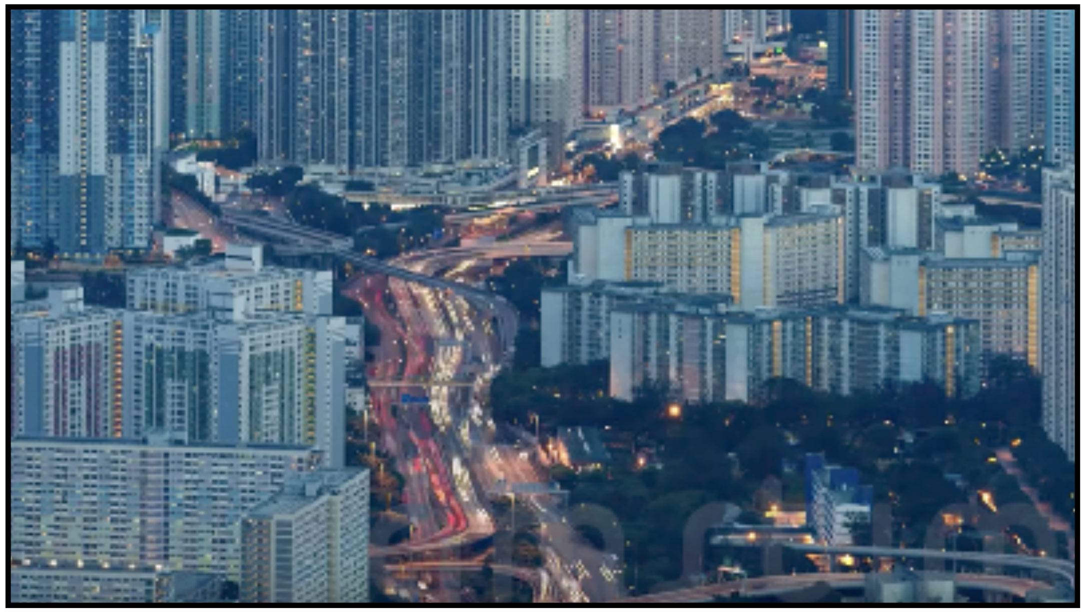
Mobile and Sensor Systems

Lecture 10: Mobile Robots, Control, and Coordination in Robot Teams

Dr. Amanda Prorok

UNIVERSITY OF CAMBRIDGE

Robots and Mobile Systems



smart infrastructure / mobility-on-demand

connected vehicles / automated highways

drone swarms / surveillance

truck platoons / long-haul transport

In this Lecture

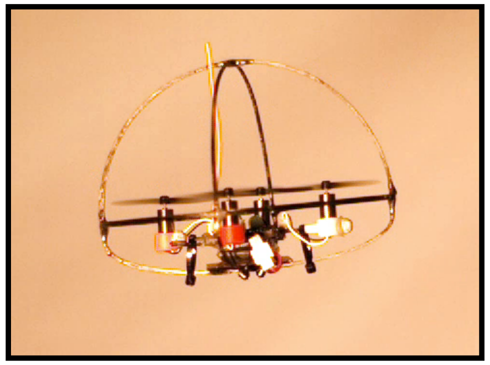
- Overview of mobile robot control
  - Basic principles of kinematics
  - Overview of classical control architectures
- Coordination in systems with multiple robots
  - Taxonomy
  - Distributed estimation
  - Distributed control

Autonomous Robots

- What is a robot?

microrobots
[Wood, Harvard]

self-foldable / self-actuated
[Sung and Rus; MIT]



lightweight aerial robots
[Kumar et al.; UPenn]

consumer-grade drones

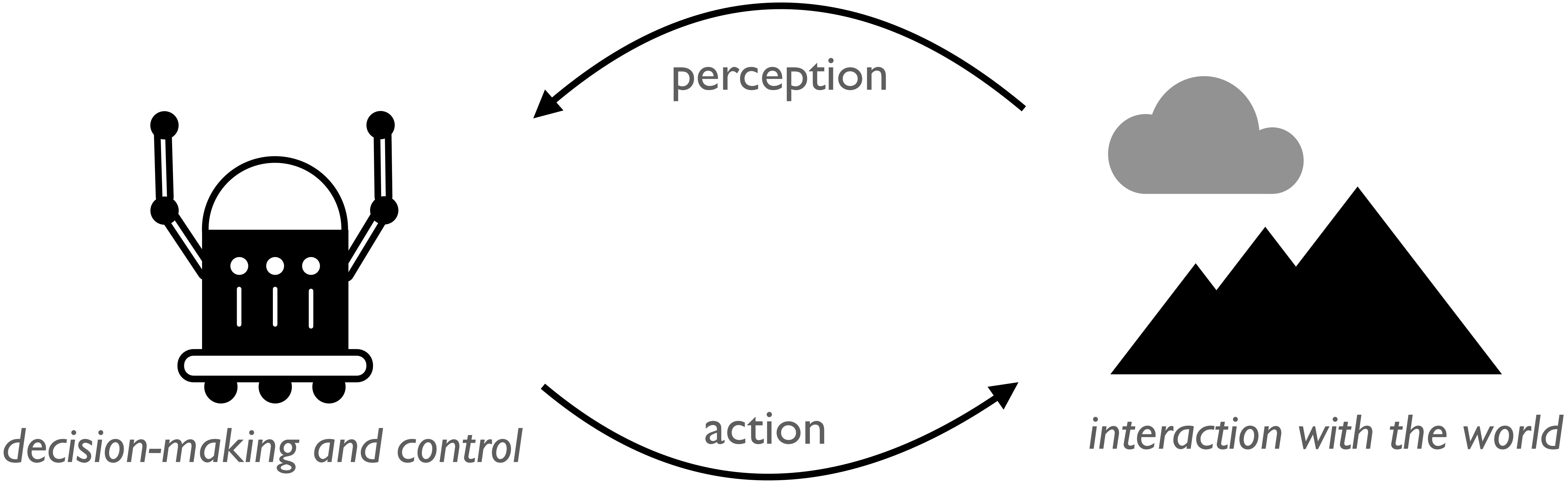
autonomous vehicles
[Google]

- Challenges:
  - How to model and perceive the world?
  - How to process information and exert control?
  - How to reason and plan in the face of uncertainty?

UNIVERSITY OF CAMBRIDGE

Perception-Action Loop

- Basic building block of autonomy!



Three main variants:
1. Reactive (e.g., nonlinear transform of sensor readings)
2. Reactive + memory (eg., filter, state variables)
3. Deliberative (e.g., planning)

Sensors for Robots

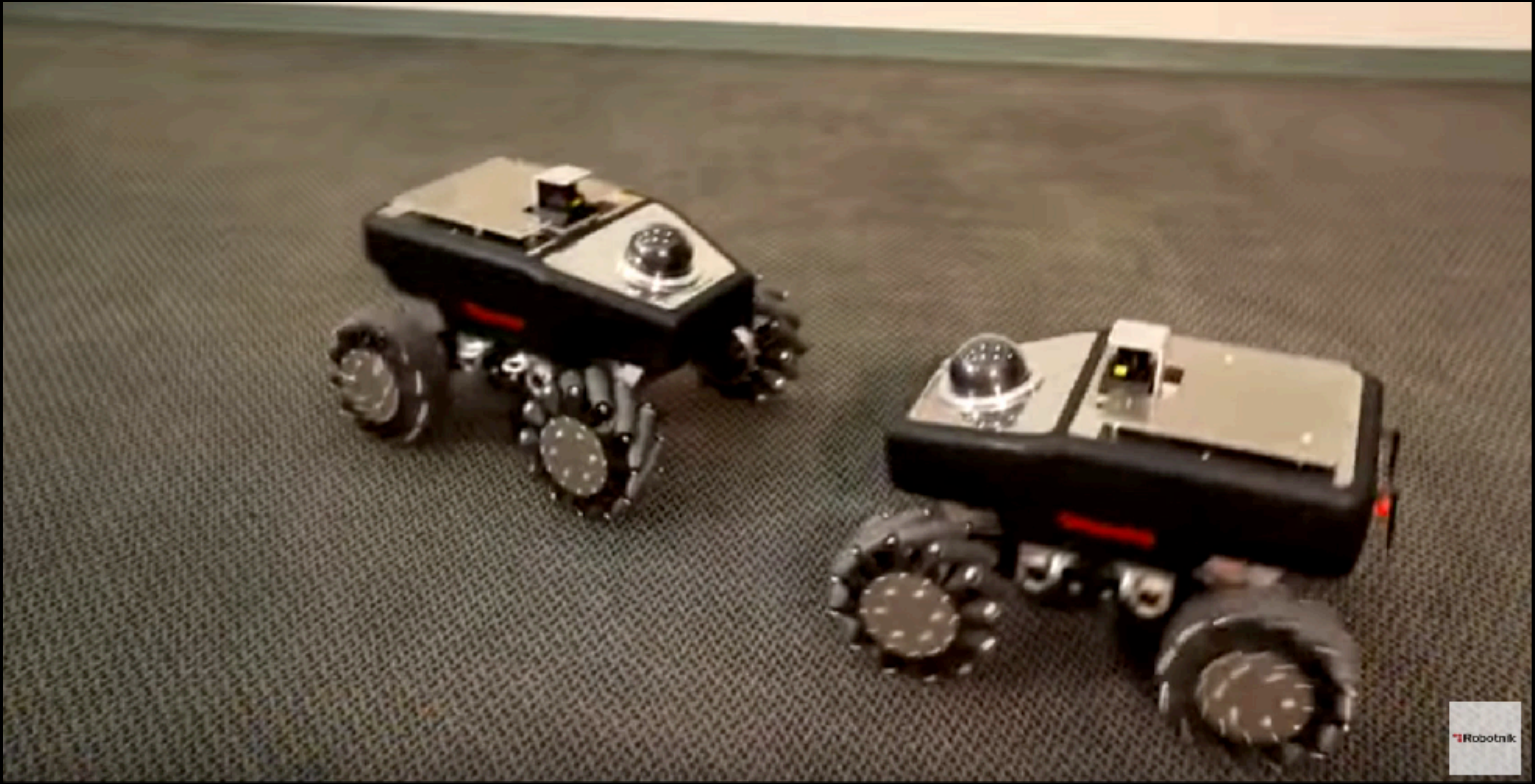
- Proprioceptive vs. exteroceptive
  - **Proprioceptive:** *“body”* sensors, e.g., motor speed, battery voltage, joint angle
  - **Exteroceptive:** *“environment”* sensors, e.g., distance measurement, light intensity
- Passive vs. active
  - **Passive:** *“measure ambient energy”*, e.g., temperature probes, cameras, microphones
  - **Active:** *“emit energy, and measure the environmental reaction”*, e.g., infrared proximity sensors, ultrasound sensors

