Multirobot Systems

Lecture
Persistent coverage

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

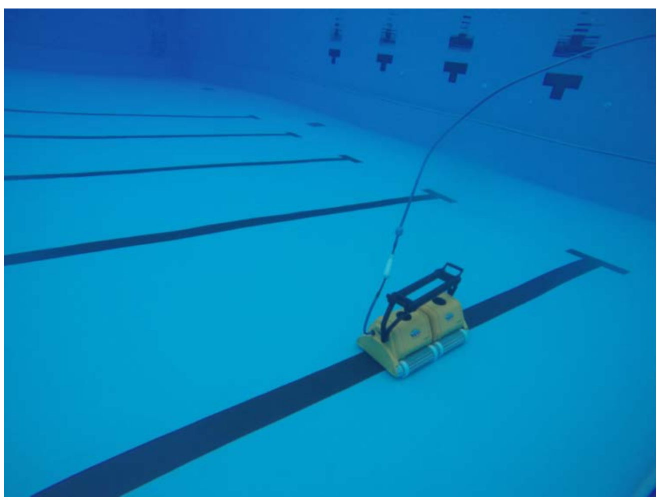
Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Index

Introduction

- What is coverage?
  - Coverage of an area refers to the goal of collecting data or performing some action at each point in a domain of interest
  - Some applications:
    - Cleaning
    - Pool cleaner
    - Lawn mower
    - Snow-blower

[Roomba robot]



[http://roboticpoolcleanersingapore.com]

[Husqvarna Automower]

雪の取り込み風景（除雪ロボット前部）

Introduction

- One agent vs multiple agents
  - One combine harvester is fine

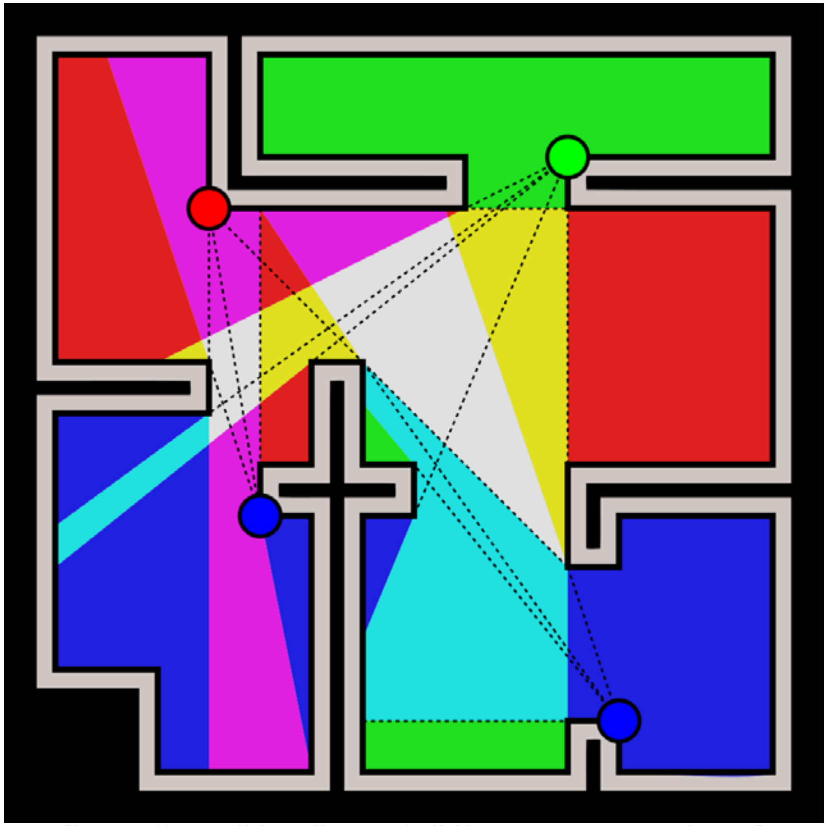
  - Two are faster

  - But many are even better

[www.deere.com]

- Static coverage
  - A set of agents is deployed to cover an area
  - Static problem, also known as deployment
  - Examples:
    - Surveillance / Monitoring:
      - Art gallery problem

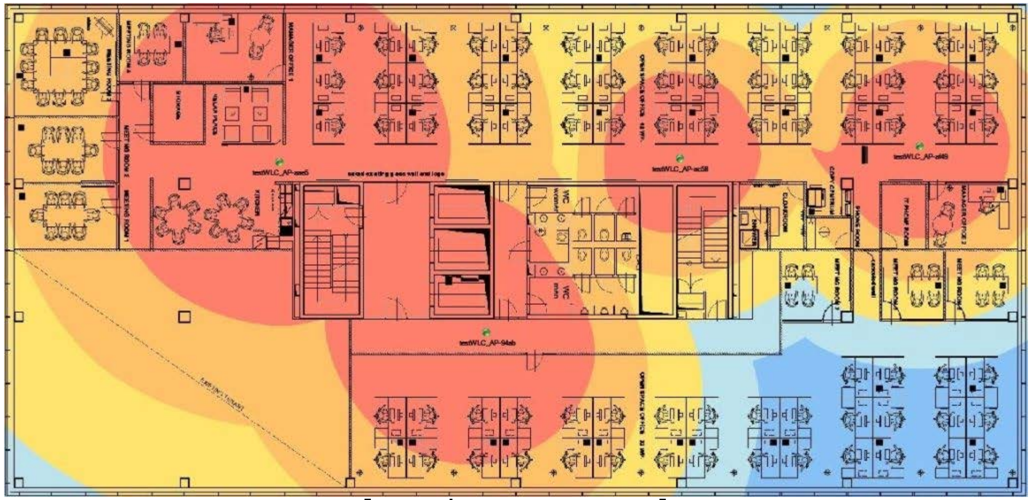
Placing four guards at the given points will guard the entire museum.



[https://en.wikipedia.org/wiki/Art_gallery_problem]

Introduction

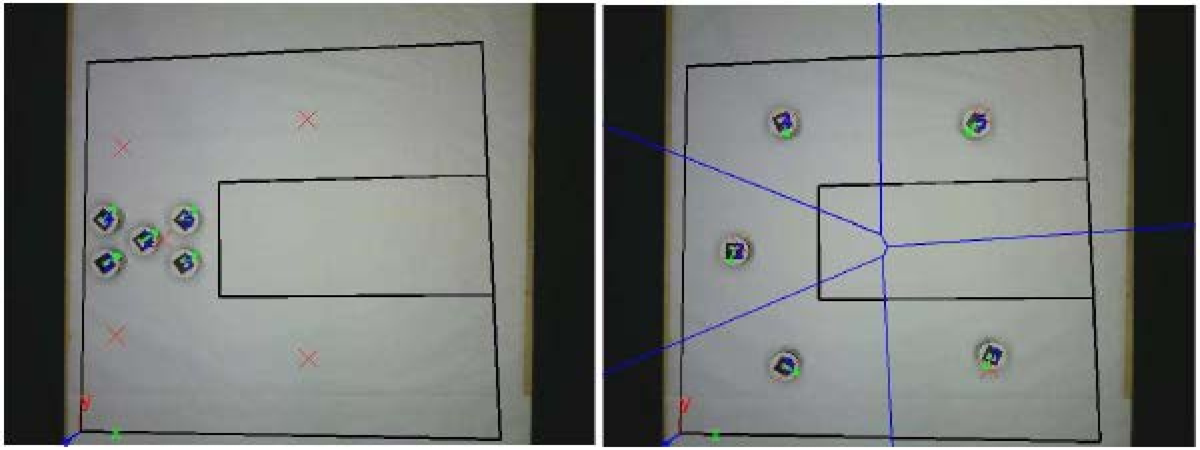
- Static coverage
  - A set of agents is deployed to cover an area
  - Static problem, also known as deployment
  - Examples:
    - WiFi coverage



[www.ittsystems.com]

Introduction

- Static coverage (Bibliography)
  - Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., Rus, D., Voronoi coverage of non-convex environments with a group of networked robots. ICRA, 2010
  - Gusrialdi, A., Hirche, S., Hatanaka, T., Fujita, M., Voronoi based coverage control with anisotropic sensors. ACC 2008
  - Popa, D. O., Helm, C., Stephanou, H. E., Sanderson, A. C., Robotic deployment of sensor networks using potential fields. ICRA, 2004
  - Cassandras, C. G., Li, W., Sensor networks and cooperative control. European Journal of Control 11 (4-5), 436-463, 2005



Introduction

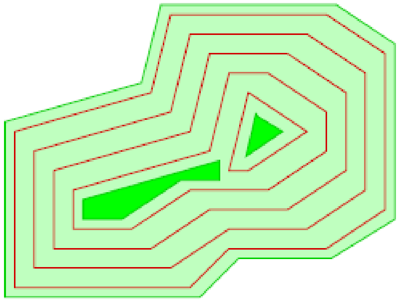
- Dynamic coverage
  - If the number of agents is not enough to cover the area in one go, a dynamic approach is required
  - Path planning
  - Multiple agents coordination

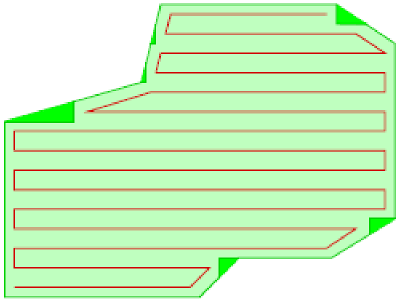
[Roomba robot]

[https://dronelife.com]

Introduction

- Dynamic coverage (Bibliography)
  - Arkin, E. M., Fekete, S. P., Mitchell, J. S. B., Approximation algorithms for lawn mowing and milling. Computational Geometry: Theory and Applications 17 (1-2), 2000, 25-50.
  - Choset, H., 2001. Coverage for robotics - a survey of recent results. Annals of Mathematic and Artificial Intelligence 31 (1-4), 113-126.



