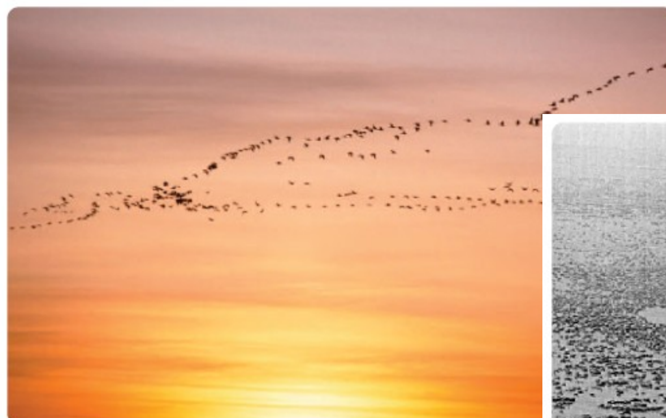
Structure and Synthesis of Robot Motion

Motion Synthesis in Groups and Formations I

Subramanian Ramamoorthy
School of Informatics

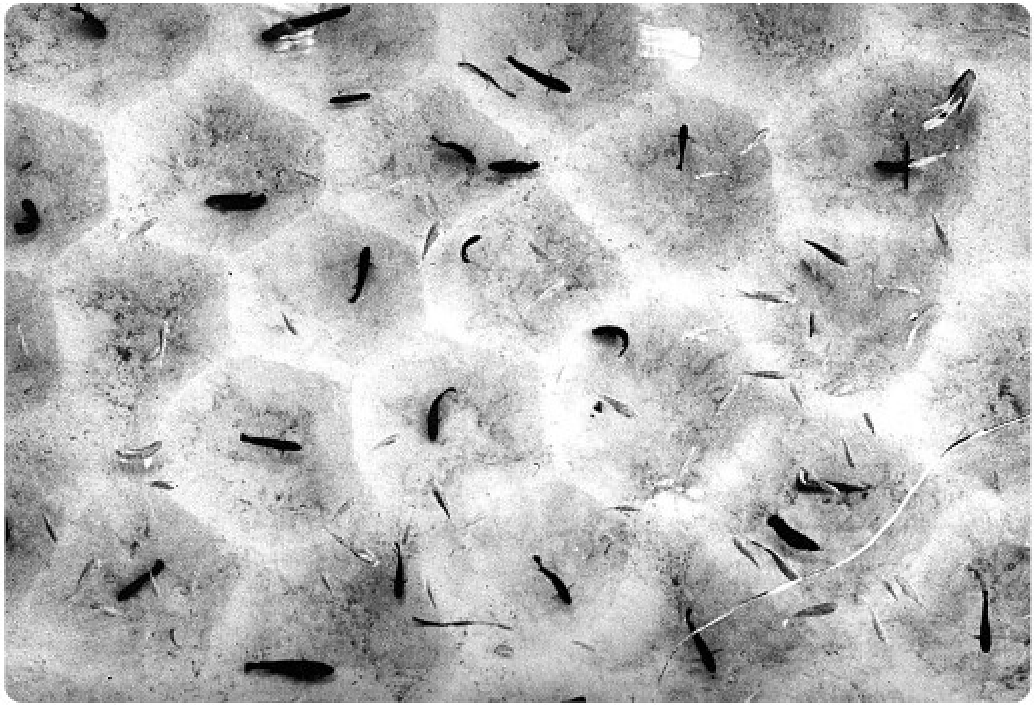
5 March 2012

Consider Motion Problems with Many Agents



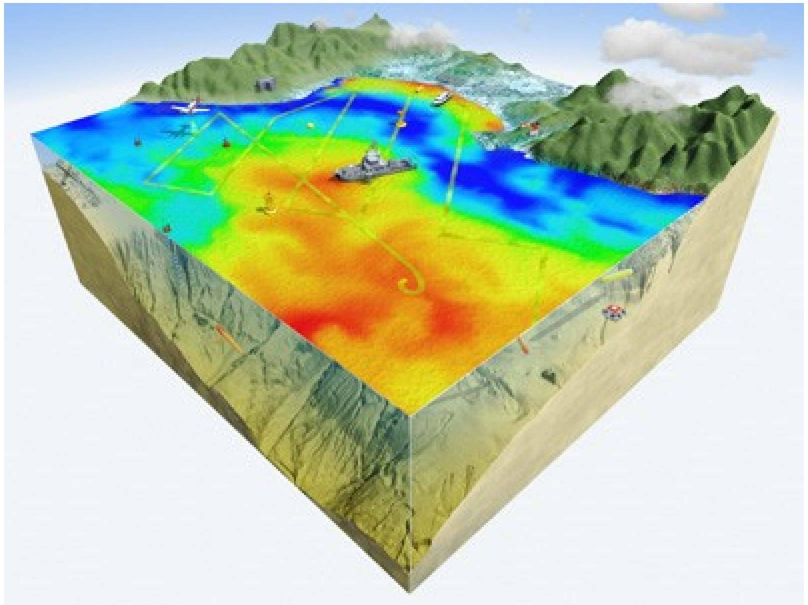
How should we model and solve these problems?