Sensor and Actuators

- Actuators
  - For different purposes: e.g., locomotion, control of a body part, heating, sound emission.
  - Examples of electrical-to-mechanical actuators: DC motors, stepper motors, servos, loudspeakers.
- Uncertainty and disturbances
  - Causes for actuation noise: e.g., wheel slip, slack in mechanism
  - Causes for sensor noise: e.g., environmental factors, cheap circuitry

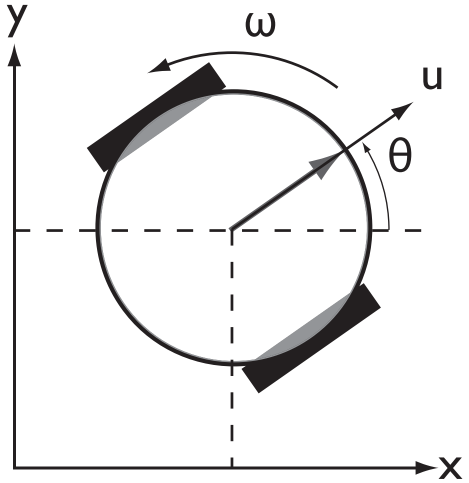
Degrees of Freedom

- Most actuators control a single degree of freedom (DOF)
  - a motor shaft controls one rotational DOF
  - a sliding part on a plotter controls one translational DOF
- Every robot has a specific number of DOF
- If there is an actuator for every DOF, then all of the DOF are controllable
- Usually not all DOF are controllable
  - **Holonomic robot:** When the number of controllable DOF is equal to robot's total DOF
  - **Non-holonomic robot:** When the number of controllable DOF is less than robot's total DOF
  - When it is larger, the robot is 'redundant'


Robotnik

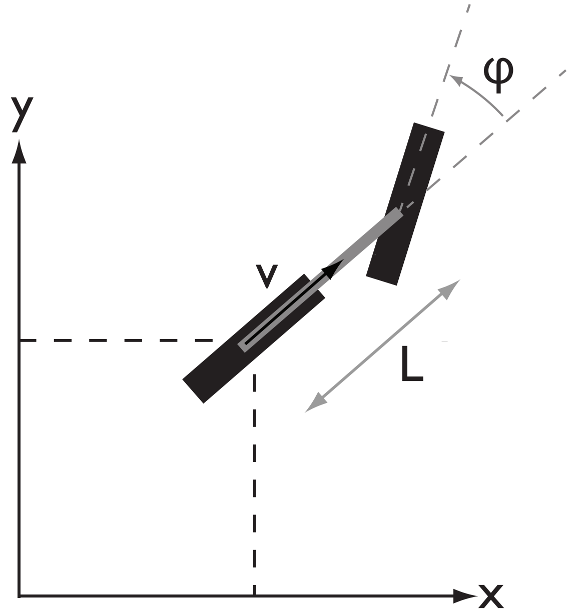
Forward Kinematics

- Differential equations describe robot motion
- How does robot state change over time as a function of control inputs?



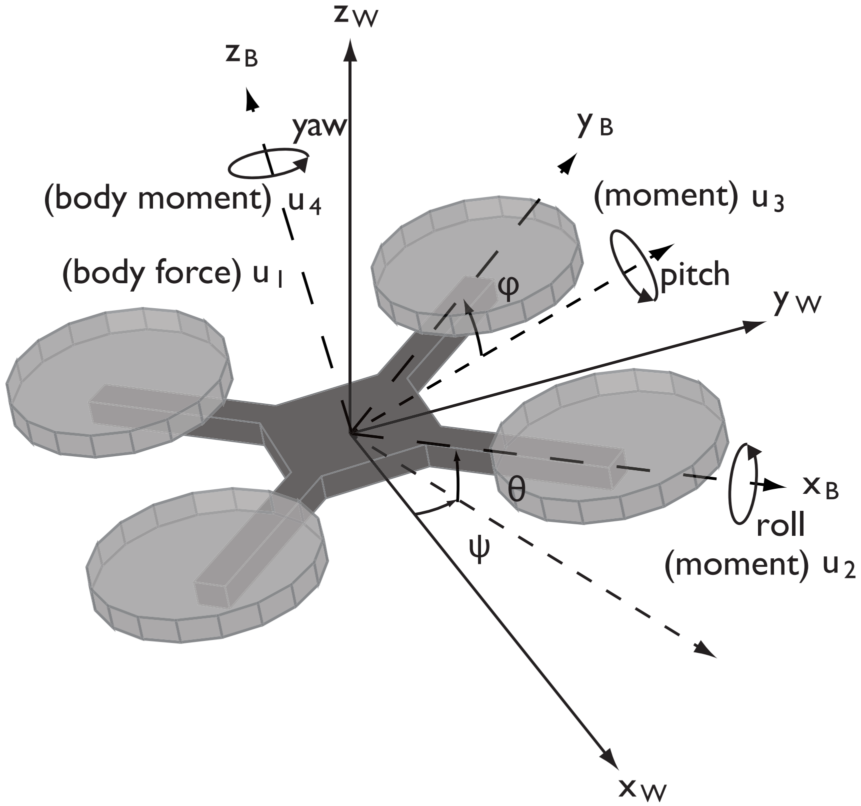
$$\begin{cases} \dot{x} = u \cdot \cos\theta \\ \dot{y} = u \cdot \sin\theta \\ \dot{\theta} = \omega \end{cases}$$

differential-drive
3 DOF (2 controllable)



$$\begin{cases} \dot{x} = v \cdot \cos\theta \\ \dot{y} = v \cdot \sin\theta \\ \dot{\theta} = v \cdot \frac{\tan\phi}{L} \end{cases}$$

Bicycle
3 DOF (2 controllable)



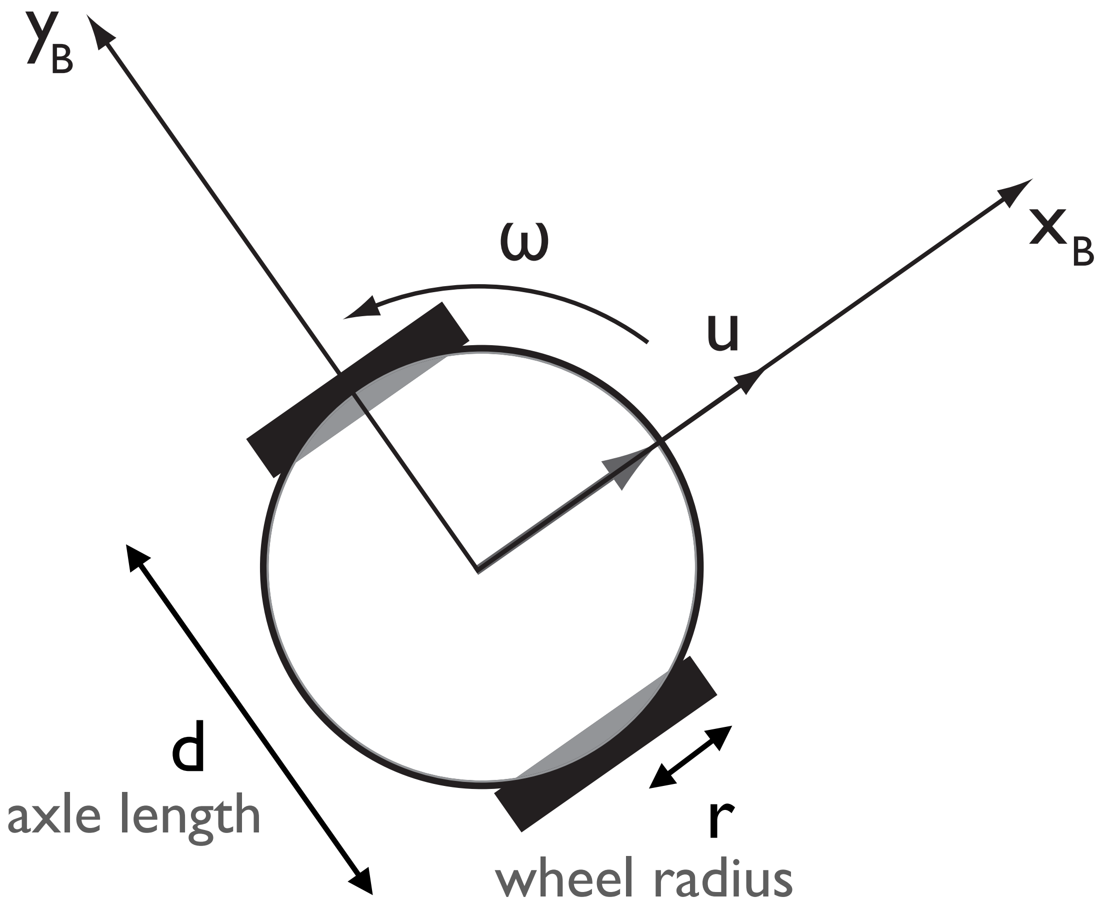
$$\begin{cases} \ddot{\mathbf{r}} = -g\mathbf{z}_W + \frac{u_1}{m}\mathbf{z}_B \\ \dot{\boldsymbol{\omega}} = I^{-1}\left(-\boldsymbol{\omega} \times \underset{\text{inertia matrix}}{I}\boldsymbol{\omega} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}\right) \end{cases}$$

