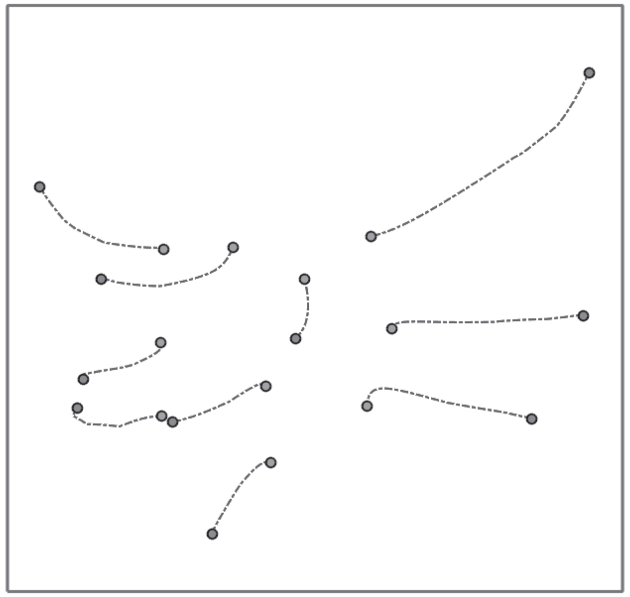
rendezvous



cyclic pursuit



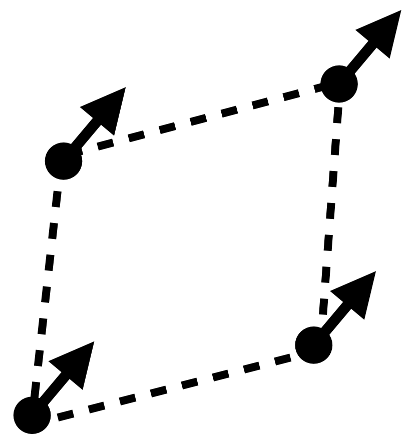
flocking



configuration

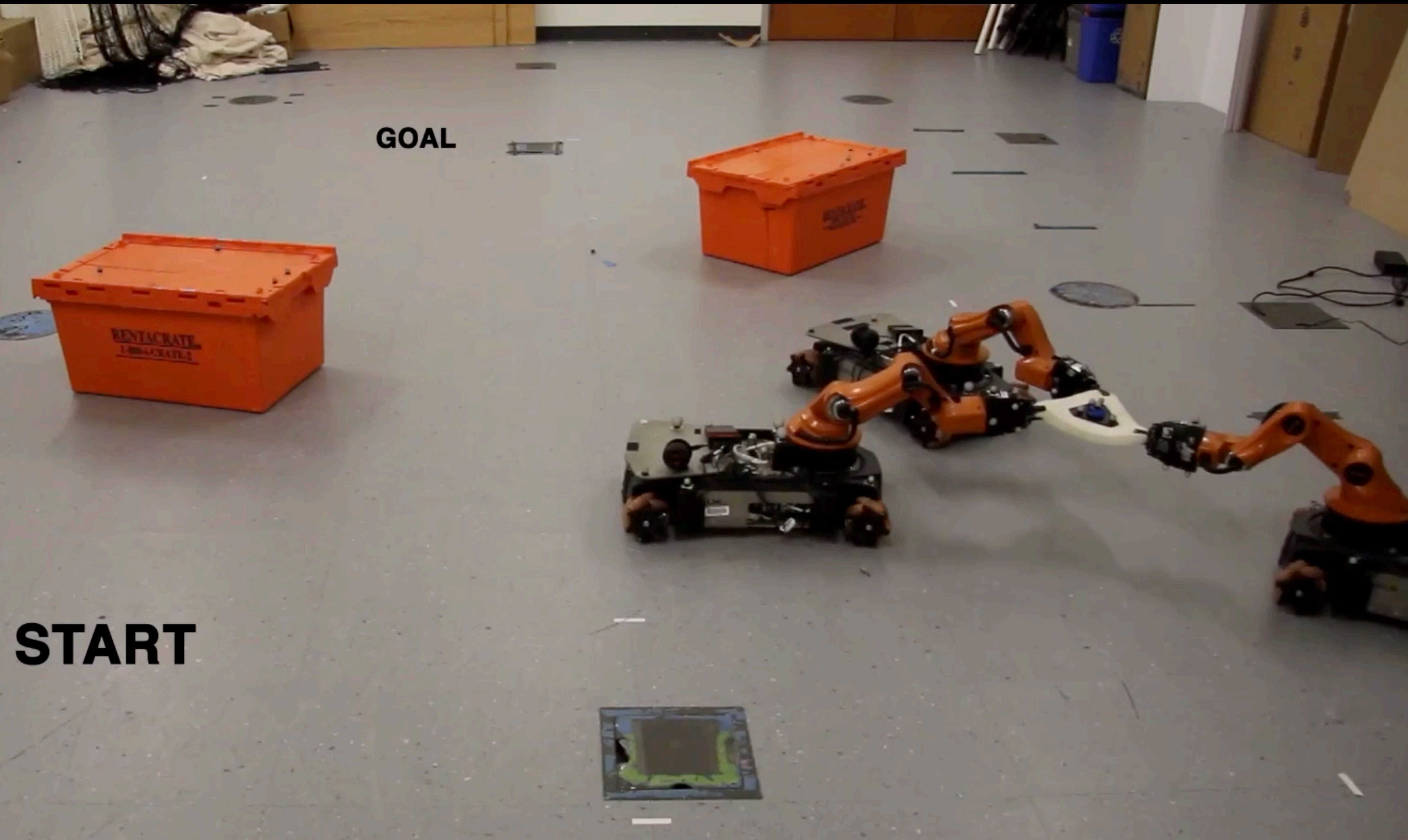
Formation Control

- Formations (versus flocks): ***specific*** geometric configurations
- Some applications benefit from multiple robots navigating as a group:
  - Transport (vehicle formations; platooning); scout platoons for reconnaissance and search; environmental monitoring; lawn mowing
- Generally required: information on state (e.g. *pose*) of all robots
- Challenges:
  - Noisy sensors; delay in sensing / actuation
  - Anonymous robots (no IDs)
  - Non-holonomicity



e.g.: diamond formation

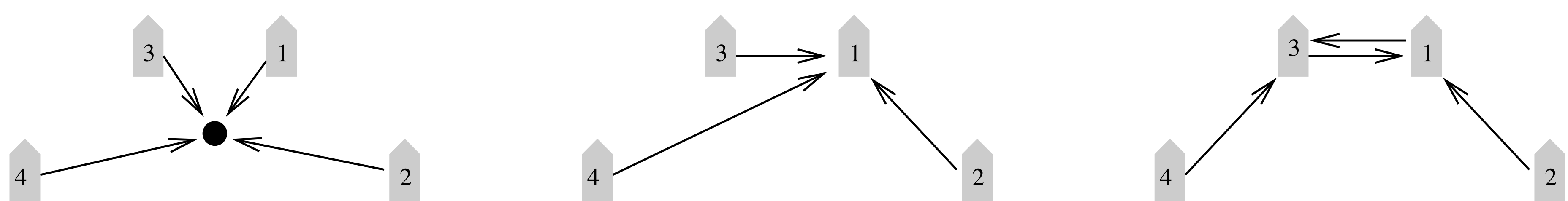
- Variants:
  - Behavior-based (Balch et al., 1999) *(recall: reactive control paradigm)*
  - Closed-loop control (Das et al., 2002) *(recall: error-based control paradigm)*



Rus et al., MIT

Formation Control

- Referencing schemes:
  - **Unit-center-referenced**: obtained by averaging positions of all robots. A robot determines its position relative to this center.
  - **Leader-referenced**: robots determine pose relative to leader, which does not attempt to maintain the formation.
  - **Neighbor-referenced**: robots attempt to maintain relative pose to one (or a select group) of neigboring robots.



unit-center leader neighbor

- How is positioning information obtained?
  - Each robot estimates its own pose, and communicates this to other robots.
  - Or: robots estimate their relative pose via sensor observations

*image credit: Balch 1999

Behavior-Based Formation Control

- Method based on 'Motor-Schema' [Balch, Arkin; 1999]
- Different motor schemes are defined; each generates a ***vector representing a behavioral response*** (direction and magnitude of movement) as a function of sensor stimuli
- A gain value is used to attribute ***relative importance*** of schemes

Parameter	Value	Units
avoid-static-obstacle		
gain	1.5	
sphere of influence	50	meters
minimum range	5	meters
avoid-robot		
gain	2.0	
sphere of influence	20	meters
minimum range	5	meters
move-to-goal		
gain	0.8	
noise		
gain	0.1	
persistence	6	time steps
maintain-formation		
gain	1.0	
desired spacing	50	meters
controlled zone radius	25	meters
dead zone radius	0	meters