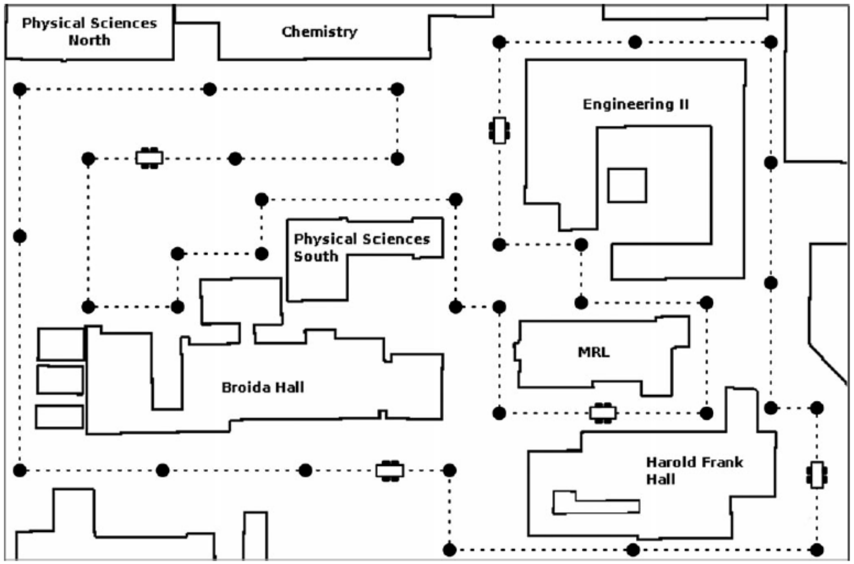


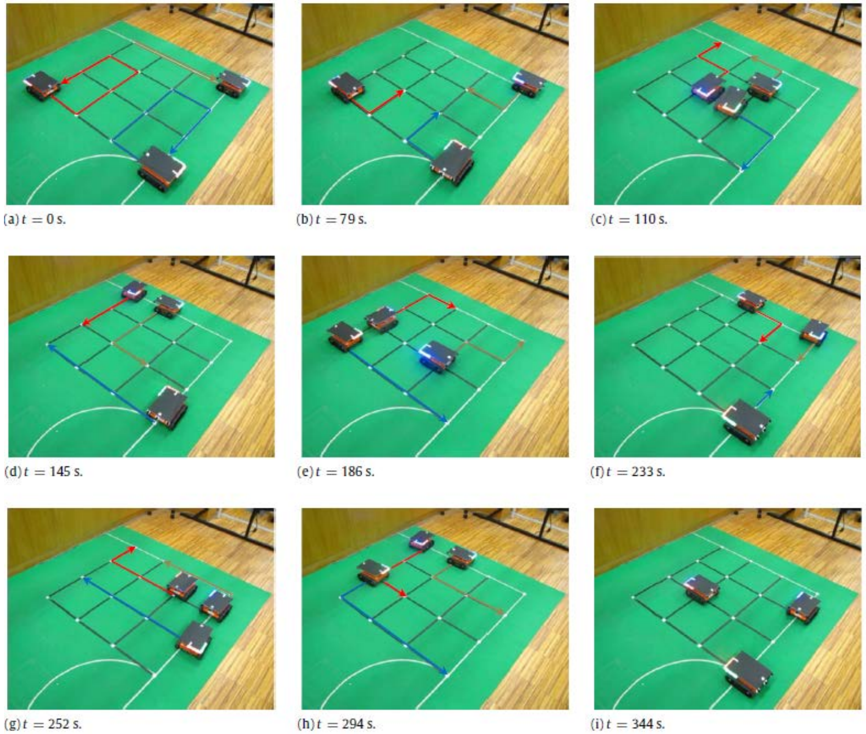
Introduction

- Persistent coverage
  - The achieved coverage degrades (or decays) with time
    - Lawn mowing: The lawn grows
    - Snow removal: The snow continues falling
    - Surveillance: Thieves can appear at any time
    - Cleaning: The room gets dirty because there are people around
  - To keep some level of coverage the agents’ actions have to be maintained in time

Introduction

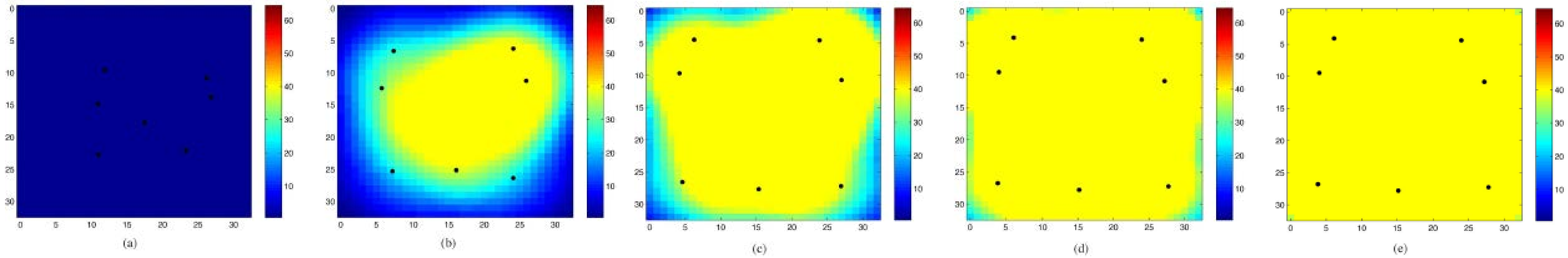
- Distributed persistent coverage (Bibliography)
  - F. Pasqualetti, J. W. Durham, and F. Bullo. Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. IEEE Trans. on Robotics, 28(5):1181–1188, 2012
  - D. Portugal and R. P. Rocha. Distributed multi-robot patrol: A scalable and fault-tolerant framework. Robotics and Autonomous Systems, 61(12):1572–1587, 2013





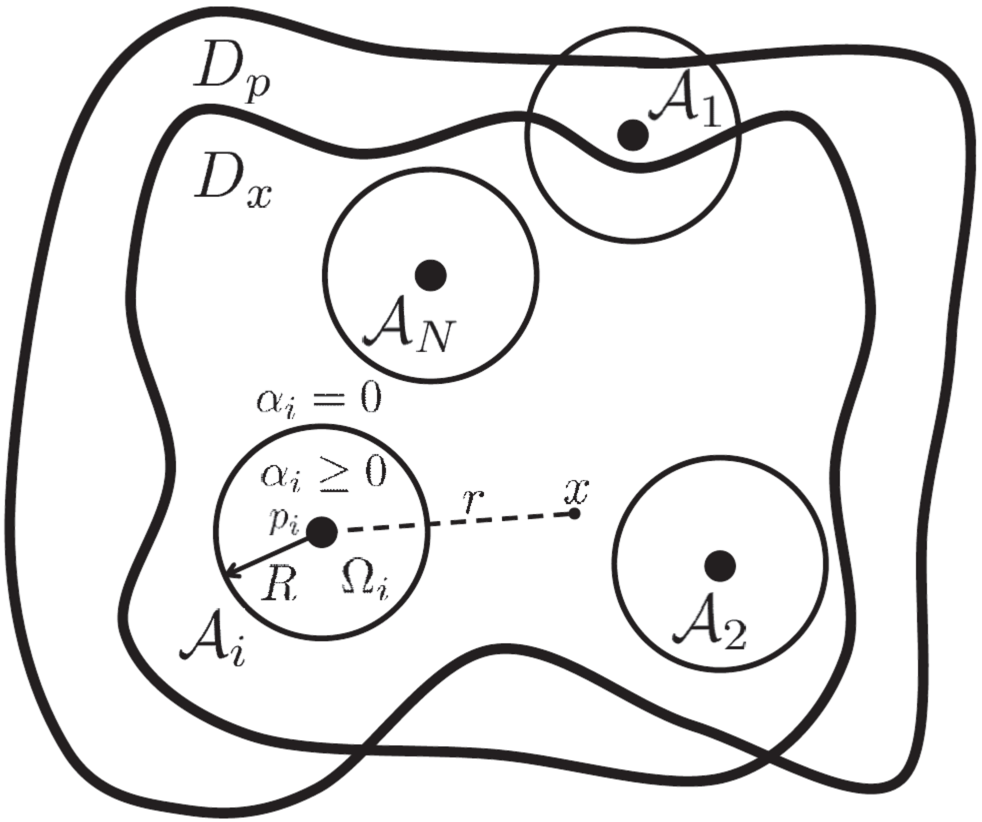
Introduction

- Distributed persistent coverage (Bibliography)
  - Hokayem, P., Stipanovic, D., Spong, M., On persistent coverage control. IEEE Conference on Decision and Control, 2007
  - I. I. Hussein and D. M. Stipanovic, “Effective coverage control for mobile sensor networks with guaranteed collision avoidance,” IEEE Trans. on Control Systems Technology, vol. 15, no. 4, pp. 642-657, 2007