Quadrotor
6 DOF (4 controllable)

Forward Kinematics (body frame)

Actuators of differential-drive:

- Left wheel speed $\dot{\phi}_l$
- Right wheel speed $\dot{\phi}_r$



Forward velocity:

$$u = \frac{r\dot{\phi}_r}{2} + \frac{r\dot{\phi}_l}{2}$$

Rotational velocity:

$$\omega = \frac{r\dot{\phi}_r}{d} - \frac{r\dot{\phi}_l}{d}$$

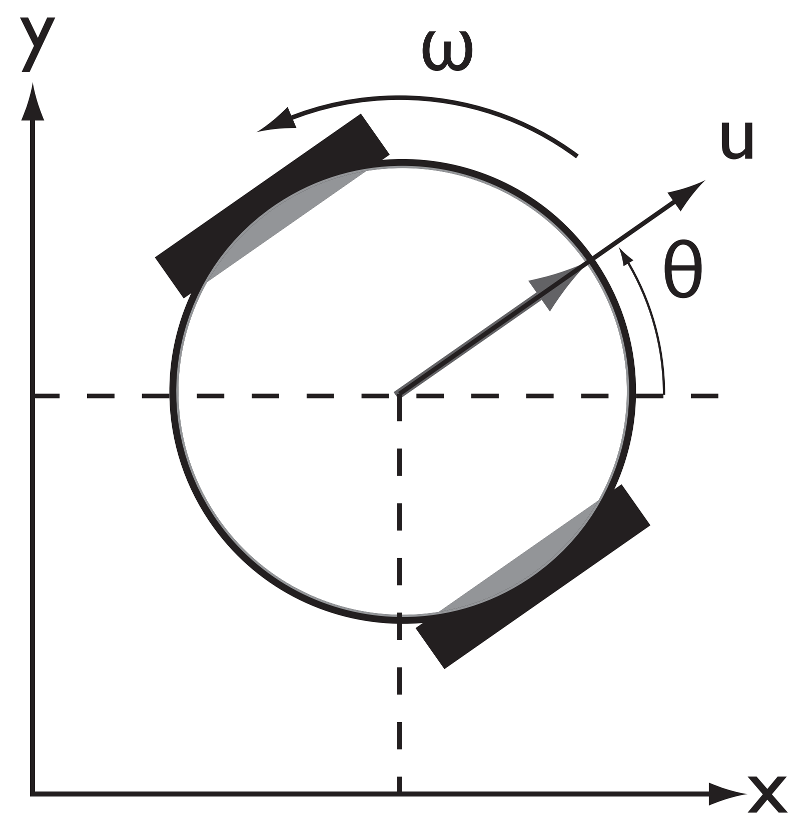
Motion:

$$\dot{x}_B = u$$
$$\dot{y}_B = 0$$
$$\dot{\theta}_B = \omega$$

UNIVERSITY OF CAMBRIDGE

Forward Kinematics (world frame)

- Rotation of coordinates
  - From body to world frames, the axes rotate by θ



$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{T(\theta)} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{\theta}_B \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 0 \\ \omega \end{bmatrix} = \begin{bmatrix} u\cos\theta \\ u\sin\theta \\ \omega \end{bmatrix}$$

Inverse Kinematics I

- We would like to control the robot velocities: $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$
- We inverse the previous equations:

$$\begin{bmatrix} u \\ 0 \\ \omega \end{bmatrix} = T^{-1}(\theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

- yielding

$$\begin{aligned} u &= \dot{x}\cos\theta + \dot{y}\sin\theta \\ \omega &= \dot{\theta} \end{aligned}$$

- under the constraint (remember than our robot is non-holonomic):

$$\dot{x}\sin\theta = \dot{y}\cos\theta$$

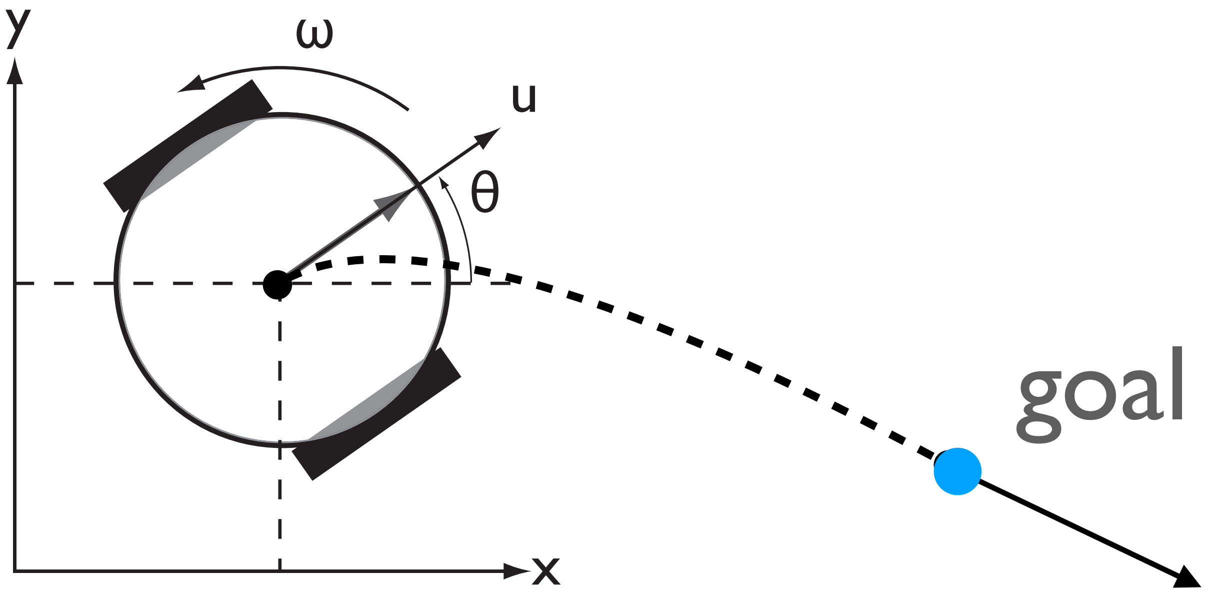
- and finally

$$\begin{aligned} \dot{\phi}_l &= u - \tfrac{\omega d}{2r} \\ \dot{\phi}_r &= u + \tfrac{\omega d}{2r} \end{aligned} \implies \begin{aligned} \dot{\phi}_l &= \dot{x}\cos\theta + \dot{y}\sin\theta - \tfrac{\dot{\theta} d}{2r} \\ \dot{\phi}_r &= \dot{x}\cos\theta + \dot{y}\sin\theta + \tfrac{\dot{\theta} d}{2r} \end{aligned}$$

Inverse Kinematics II

- We would like to control the robot to reach a goal pose: $\begin{bmatrix} x_G \\ y_G \\ \theta_G \end{bmatrix}$
- Ideally (if the robot would be holonomic), we would set

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = K \begin{bmatrix} x_G - x \\ y_G - y \\ \theta_G - \theta \end{bmatrix}$$



- To satisfy our constraint, we need to be creative. Cubic Bézier curves, for example, would satisfy our constraint if we set

$$\mathbf{p}_1 = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mathbf{p}_2 = \begin{bmatrix} x + K_1 \cos\theta \\ y + K_1 \sin\theta \end{bmatrix} \quad \mathbf{p}_3 = \begin{bmatrix} x_G + K_2 \cos\theta_G \\ y_G + K_2 \sin\theta_G \end{bmatrix} \quad \mathbf{p}_4 = \begin{bmatrix} x_G \\ y_G \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{B}(t|\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) \quad \text{with curvature:} \quad \dot{\theta} = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}$$

UNIVERSITY OF CAMBRIDGE

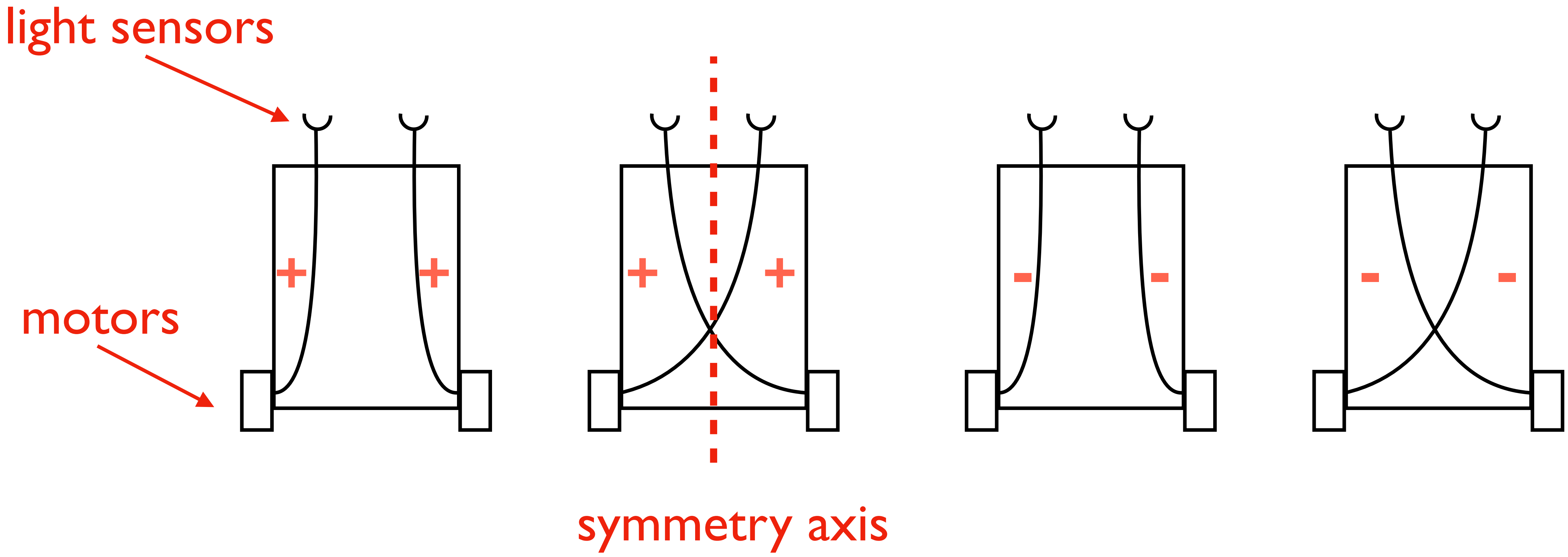
Closed-Loop vs Open-Loop

- Once we have a path that enables the robot to reach its goal, we need to follow that path:
  - **Open-loop:** Robot follows path blindly by applying the pre-computed control inputs
  - **Closed-loop:** Robot can follow path for a small duration, then observe if anything changed in the world, recompute a new adapted path (repeatedly)
- Closed-loop is much more robust to external perturbations:
  - Noisy sensors: wrong estimate of the goal position, wrong estimate of the robot position.
  - Unforeseen events, dynamic obstacles, e.g., someone walks in front of the robot.