Interesting Effects in Such Groups



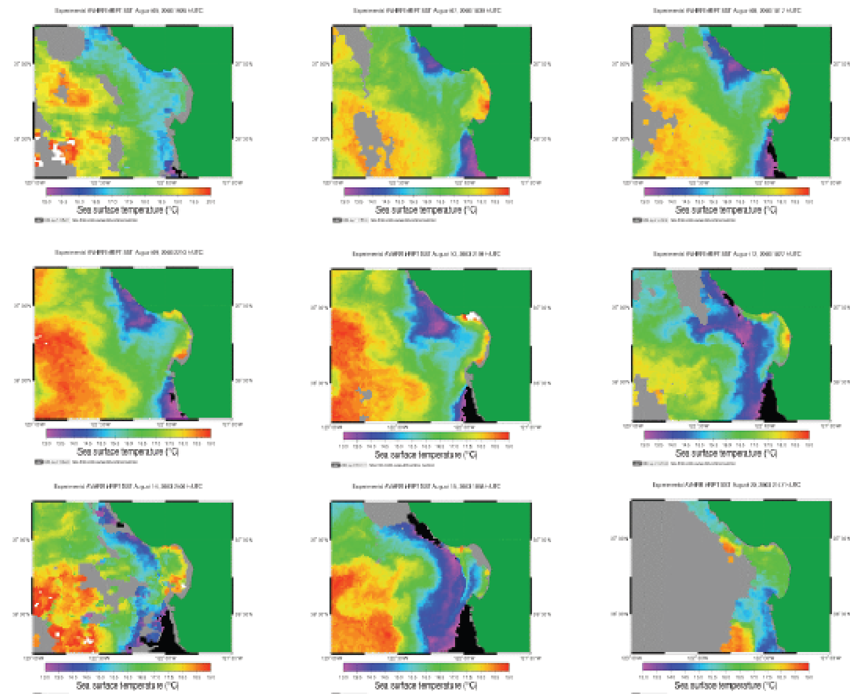
Territorial behaviour in tilapia mossambica
- No single fish has this global view so what do they do?

Increasingly Important within Robotics



N.E. Leonard et al., Collective Motion, Sensor Networks, and Ocean Sampling, *Proc. IEEE* 95(1):48–74, 2007.