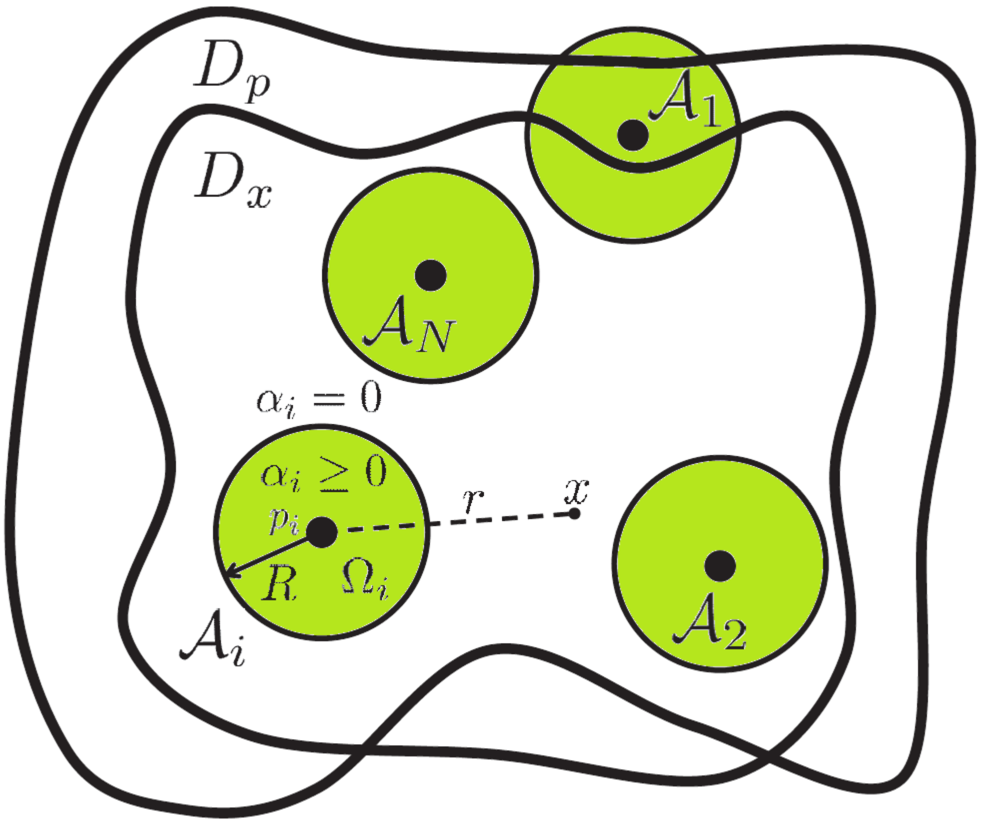


Problem formulation

- Problem formulation of the persistent coverage task

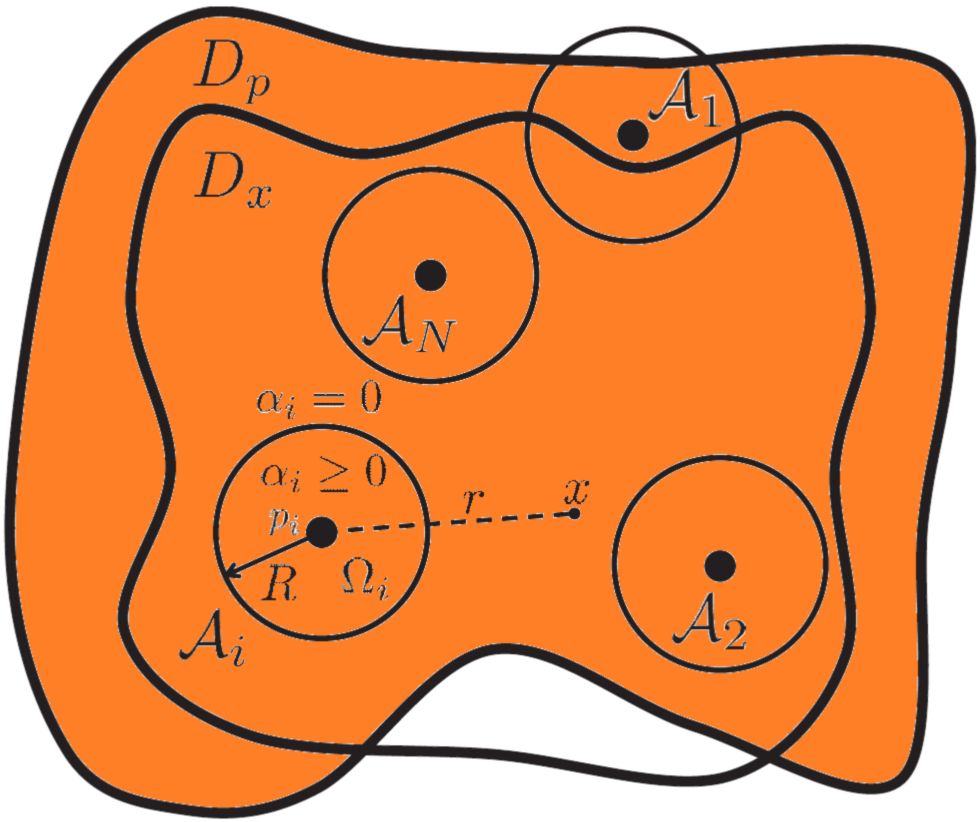


- The agents:
  - Let us consider a team of N agents $A=\{A_1, \ldots, A_N\}$
  - The coverage action of each agent is performed in domain Ω_i



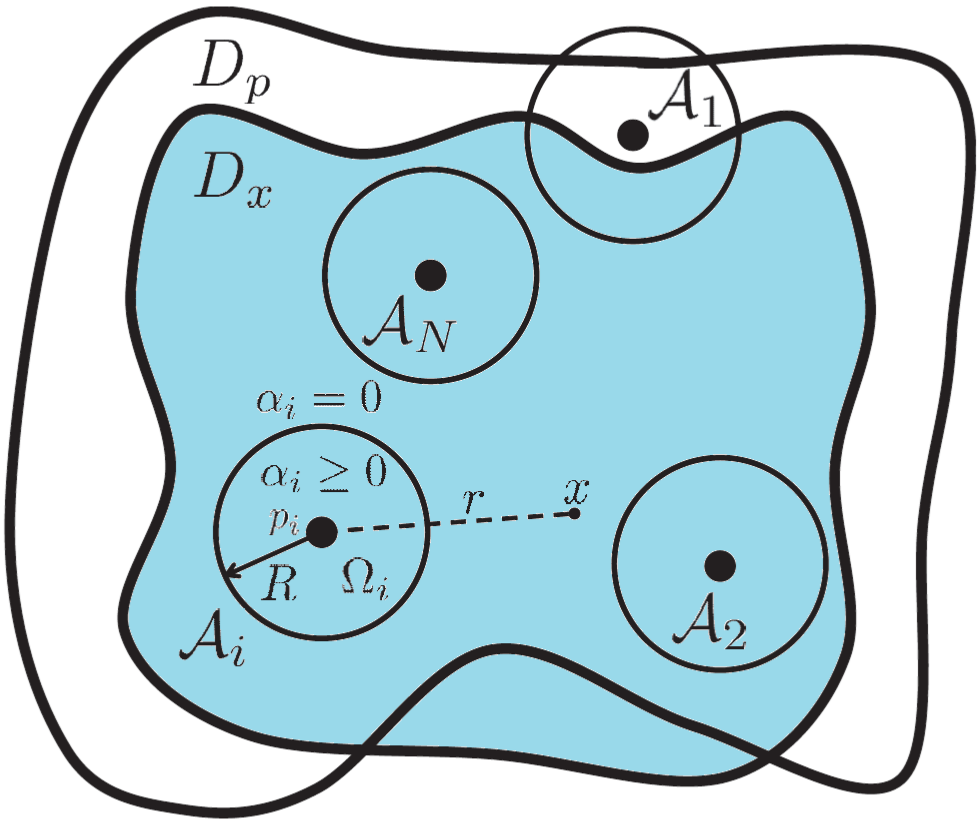
Problem formulation

- The agents:
  - The agents move in a domain D_p
  - Position of the agent i: $p_i(t) = [\, p_{ix} , p_{iy} \,]^T$
  - Mobile agents are holonomic: $\dot{p}_i = u_i$ with u_i the input motion



Problem formulation

- The environment:
  - The goal is to reach a desired coverage level for all points in domain D_x



Problem formulation

- The environment:
  - Evolution of the coverage level

$$\frac{\partial \Lambda}{\partial t} = A \cdot \Lambda + B \cdot \alpha$$

$\Lambda(x,t) \in \mathbb{R}^+$: Coverage Level
$\alpha(x,t) \in \mathbb{R}^+$: Coverage action
$A \in \mathbb{R}$: state gain $\quad A < 0$
$B \in \mathbb{R}$: input gain $\quad B > 0$

  - Error coverage (Domain D_x)

$$e_{D_x} = \int_{D_x} \Phi \cdot (\Lambda^* - \Lambda)^2 dx$$

$D_x \subset \mathbb{R}^2$: Coverage Domain
$\Lambda^*(x) \in \mathbb{R}^+$: Coverage Objective
$\Phi(x) \in (0,1]$: Coverage Priority

  - Coverage priority $\Phi \in (0, 1]$ is the priority to cover each point
    - Example: watering the garden