Control Architectures

- Sensing: proximal vs. distal
    - **Proximal** architectures are close to sensor input (e.g., Braitenberg; ANN), whereas **distal** are composed of behavioral blocks (e.g., rule-based, motor-schema).
- Planning: reactive vs. deliberative
    - **Reactive:** control uses current estimate of world, time-invariant rules produce action; simple and fast to compute
    - **Deliberative:** predictions of future states are made; sequences of actions are planned that minimize some metric (e.g., collisions, energy consumption); computationally involved

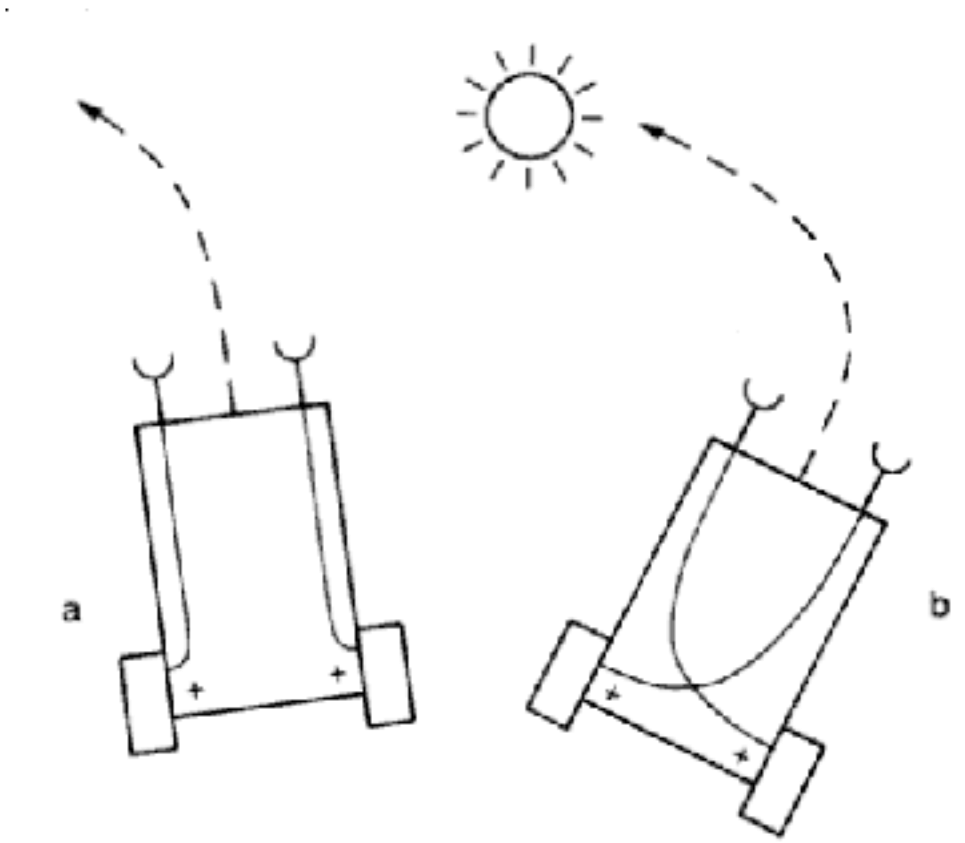
Braitenberg Vehicle I

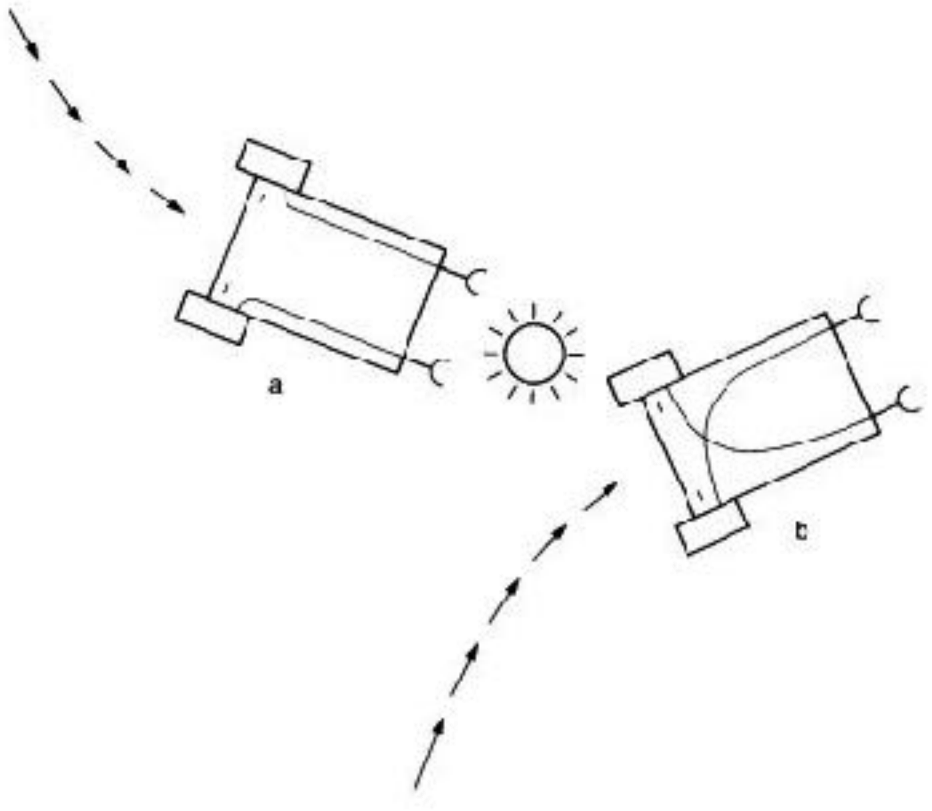


- Difference (gradient) between sensors (across symmetry axis)
- Sensors can (+) excite or (-) inhibit motors
- Original idea worked with light sensors

Valentino Braitenberg:Vehicles: Experiments in Synthetic Psychology, MIT Press,1986