*image credit: Balch 1999

Behavior-Based Formation Control

Motor schemas:

1. **move-to-goal**: attract to goal with variable gain

$V_{magnitude}$ = adjustable gain value
$V_{direction}$ = in direction towards perceived goal

2. **avoid-obstacle / robot**: repel from object with variable gain and sphere of influence.

$$O_{magnitude} = \begin{cases} 0 & for \;\; d > S \\ \frac{S-d}{S-R} * G & for \;\; R < d \leq S \\ \infty & for \;\; d \leq R \end{cases}$$

where:

S = Adjustable Sphere of Influence (radial extent of force from the center of the obstacle)
R = Radius of obstacle
G = Adjustable Gain
d = Distance of robot to center of obstacle

$O_{direction}$ = along a line from robot to center of obstacle moving away from obstacle

3. **noise**: random wander with variable gain and persistence; used to overcome local maxima, cycles, and for exploration.

$N_{magnitude}$ = Adjustable gain value
$N_{direction}$ = Random direction that persists for $N_{persistence}$ steps ($N_{persistence}$ is adjustable)

Behavior-Based Formation Control

4. **maintain-formation**: decomposed into two parts

maintain-formation-speed

$$V_{speed} = R_{mag} + K \times \delta_{speed}$$

maintain-formation-steer

$$H_{desired} = F_{dir} - \delta_{heading}$$

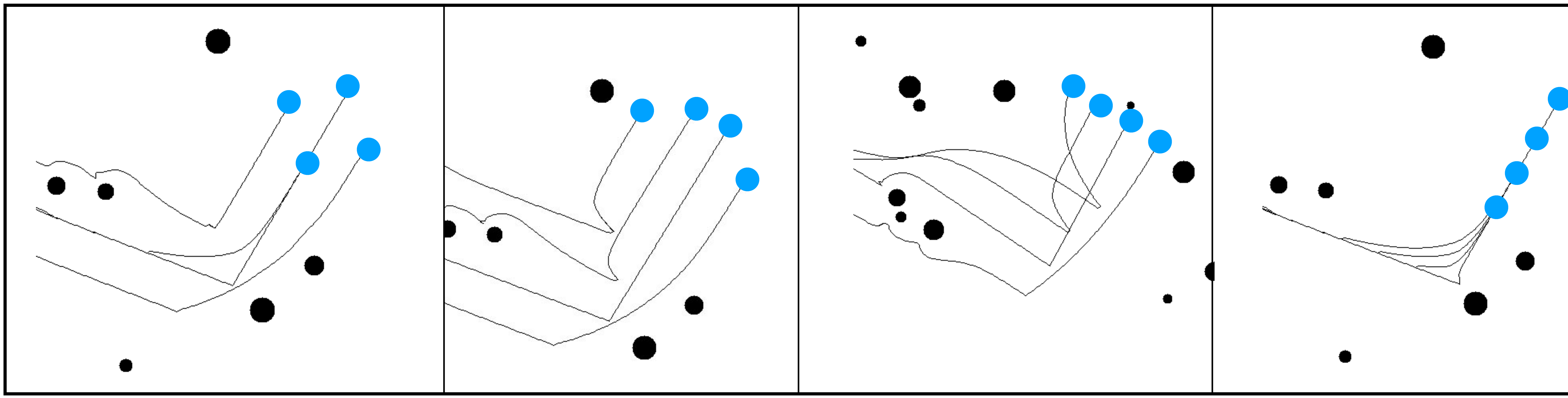
$$V_{steer} = H_{desired} - R_{dir}$$

- R_{pos}, R_{dir} the robot's present position and heading.
- R_{mag}, the robot's present speed.
- F_{pos}, the robot's proper position in formation.
- F_{dir}, the direction of the formation's movement; towards the next navigational waypoint.
- F_{axis}, the formation's axis, a ray passing through F_{pos} in the F_{dir} direction.
- $H_{desired}$, desired heading, a computed heading that will move the robot into formation.
- $\delta_{heading}$, the computed heading correction.
- δ_{speed}, the computed speed correction.
- V_{steer}, steer vote, representing the directional output of the motor behavior, sent to the steering arbiter.
- V_{speed}, speed vote, the speed output of the motor behavior, sent to the speed arbiter.