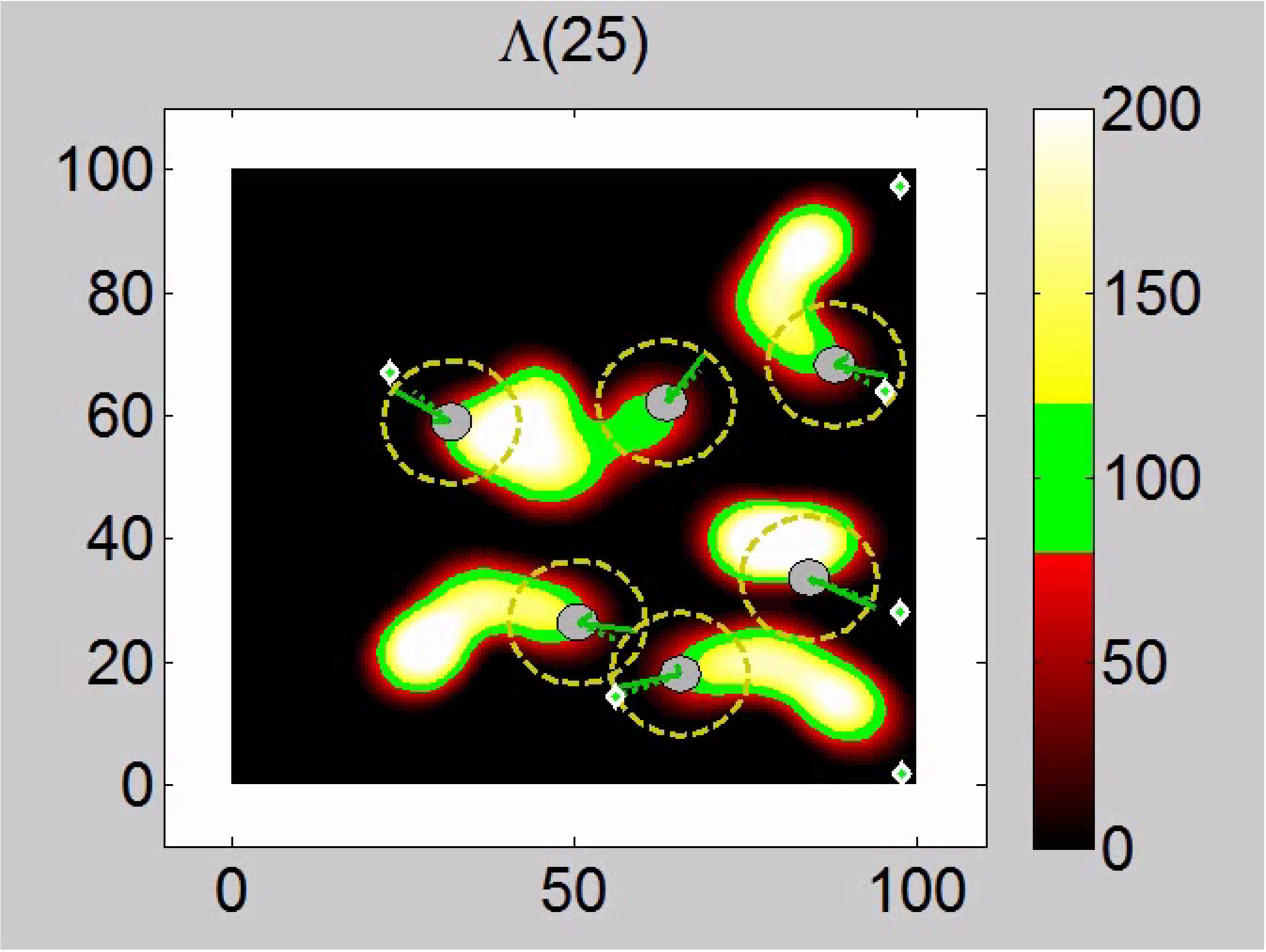
Problem formulation

- The goal:
  - The aim of our problem is to minimize the coverage error of the domain by reaching the desired coverage level $\Lambda^*(x)$ all over the domain
    - No more, no less:
      - There are applications that require a particular coverage level, and higher coverage leads to a waste of energy, as for example cleaning, or to bad results as painting.

[Y. Tang, W. Chen 2015]

Problem formulation

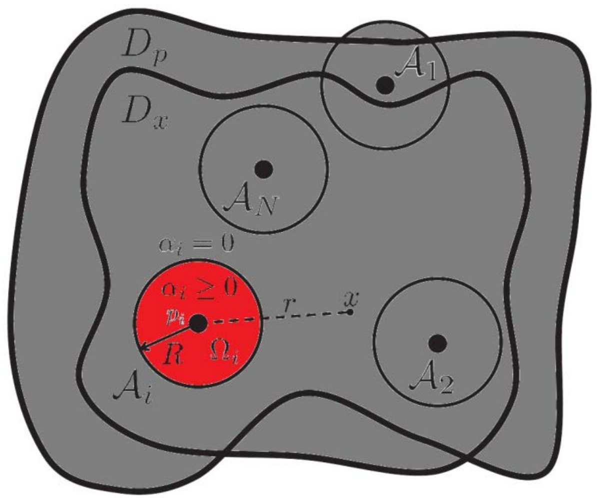
- The goal:

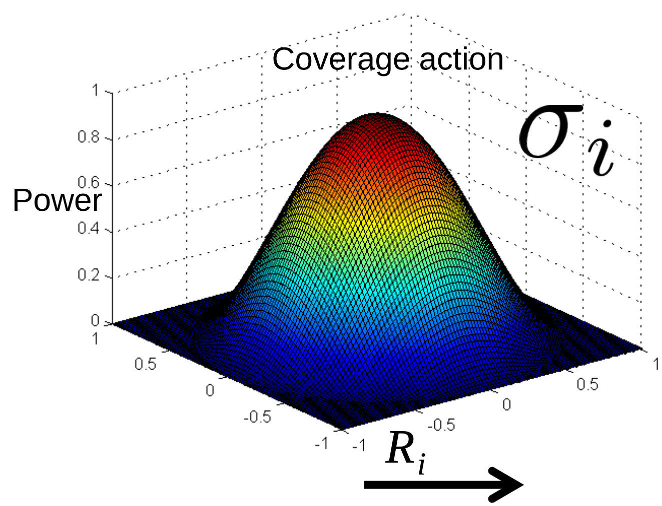


Coverage action control

- Coverage action
  - Developed in domain Ω_i of each agent
  - We consider circular actuators with range R_i
  - Coverage function:

$$\begin{cases} \alpha_i \geq 0 & if \quad r < R_i \quad (x \in \Omega_i) \\ \alpha_i = 0 & if \quad r \geq R_i \quad (x \notin \Omega_i) \end{cases}$$





σ_i = Shape of the coverage action

$\int_{\Omega_i} \sigma_i dx = 1$: Normalization of the shape of the coverage action

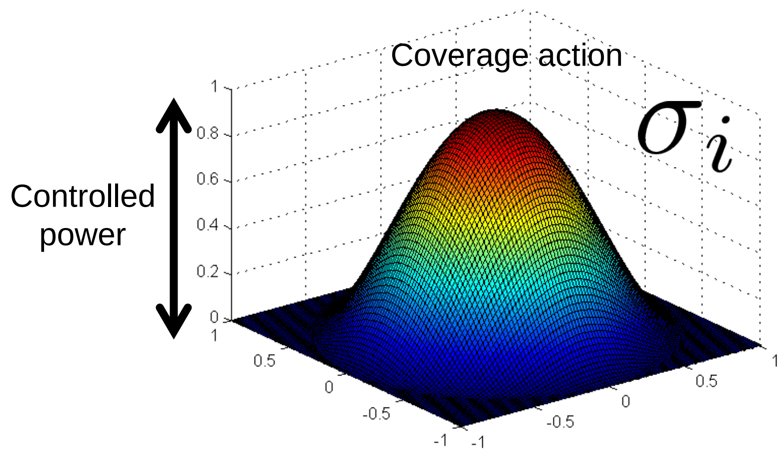
Coverage action control

- Variable coverage power control
  - Adaptive and efficient action
- Tries to reduce the error over the actuator area with an action α_i proportional C to the weighted error $\sigma_i \cdot (\Lambda^* - \Lambda)$ of the coverage domain of each agent Ω_i

$$\alpha_i = K \cdot \sigma_i$$

- Weighted error in the agent domain

$$K = C \cdot \left[\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \right]^{2 \cdot q - 1}$$



Coverage action control

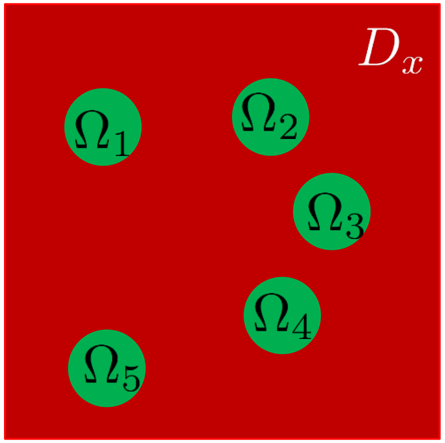
- Variable coverage power control
  - Evolution of the error with time:

$$\frac{de_{D_x}}{dt} = -2 \cdot \left[\int_{D_x} A \cdot \Phi \cdot \Lambda \cdot (\Lambda^* - \Lambda) dx + \int_{D_x} B \cdot \alpha_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right]$$

$$= -2 \cdot \left[\int_{D_x} A \cdot \Phi \cdot \Lambda \cdot (\Lambda^* - \Lambda) dx + K \cdot \left(\int_{\Omega_i} B \cdot \sigma_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right) \right]$$