Braitenberg Vehicle II



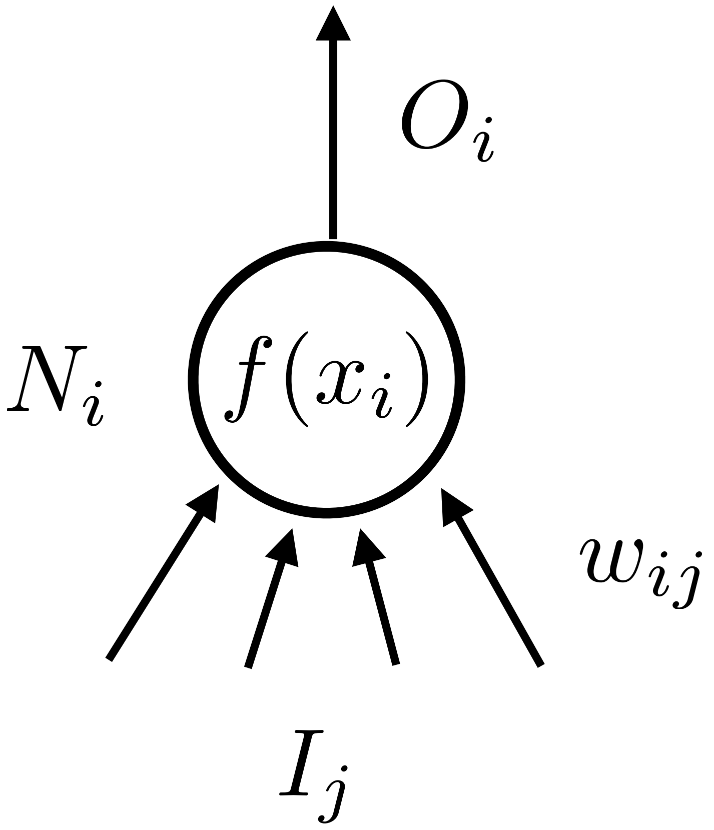


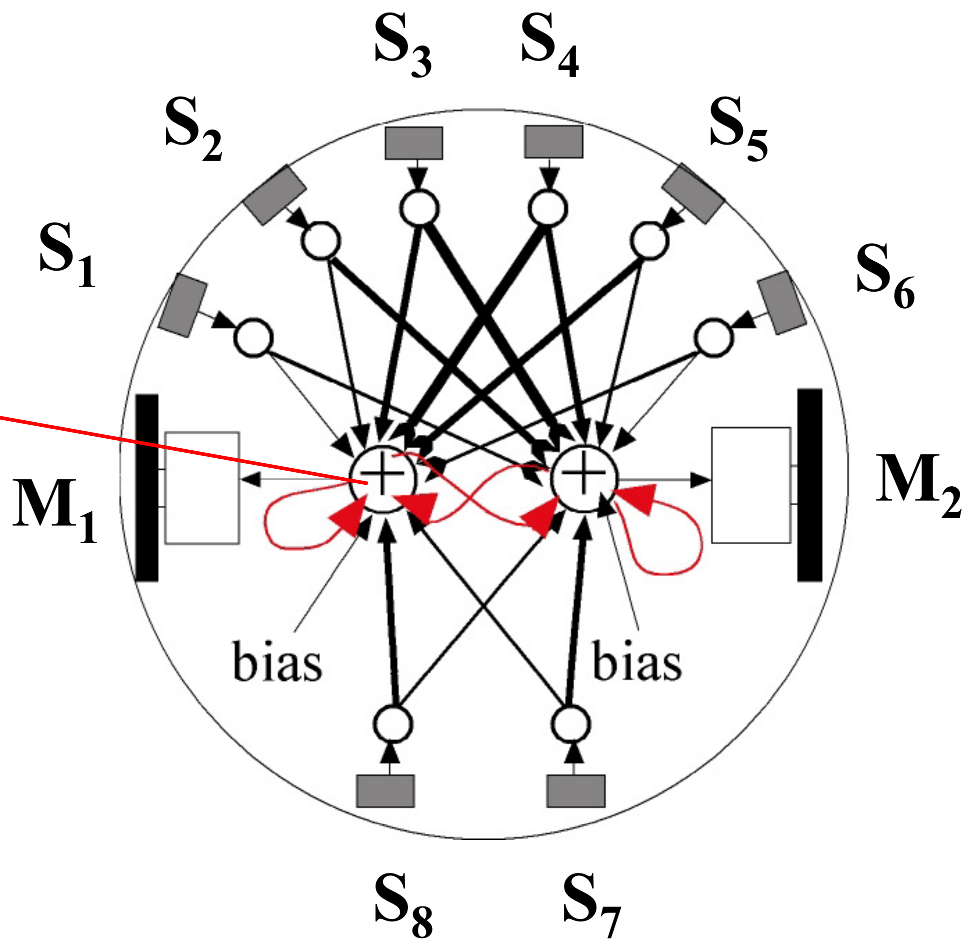
Excitatory connections

Inhibitory connections

Neural Network

$$O_i = f(x_i)$$





neuron N_i with transfer function f

$$f(x) = \tanh(x)$$

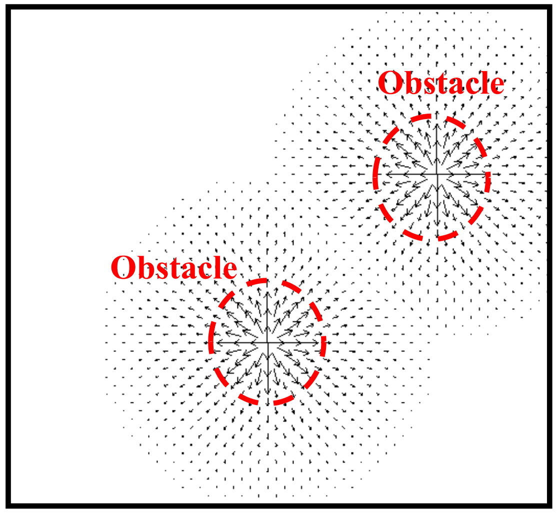
$$x_i = \sum_{j=1}^{m} w_{ij} I_j + I_0$$

Rule-Based

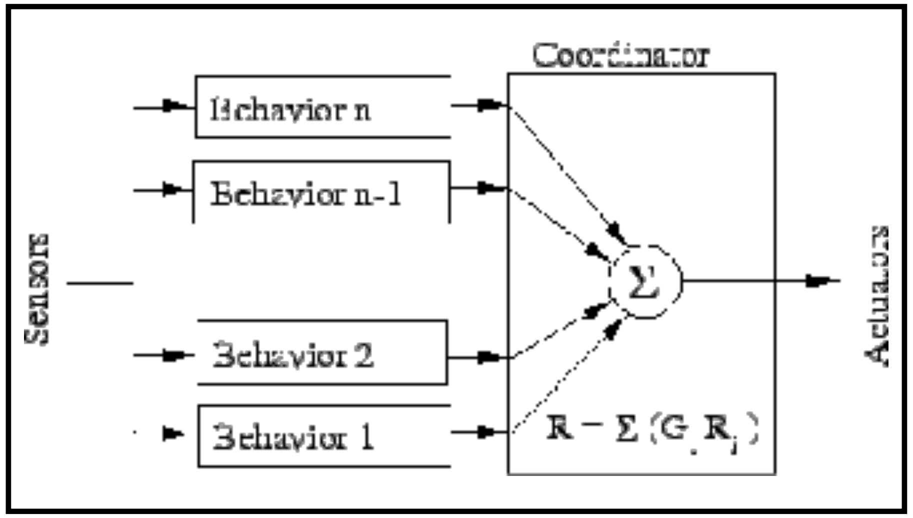
```
forever do:
  rule 1:
    if (proximity sensors on left active) then:
      turn right
  rule 2:
    if (proximity sensors on right active) then:
      turn left
  rule 3:
    if (no proximity sensors active) then:
       move forwards
```

Other Classical Paradigms

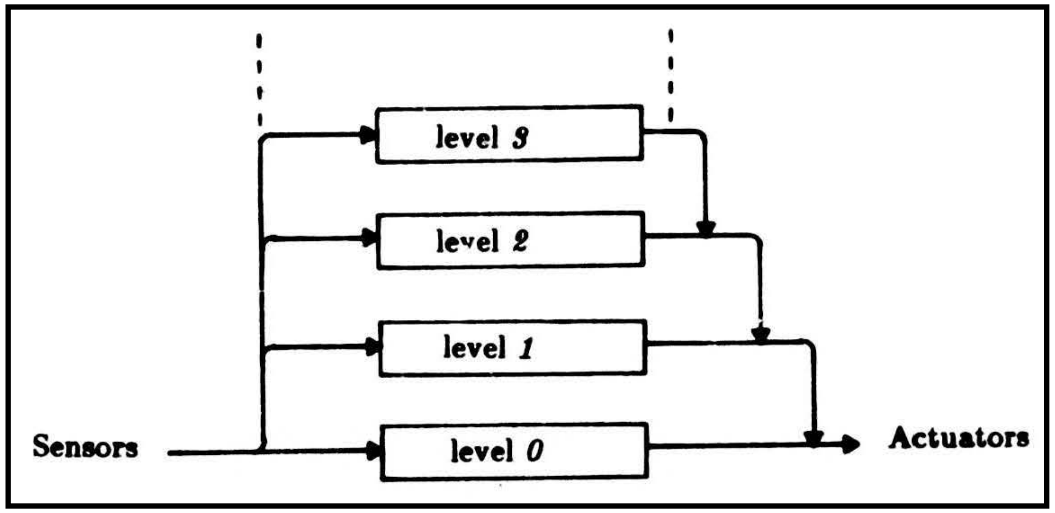
- Potential Field (Khatib, 1986)
- Motor Schema (Arkin, 1989)
- Subsumption Architecture (Brooks, 1986)



potential field



motor schema

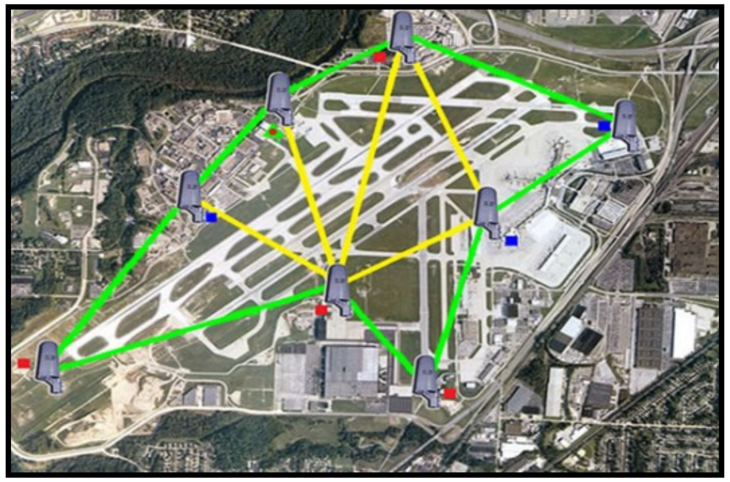


subsumption architecture

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
- Numerous commercial, civil, military applications

search & rescue



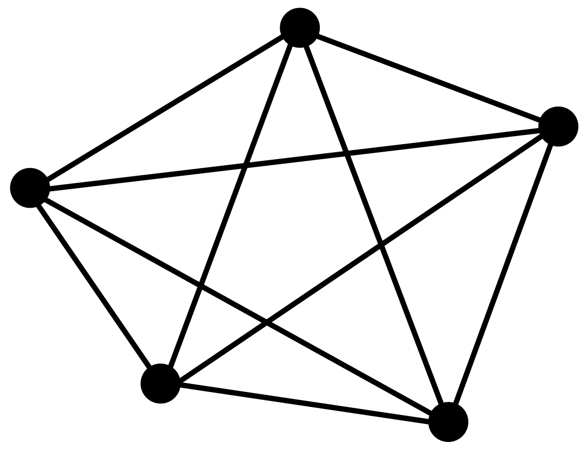
surveillance / monitoring

product pickup / delivery

Taxonomy

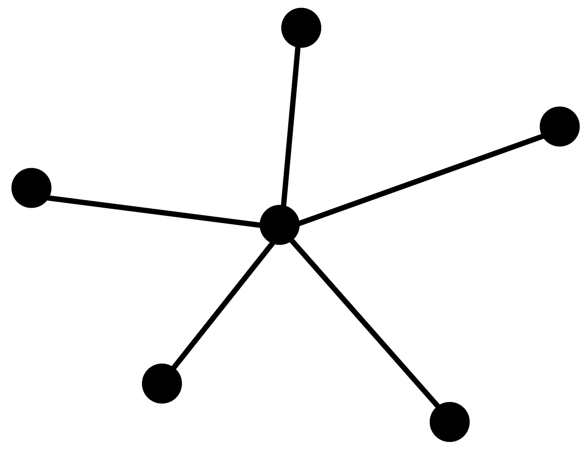
- Architecture: centralized vs. decentralized
  - **Centralized:** one control/estimation unit communicates with all robots to issue commands; requires synchronized, reliable communication channels; single-point failures
  - **Decentralized:** scalable, robust to failure; often asynchronous; sub-optimal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