[Balch, Arkin; 1999]

Behavior-Based Formation Control

Example of results, for leader-referenced scheme [Balch '99]:



diamond wedge line column

Assumptions:

- fully networked system; robots have IDs (non-anonymous)
- robot positioning with little noise and delay
- straight-forward implementation for **holonomic (point-) robots**

*image credit: Balch 1999

Formation Control

- Non-holonomic robots:
  - Proposed method: *fore-aft / side-side corrections*
  - Separate motor behaviors a generated for steering / speed. **Arbiters** accept votes from the motor schemas to compute speed / steering values.
  - Combined with a rule-based program that selects final speed / steering value.
- Issues:
  - Behavior-based methods have no guarantees:
    - Convergence to desired formation? Stability of formation?
    - **Need for more principled approaches**
- Introduction of control-theoretic principles to provide these guarantees
  - One of the first such approaches presented by Das et al., 2002

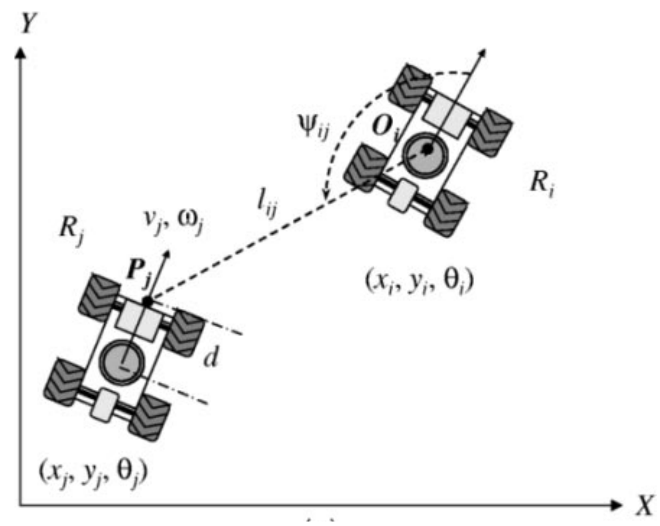


*image credit: Das 2002

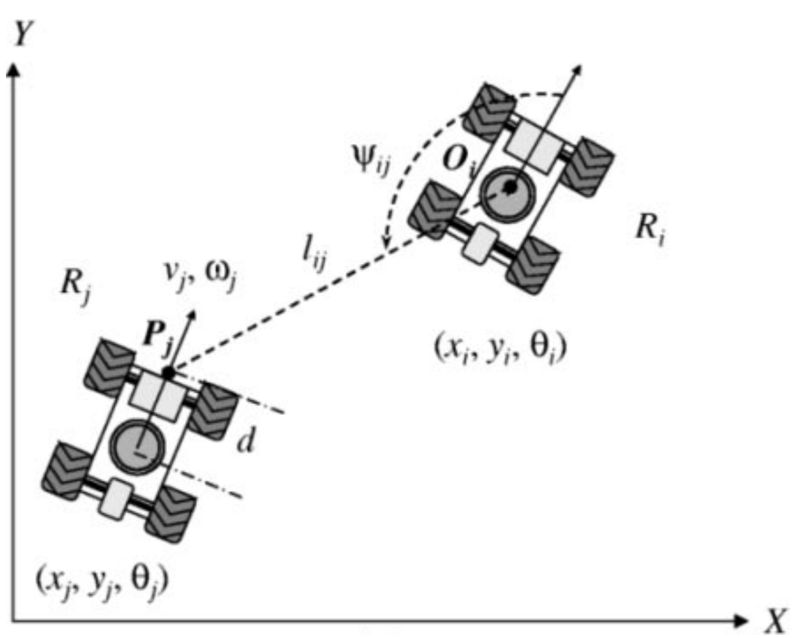
Closed-Loop Control for Formations

- Method based on feedback linearization [Das et al., 2002]
- Basic case: leader-referenced control based on **separation distance** and **relative bearing**: $\mathbf{z}_{ij} = [l_{ij}, \psi_{ij}]^\top$