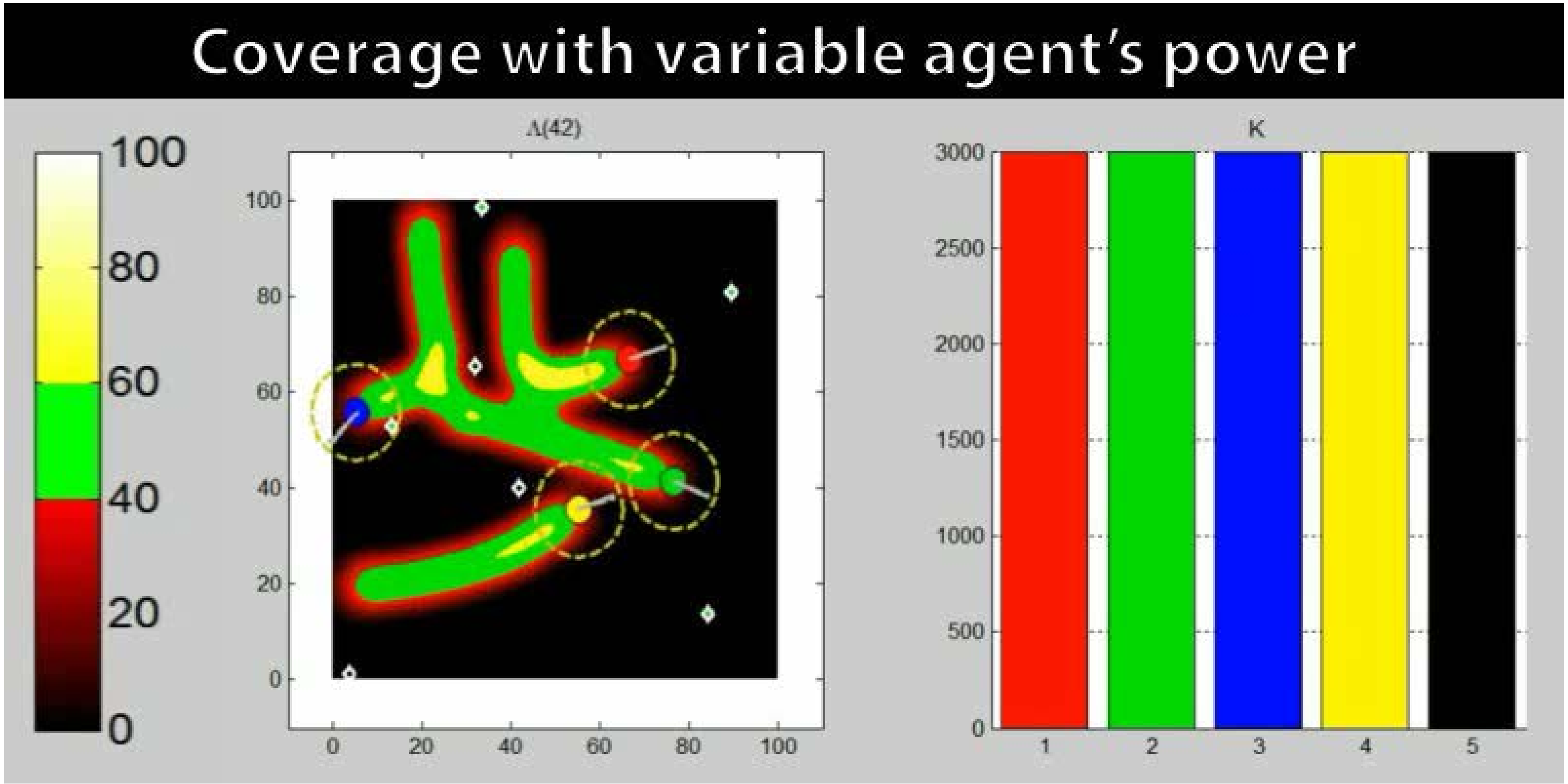
$$= -2 \cdot \left[\int_{D_x} A \cdot \Phi \cdot \Lambda \cdot (\Lambda^* - \Lambda) dx + C \cdot \left(\int_{\Omega_i} B \cdot \sigma_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx \right)^{2 \cdot q} \right]$$

- The coverage decays / vanishes and makes the error grow throughout the domain
- The design of the coverage action guarantees that in the actuator domain of agents error decreases