Communication Topologies



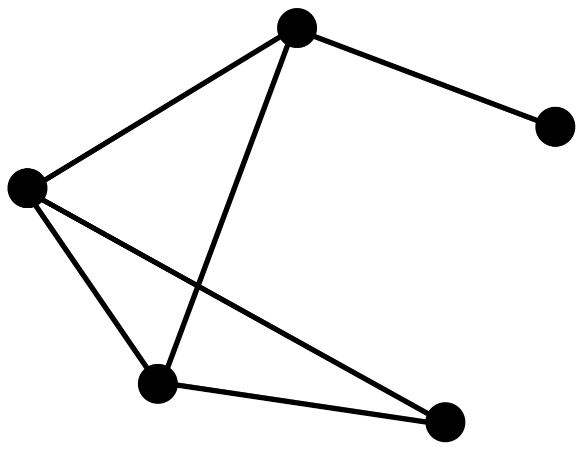
fully connected

centralized / decentralized
coordination



star topology

centralized / decentralized
coordination



random mesh

decentralized
coordination

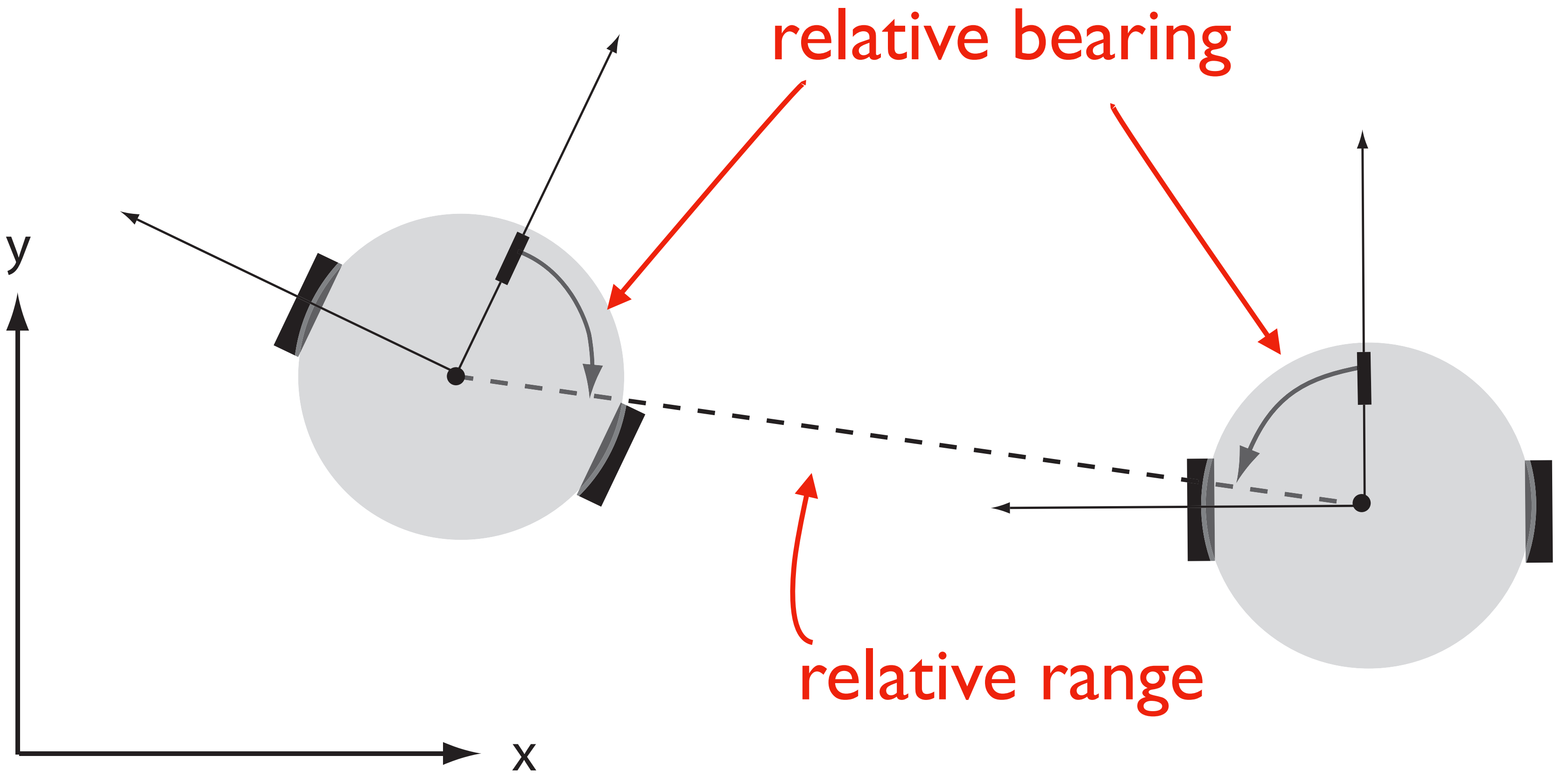
Decentralization

- 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 (with graceful degradation)

Distributed Estimation

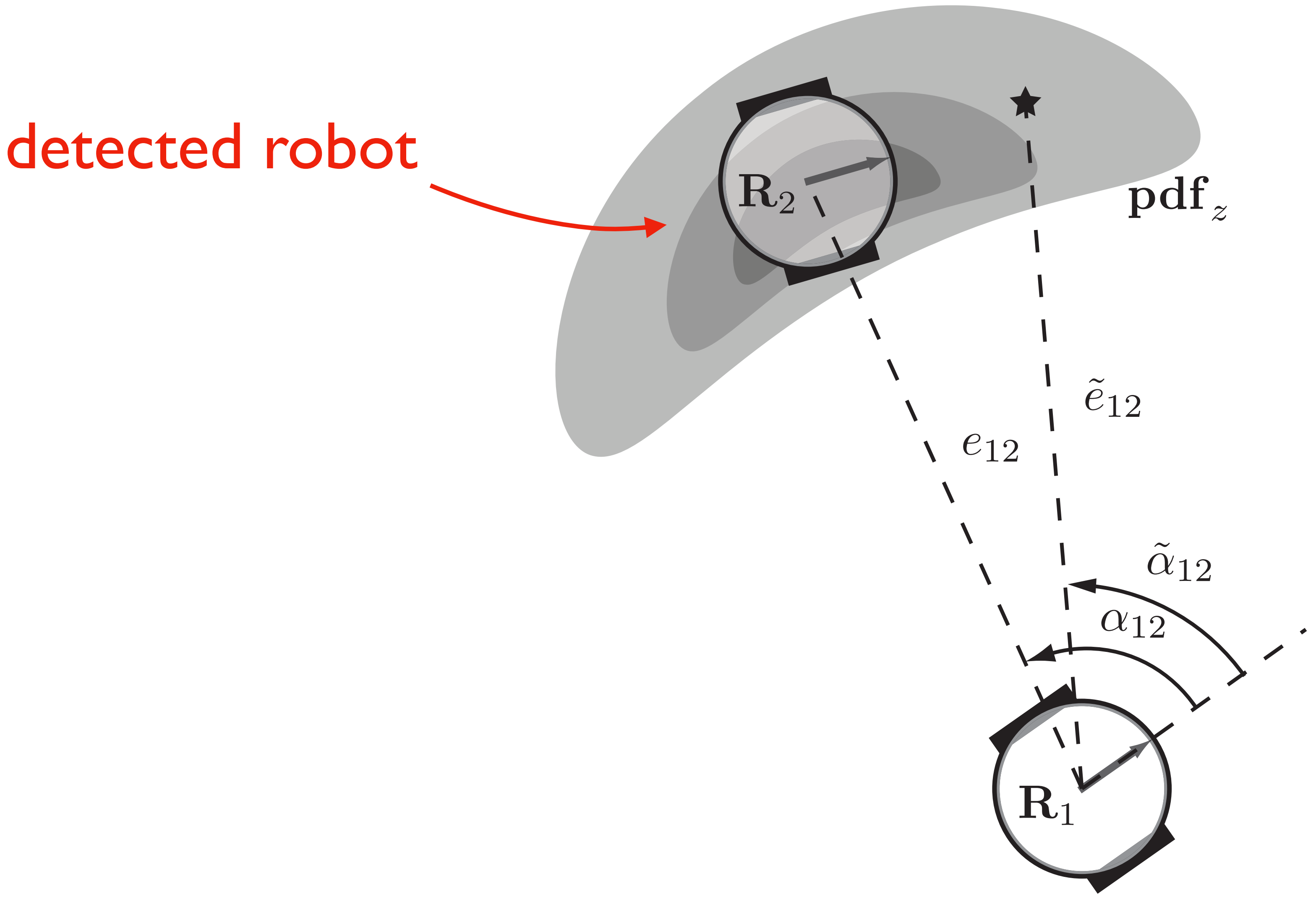
- Goal: Estimate a local or global variable in distributed manner
- Filters can be distributed
  - Examples: Kalman filter, particle filter
  - Method: fuse relative observations of other robots
  - Correct implementation considers relative observations as dependent measurements; the whole history of measurements needs to be tracked (to avoid rumor propagation)!
- Other mechanisms:
  - Opportunistic mechanisms
  - Consensus (agreement mechanism)

Collaborative Localization I



- Collaborative localization uses relative inter-robot observations
- Robots communicate their position estimate
- Fuse relative observation by transforming position into local frame