  Control input: $\mathbf{u}_j = [v_j, \omega_j]^\top$ (forwards and rotational velocities)

leader, robot $\boldsymbol{i}$

follower, robot $\boldsymbol{j}$



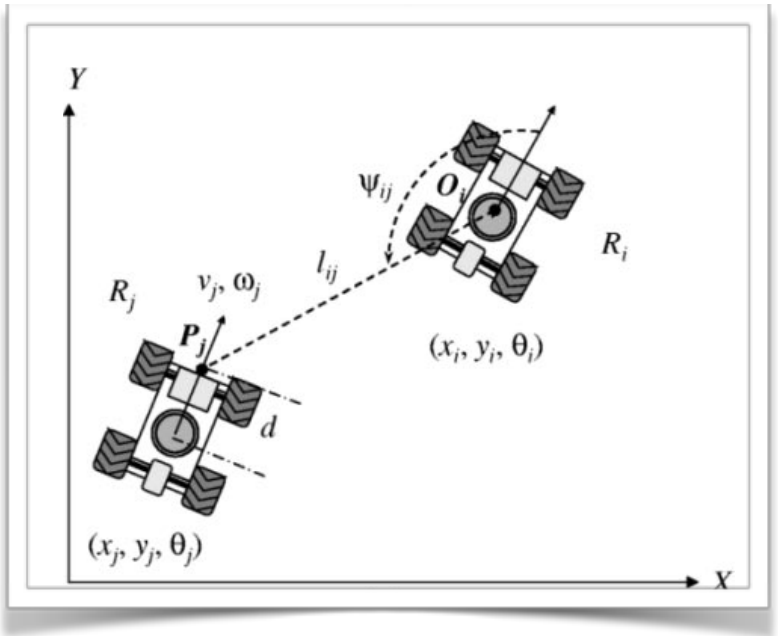
Aim: Find $\mathbf{u}_j$ such that desired separation l_{ij}^d and desired bearing ψ_{ij}^d are reached, and stably maintained.

Closed-Loop Control for Formations

Dynamical system model: $\dot{\mathbf{z}}_{ij} = G\,\mathbf{u}_j + F\,\mathbf{u}_i$

with:

$$G = \begin{bmatrix} \cos\gamma_{ij} & d\sin\gamma_{ij} \\ \frac{-\sin\gamma_{ij}}{l_{ij}} & \frac{d\cos\gamma_{ij}}{l_{ij}} \end{bmatrix} \qquad F = \begin{bmatrix} -\cos\psi_{ij} & 0 \\ \frac{\sin\psi_{ij}}{l_{ij}} & -1 \end{bmatrix}$$



where relative orientation is: $\beta_{ij} = \theta_i - \theta_j$ and $\gamma_{ij} = \beta_{ij} + \psi_{ij}$

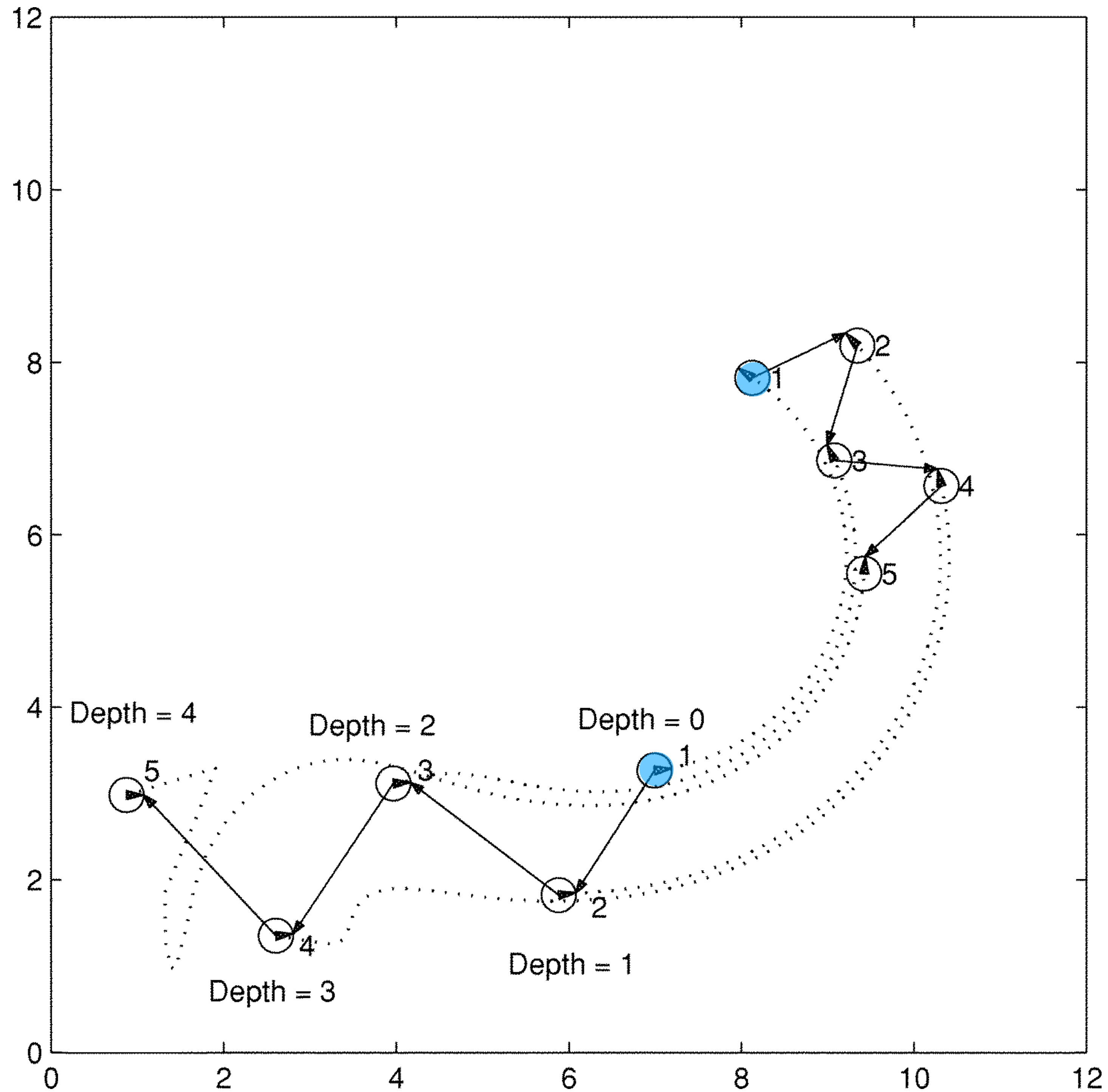
Control: $\mathbf{u}_j = G^{-1}\left(\mathbf{k}(\mathbf{z}^d_{ij} - \mathbf{z}_{ij}) - F\mathbf{u}_i\right)$