Coverage action control

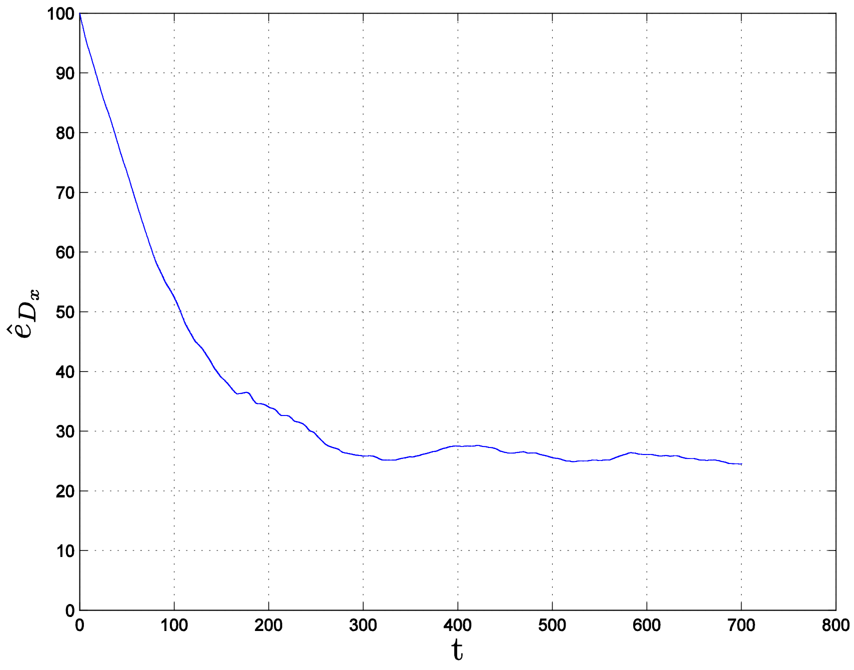
- Variable coverage power control



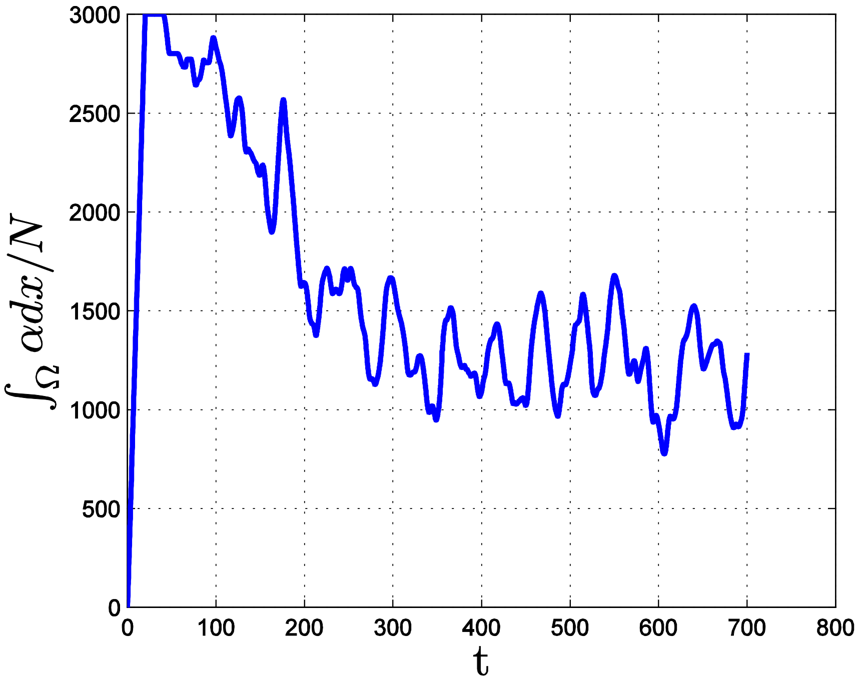
Coverage action control

- Variable coverage power control

Error of the domain



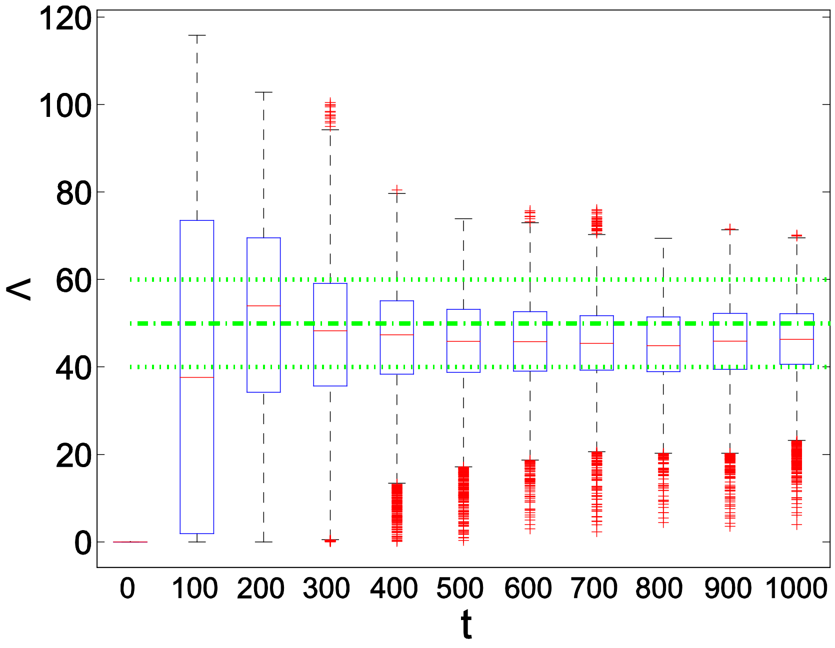
Average coverage power



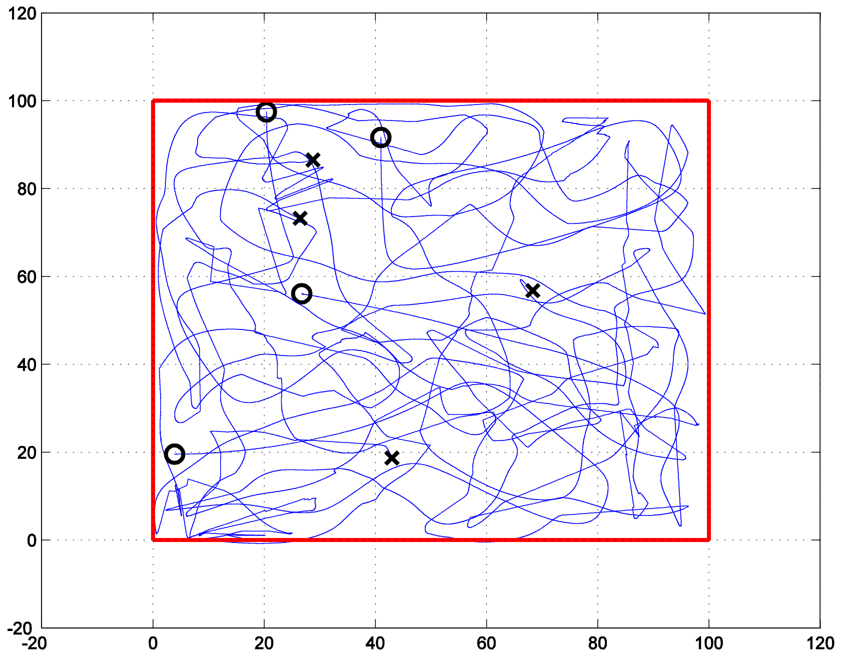
Coverage action control

- Variable coverage power control

Boxplot along time of the coverage level

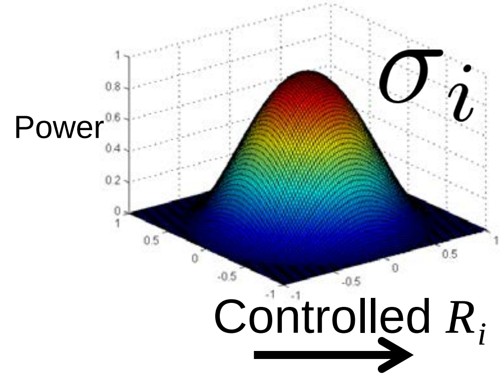


Agent's paths



Coverage action control

- Variable coverage range control
  - Range of the actuator: R_i



  - Taking the second derivative over time of the error

$$\frac{\partial^2 e_{D_x}}{\partial t^2} = -2\left[\int_{D_x} A \cdot \Phi \cdot \frac{\partial \Lambda}{\partial t} \cdot (\Lambda^* - 2 \cdot \Lambda) dx \right.$$
$$\left. + 2 \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx\right)^{2 \cdot q - 1} \cdot \int_{\Omega_i} B \cdot \Phi \cdot \left(\frac{\partial \sigma_i}{\partial R} \cdot \frac{\partial R}{\partial t} \cdot (\Lambda^* - \Lambda) - \sigma_i \cdot \frac{\partial \Lambda}{\partial t}\right) dx\right]$$

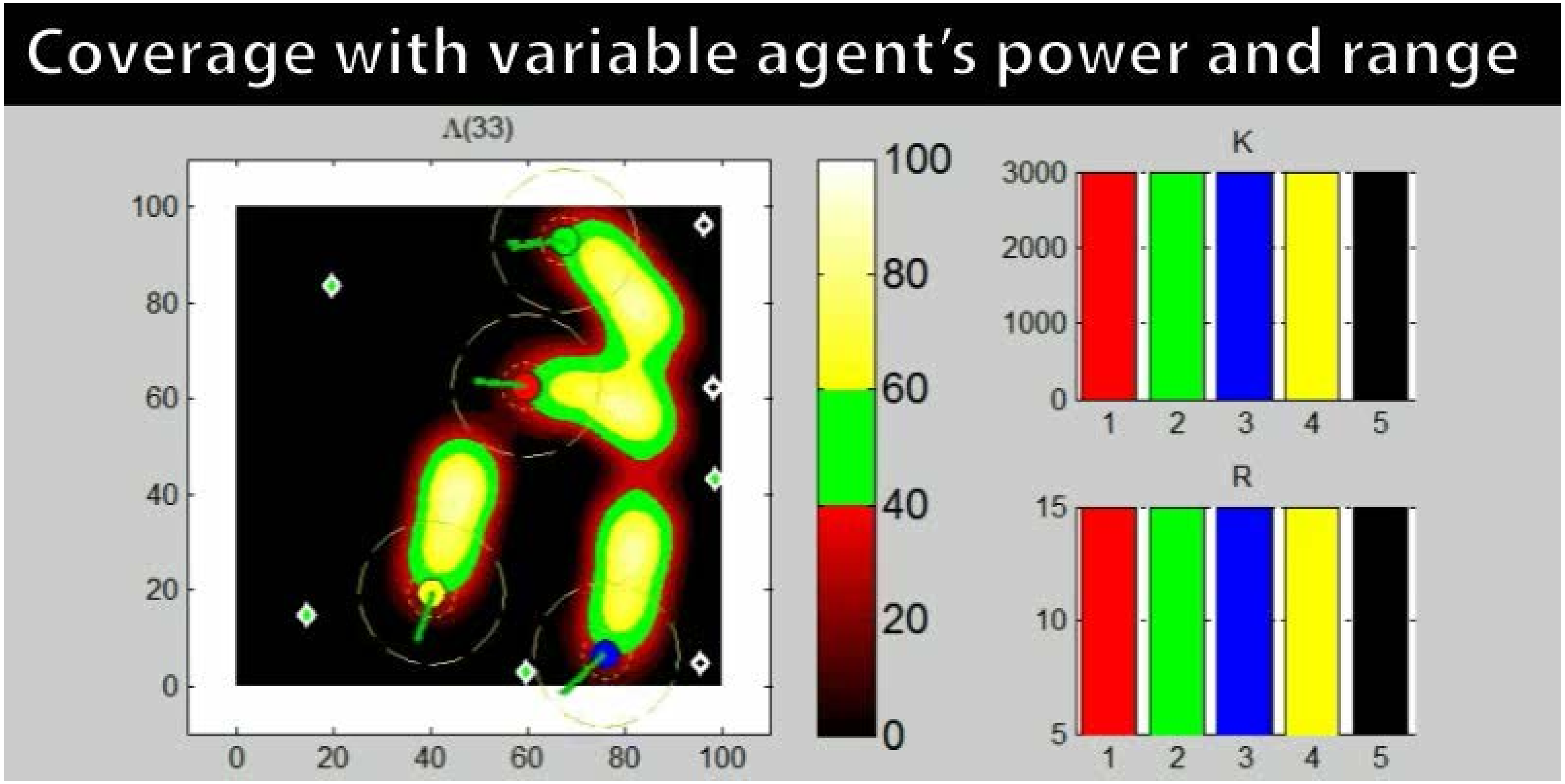
  - Design a range control that makes the second of the addend always negative

$$\frac{\partial R}{\partial t} = k_i^R \int_{\Omega_i} B \cdot \Phi \cdot \sigma_i \cdot (\Lambda^* - \Lambda) dx \cdot \int_{\Omega_i} B \cdot \Phi \cdot \frac{\partial \sigma_i}{\partial R} \cdot (\Lambda^* - \Lambda) dx,$$