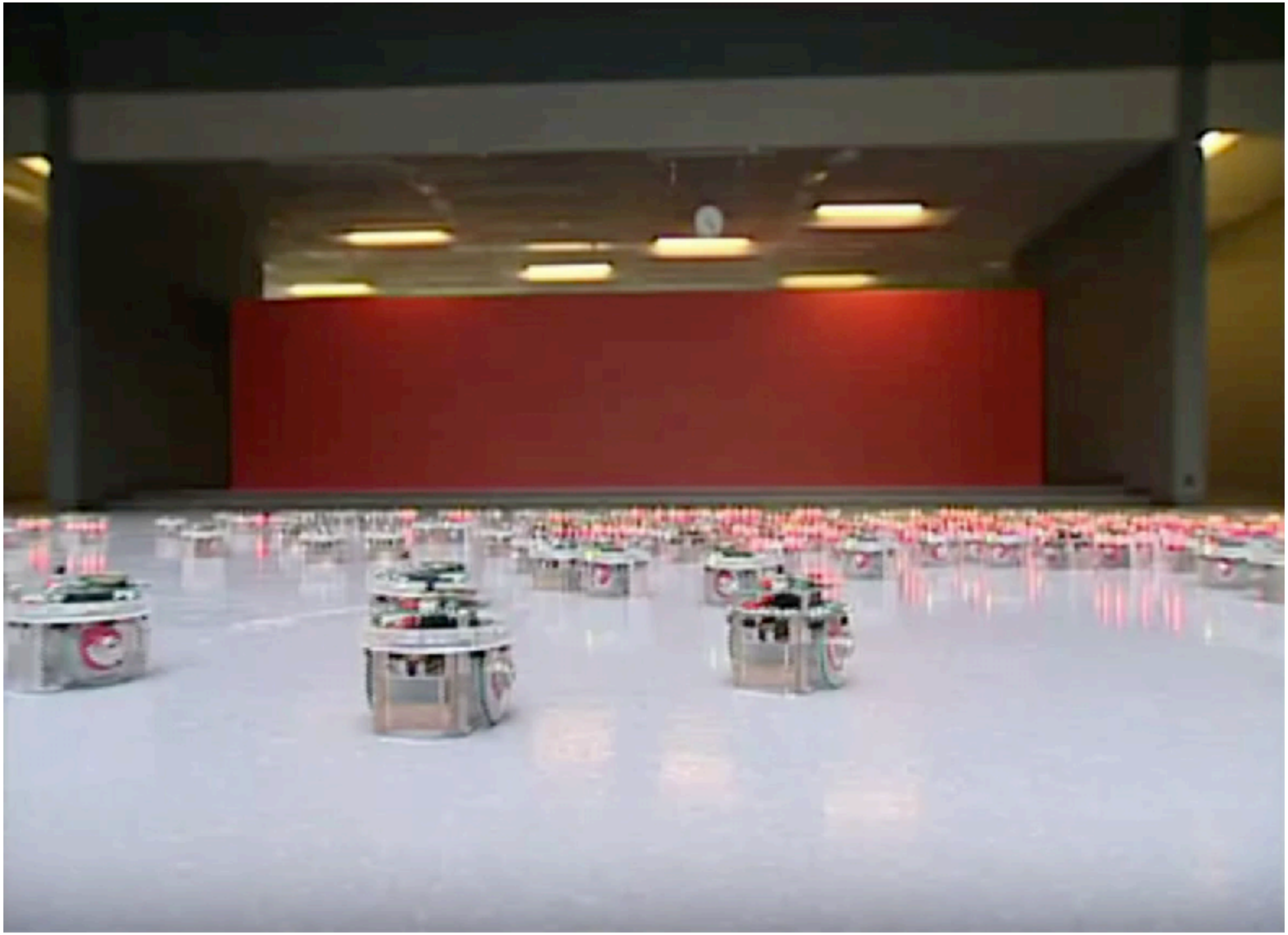
Collaborative Localization II



- This example considers a particle filter (Kalman filter also possible)
- Detected robot weights its particles using belief of detecting robot
- Particles re-sampled according to new weights (standard filter)

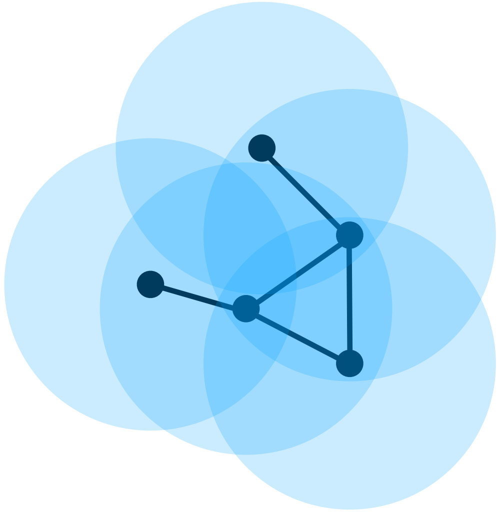
Collaborative Localization III

Coordination



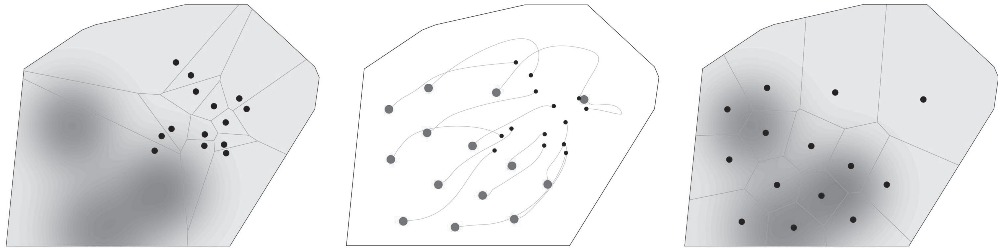
Distributed Coordination

- Coordinated motion: *formations, flocking*
  - Potential field (sum of local forces)
  - Network control: Use graph as an abstraction of communication network; use proximity graphs
  - Leader-follower formations



disc-graph

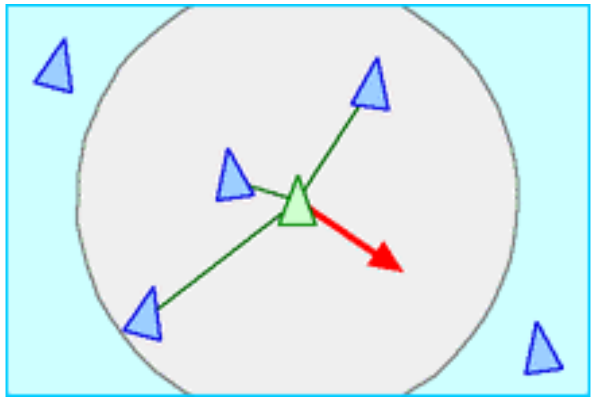
- Allocation problems: *role / resource distribution*
  - Market-based algorithms
  - Threshold-based algorithms
- Coverage: *coverage of spatial areas*
  - Lloyds algorithm



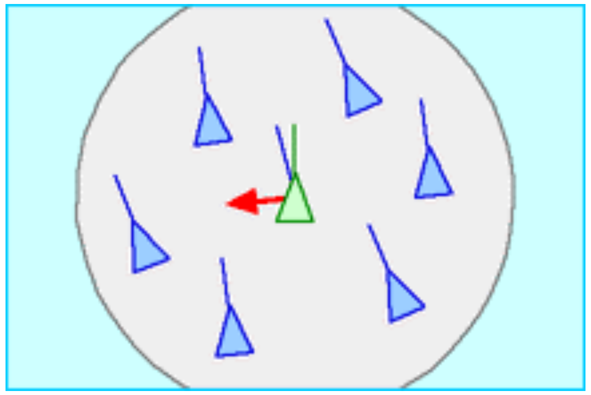
gradient-based coverage control

Formation Control / Flocking

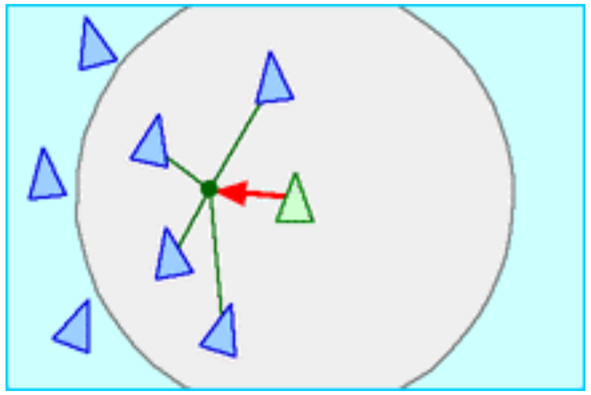
Reynolds' boids (1987)