which satisfies: $\dot{\mathbf{z}}_{ij} = \mathbf{k}(\mathbf{z}^d_{ij} - \mathbf{z}_{ij})$

closed-loop linearized system

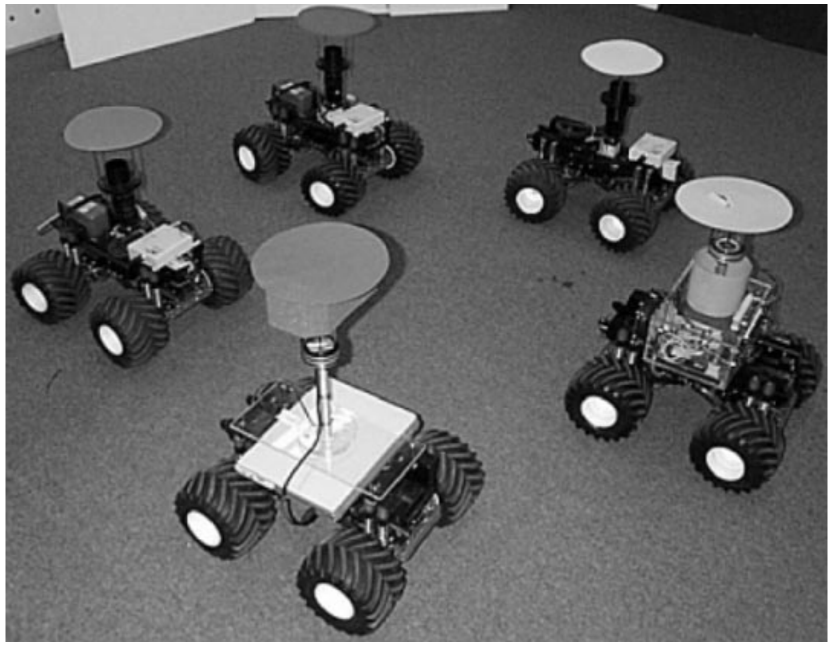
This guarantees convergence to desired relative state $\mathbf{z}^d_{ij}$ (Stability is proven in paper.)