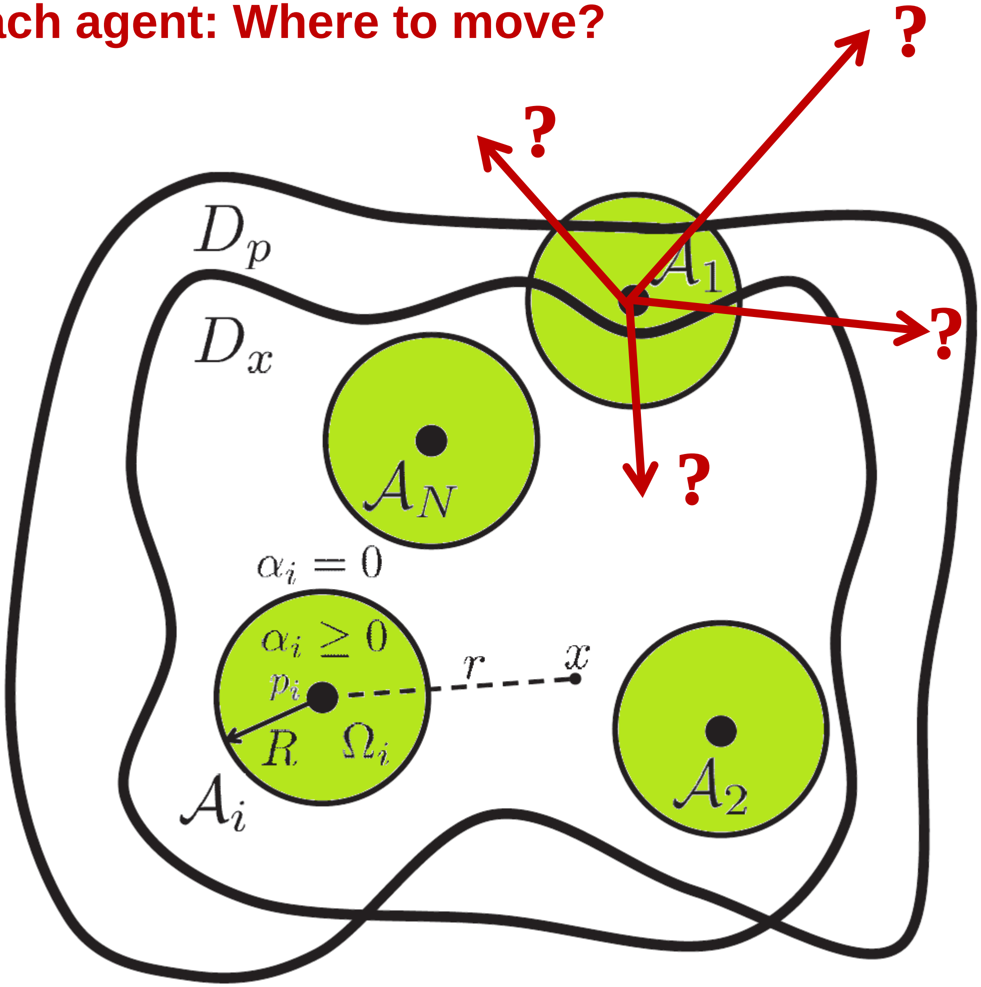
  - The idea is to make the second derivative of the error as low as possible to reduce the coverage error

Coverage action control

- Variable coverage range control



- Motion action control
  - The objective of the motion control law is to keep decreasing the error
    - **For each agent: Where to move?**

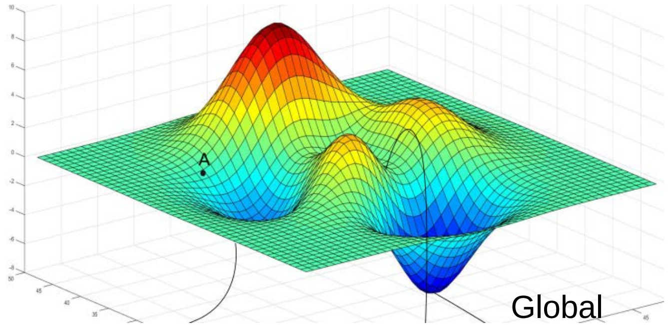


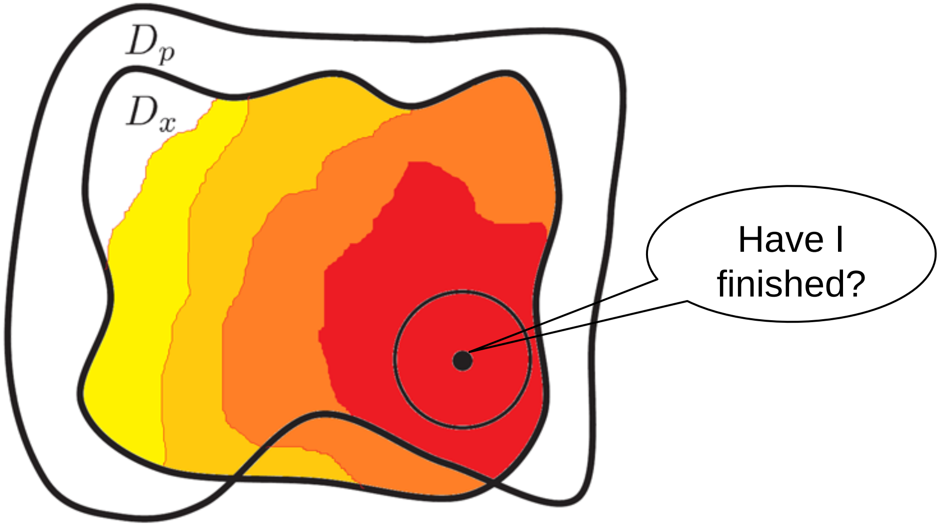
Motion control

- Motion action control
  - Local control law based on gradient of the error with respect to agent position:

$$u_i^{loc}(t) = \frac{\partial}{\partial p_i}\left(\frac{\partial e_{D_x}}{\partial t}\right) = -4 \cdot q \cdot C \cdot \left(\int_{\Omega_i} B \cdot \sigma_i \cdot \Phi \cdot (\Lambda^* - \Lambda) dx\right)^{2 \cdot q - 1} \cdot \int_{\Omega_i} B \cdot \Phi \cdot \frac{\partial \sigma_i}{\partial r} \cdot \frac{p_i - x}{\|p_i - x\|} \cdot (\Lambda^* - \Lambda) dx$$

    - However, a local control law is known to get stuck in local minima





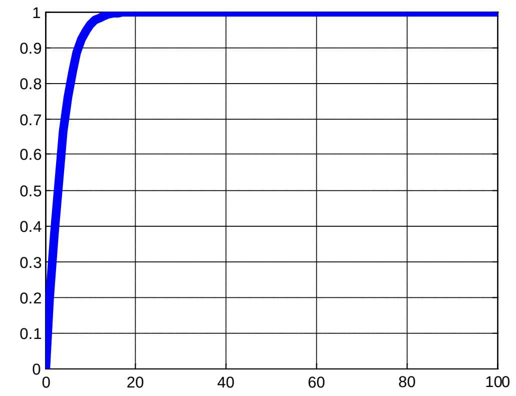
Motion control

- Motion action control
  - Global control law to avoid local minima tries to reach global targets

$$u_i^{glo} = k_i^G \cdot \frac{p_i - p_i^{obj}}{\|p_i - p_i^{obj}\|}$$

$$k_i^G = tanh\left(\frac{2 \cdot d_i^{obj}}{R}\right)$$

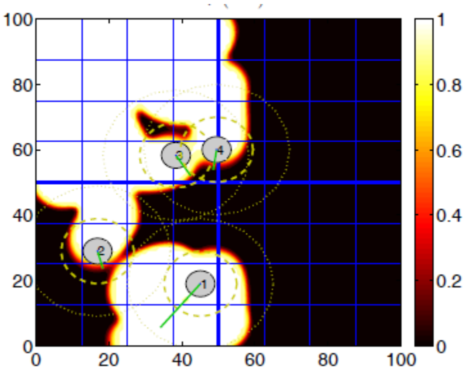
k_i^G



$\|p_i - p_i^{obj}\|$

- Selection of global objectives
  - Hierarchical grid of the domain
  - Blob analysis of the domain

$$p_i^{obj}$$

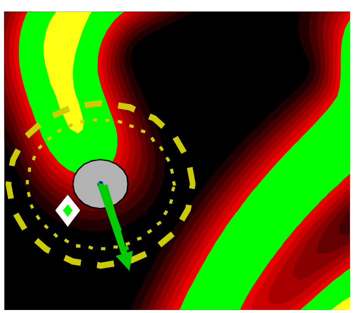


- Motion action control
  - Combination of local control law and global control law
  - Based on the coverage error of the actuator domain Ω_i we introduce a normalized local error

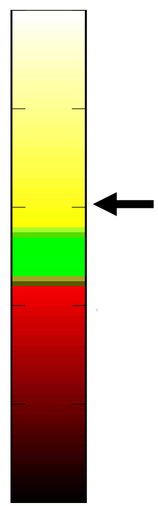
$$\varsigma_{\Omega_i} = \int_{\Omega_i} \Phi \cdot \sigma_i \cdot \frac{(\Lambda^* - \Lambda)}{\Lambda^*} dx \qquad \varsigma_{\Omega_i}(t) \in (-\infty, 1]$$

  - Indicates that agent's neighborhood is satisfactorily covered when it is negative or 0

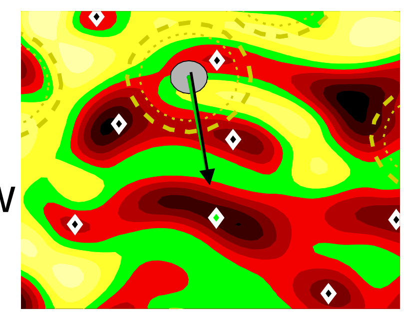
$$u_i^{cov} = (\varsigma_{\Omega_i}^+)^\beta \cdot \hat{u}_i^{loc} + (1 - (\varsigma_{\Omega_i}^+)^\beta) \cdot u_i^{glo} \qquad \varsigma_{\Omega_i}^+ = \max(0, \varsigma_{\Omega_i})$$



High local error agent obeys **local** control law



Low local error agents obeys **global** control law



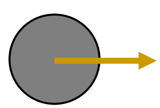
Motion control

- Motion action control
  - Finally, the velocity control law

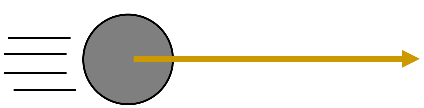
$$u_i = k_i \cdot (1 - \varsigma^+_{\Omega_i}) \cdot u_i^{cov}$$