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

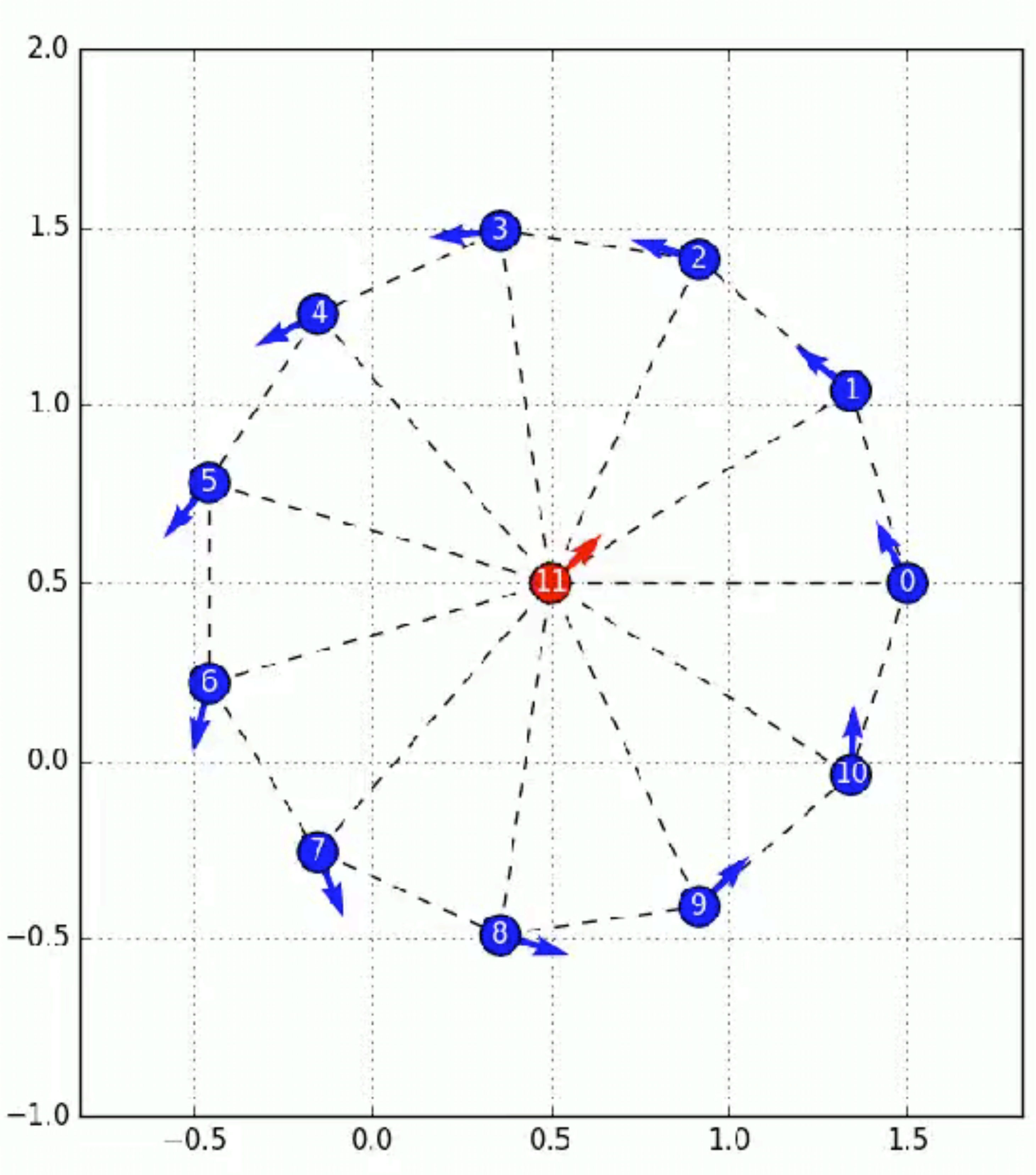
Reynolds, C.W. ; Flocks, Herds, and Schools: A Distributed Behavioral Model, SIGGRAPH 1987

The Consensus Algorithm

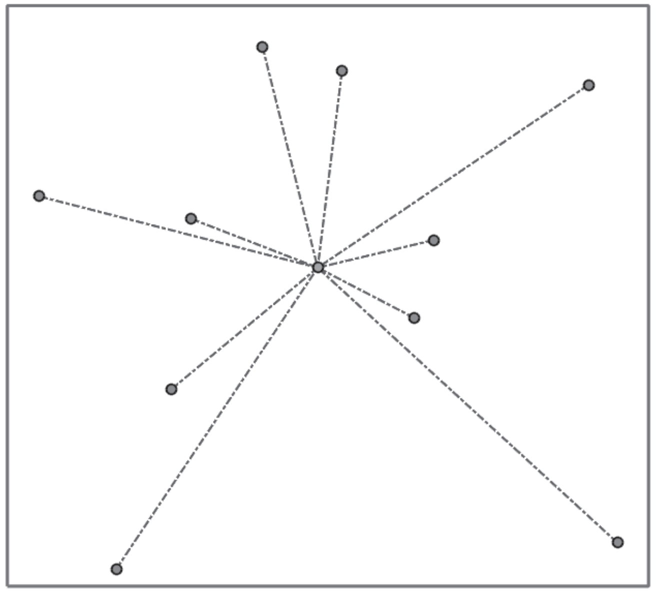
- Aim of consensus:
  - Reach decentralized agreement
  - Purely based on local interactions
- Consensus applications
  - Motion coordination
  - Cooperative estimation
  - Synchronization
- Consensus update: $x_i[t+1] = \underset{\substack{\uparrow \\ \text{averaging function}}}{f}(x_i[t], \{x_j[t] | j \in \underset{\substack{\uparrow \\ \text{all neighbor values}}}{\mathcal{N}_i}\})$
- Consensus outcome:
  - All robots converge to same value (at exponential rate)

Consensus and cooperation in networked multi-agent systems; Olfati-Saber, Fax, Murray; 2007

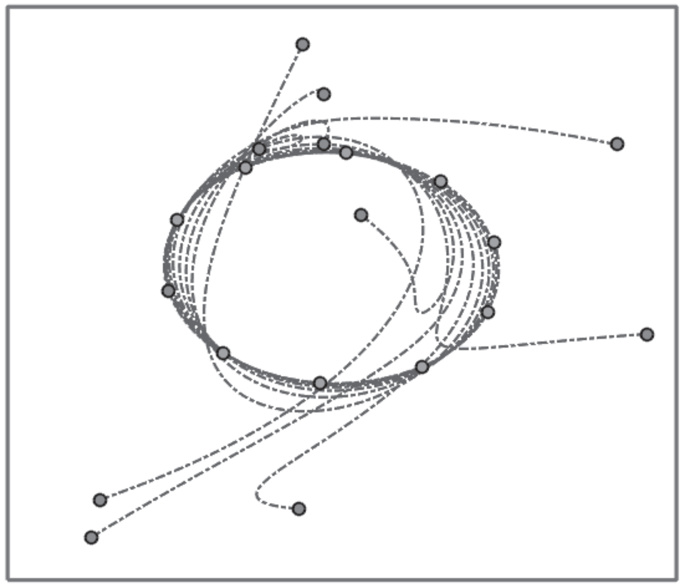
Consensus for Flocking



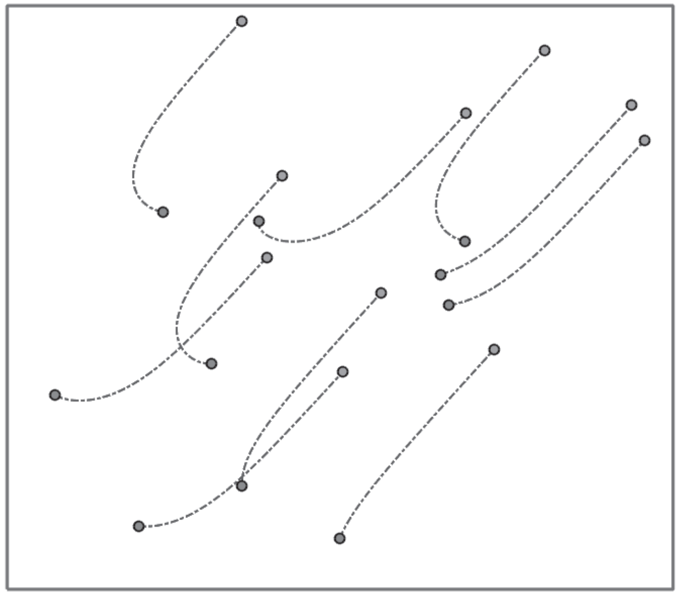
Applications of Consensus



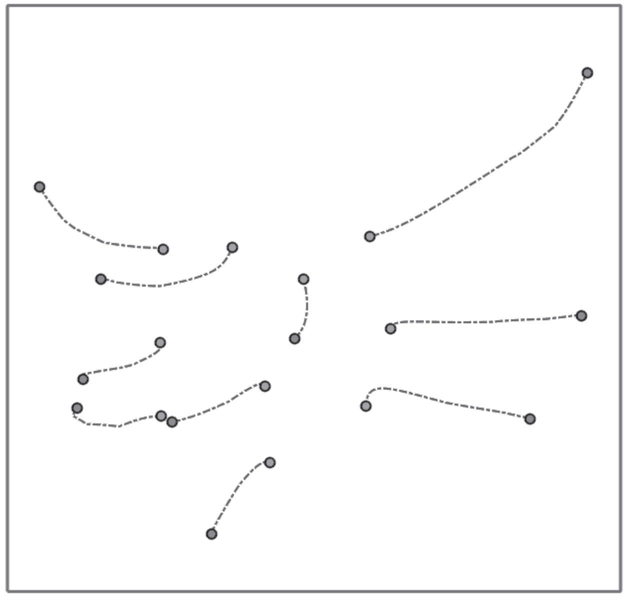
rendezvous



cyclic pursuit



flocking



formation

Cortes and Egerstedt: Coordinated Control of Multi-Robot Systems: A Survey, 2017

Summary

- Mobile robot control
  - Kinematic principles and control architectures
- Multi-robot systems: estimation and coordination
  - Collaborative localization as an example
  - Flocking / formation control as an example
- What we did not talk about (there is much more!):
  - Noise and uncertainty
  - Planning algorithms
  - Learning algorithms (AI)

References

Fundamental concepts:

- Elements of Robotics, F Mondada et al., 2018
- Autonomous Mobile Robots, R Siegwart et al., 2004

State of the art:

- The grand challenges of Science Robotics, *Science*, Yang et al. 2018

Further reading:

- Probabilistic Robotics, S Thrun et al, 2005
- Springer Handbook of Robotics, B Siciliano et al., 2008
- Graph Theoretic Methods in Multi-agent Networks, Egerstedt et al., 2010

Seminal papers:

- Motor Schema-Based Mobile Robot Navigation, RC Arkin, 1989
- A Robust Layered Control System for a Mobile Robot, RA Brooks, 1985
- Real-time obstacle avoidance for manipulators and mobile robots, O Khatib, 1986

Internships:

If you are interested - contact me!

asp45@

Course:

Mobile Robot Systems

Lent 2018-19, as part of the new Paper 10

The course will be open to both Part II and Part III students (the max. student number will be capped)