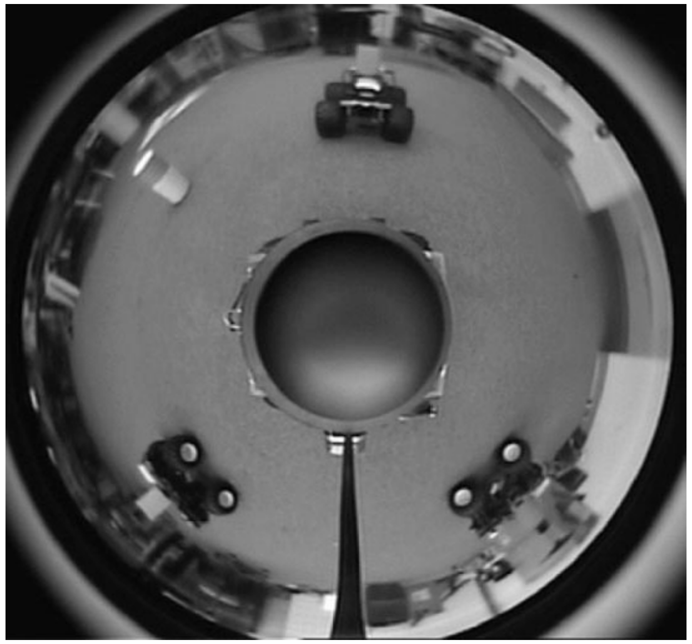
Closed-Loop Control for Formations



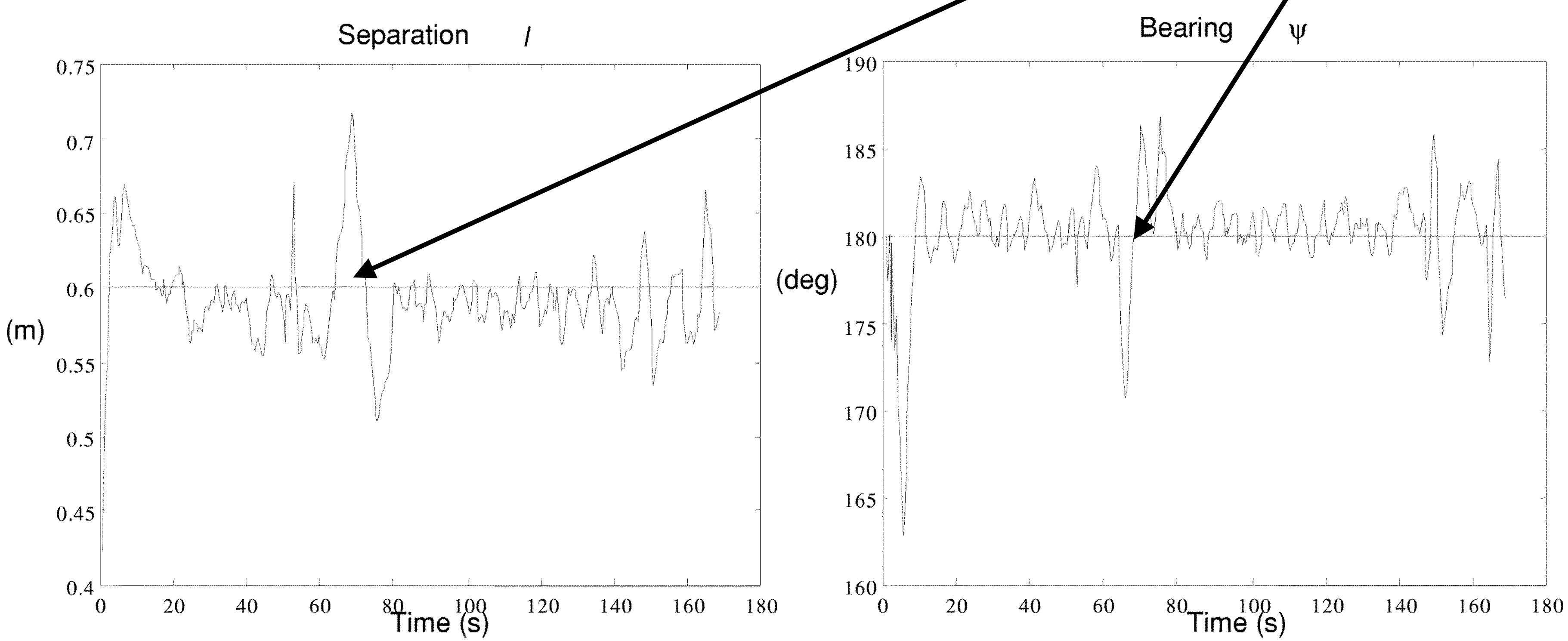
Closed-Loop Control for Formations

Four robots with omnidirectional cameras:



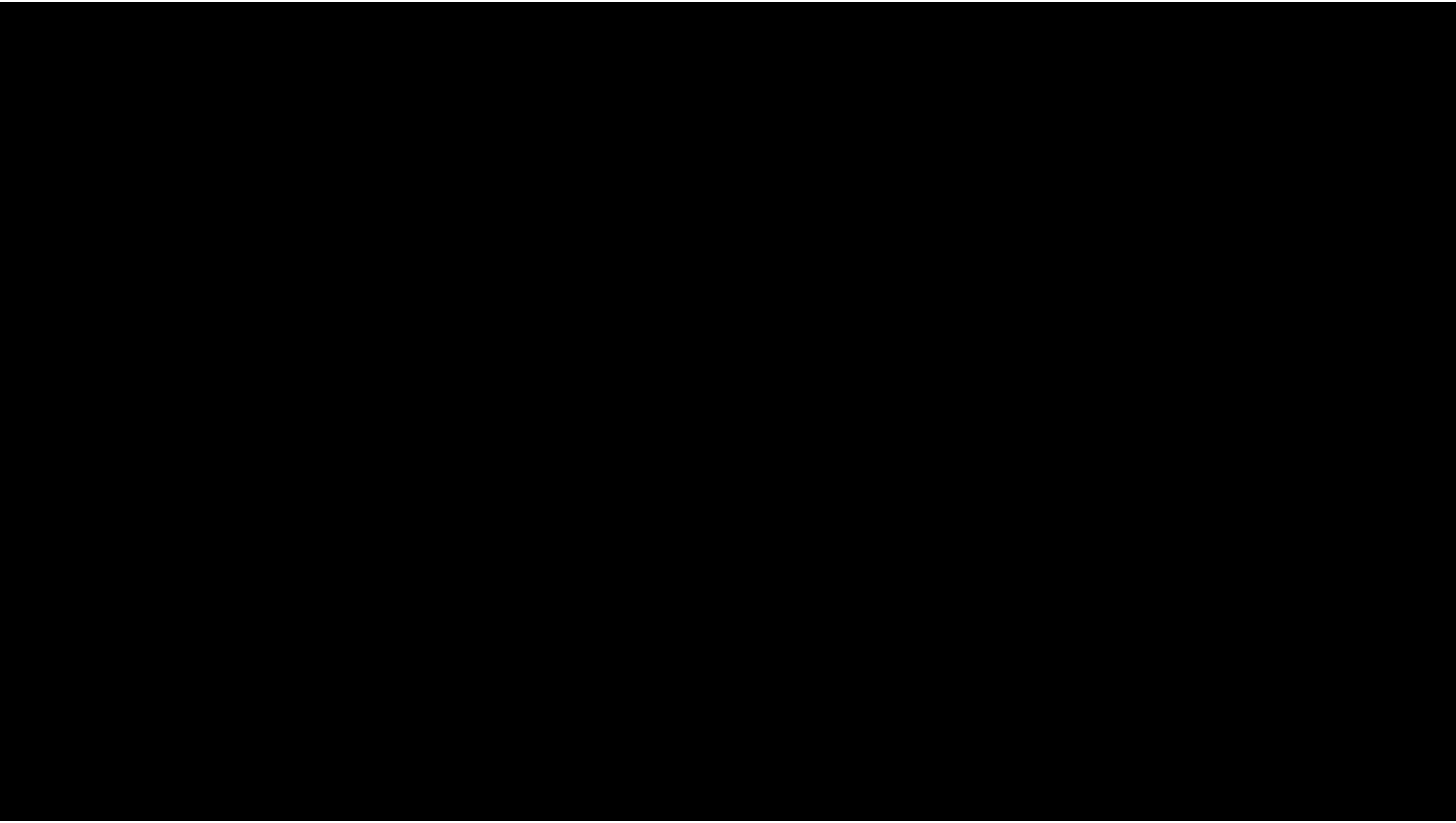


manually restrained
follower



[Das et al., 2002]

A Figure 8 with Range & Bearing



*movie credit: Gowal, Martinoli, EPFL

Further Reading

Seminal papers:

- Behavior-Based Formation Control for Multi-Robot Teams; T Balch, R Arkin; 1999

Seminal papers (advanced):

- A Vision-Based Formation Control Framework; A K. Das, R Fierro, R. V Kumar, J P. Ostrowski, J Spletzer, C J. Taylor; 2002
- Consensus and cooperation in networked multi-agent systems; Olfati-Saber, Fax, Murray; 2007