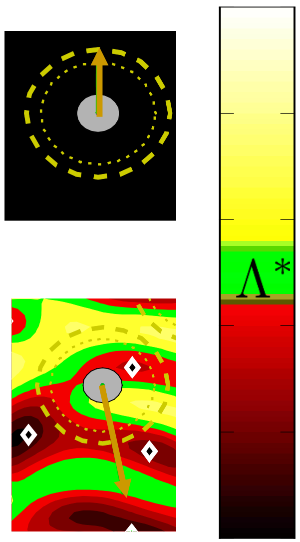
    - Bounded by the motion gain k_i

  - Based on the coverage error of the actuator domain Ω_i
    - High error: slow down and develop coverage carefully



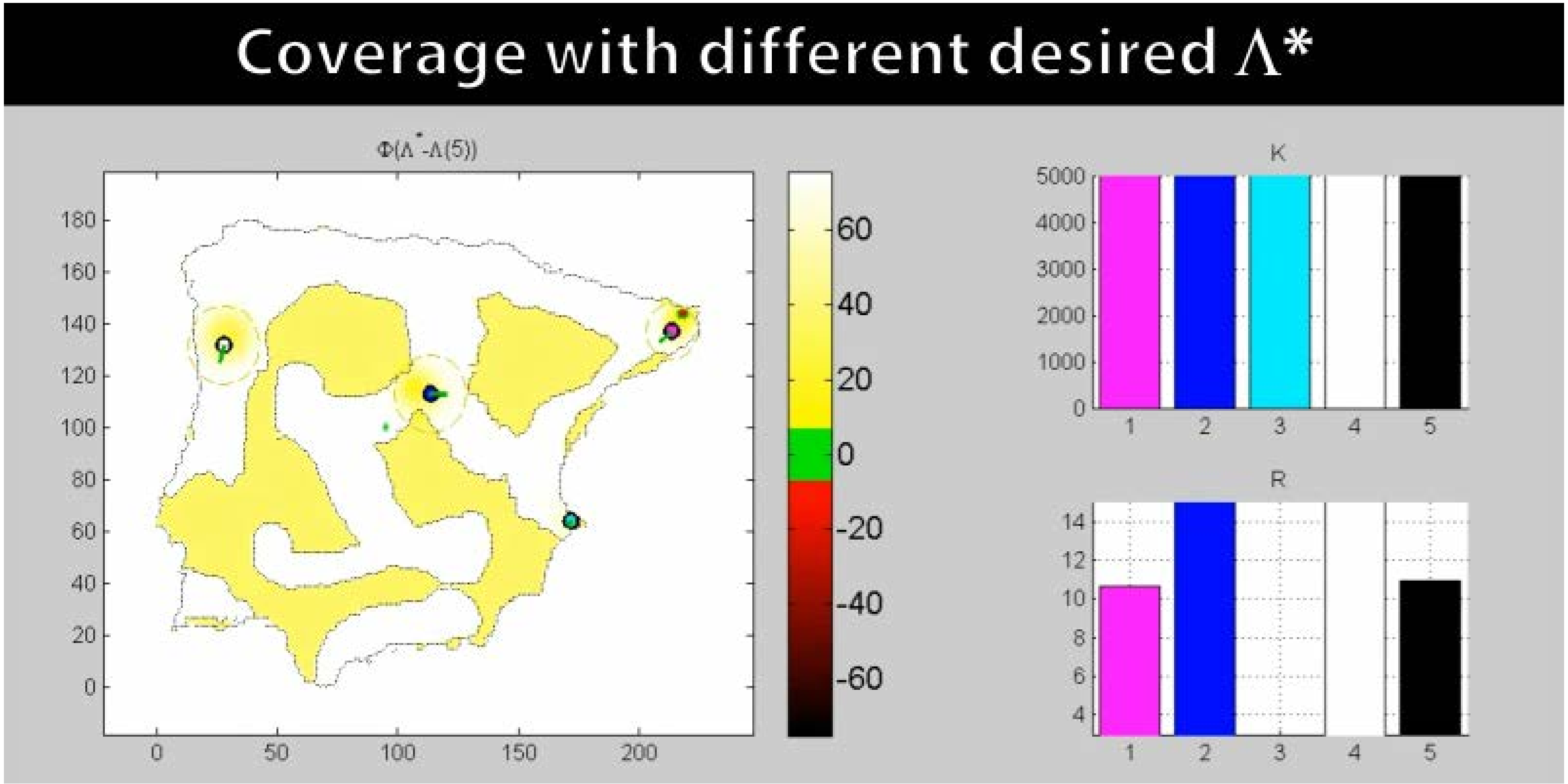
    - Low error: speed up to reach zones with higher error





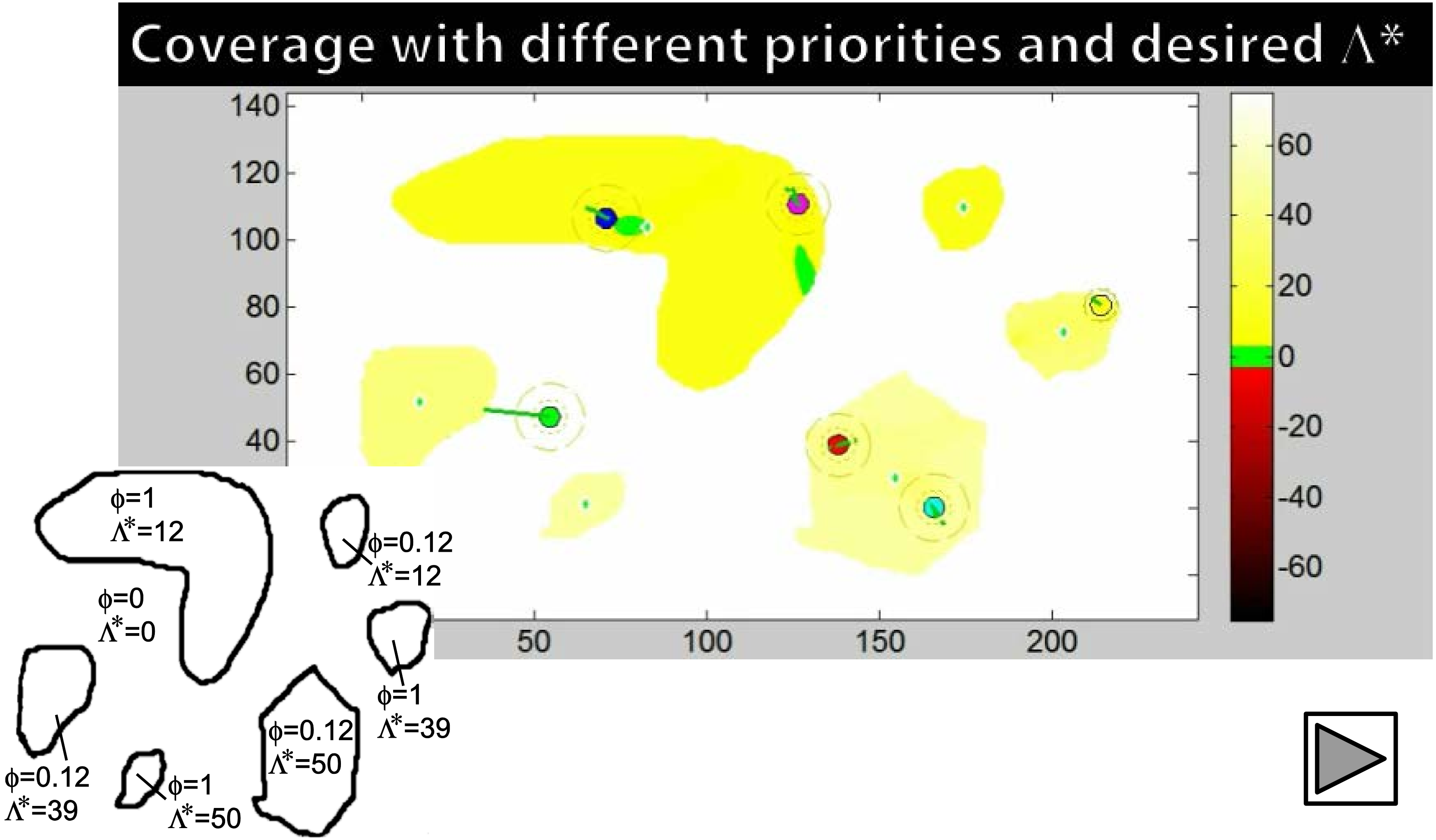
Motion control

- Example



Motion control

- Example



Bibliography

- Some additional topics
  - Anisotropic sensors
  - Purely distributed algorithms
  - Collision avoidance

Bibliography

- C. Franco, G. López-Nicolás, C. Sagüés, S. Llorente. Persistent coverage control with variable coverage action in multi-robot environment. IEEE Conference on Decision and Control, 2013
- C. Franco, G. López-Nicolás, C. Sagüés, S. Llorente. Adaptive action for multi-agent persistent coverage. Asian Journal of Control, vol. 18, no. 2, pp. 419-432, 2016
- C. Franco, D. M. Stipanovic, G. López-Nicolás, C. Sagüés, S. Llorente. Persistent coverage control for a team of agents with collision avoidance. European Journal of Control, vol. 22, pp. 30-45, 2015