The Autonomous Ocean Sampling Network

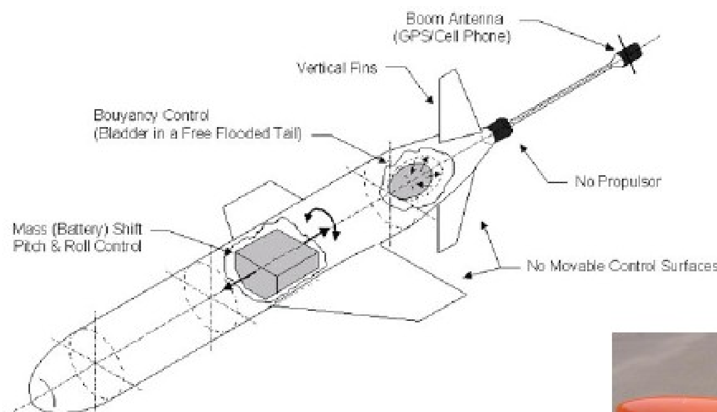


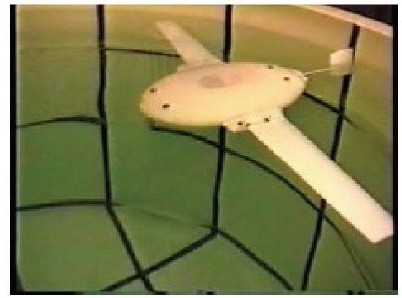
Satellite Sea Surface Temperature (SST),
Monterey Bay, CA, Aug 5-20, 2003

The Robots

Buoyancy Driven Gliders

- Autonomous vehicles with very long range (i.e., lifetime)





ROGUE

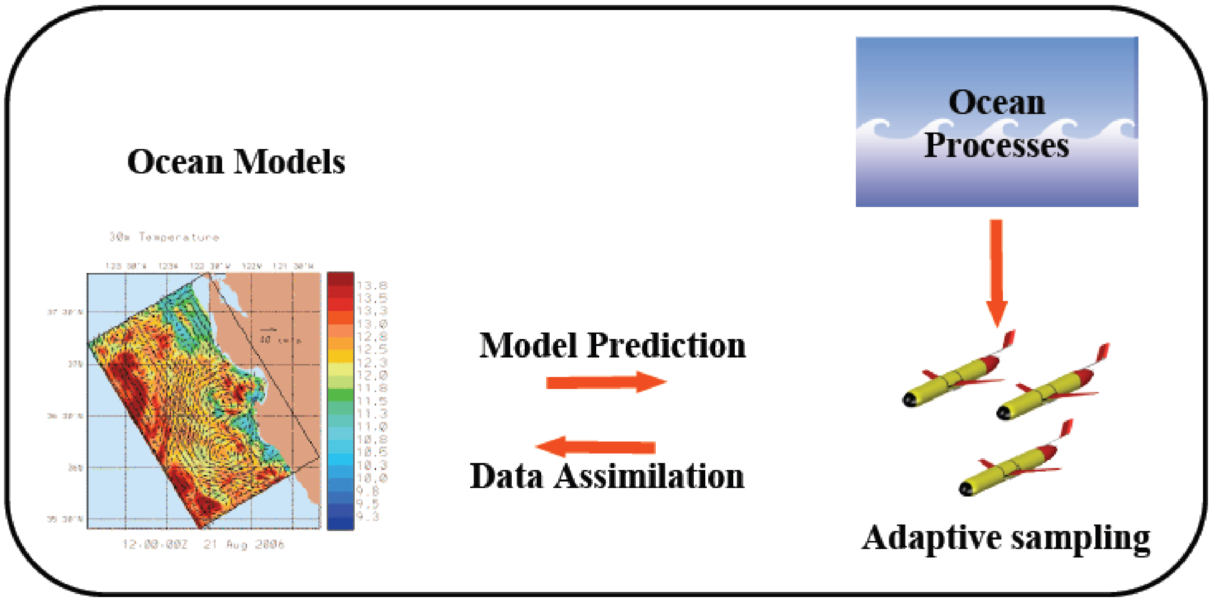
Slocum Glider

Spray Glider

Scientific Goals

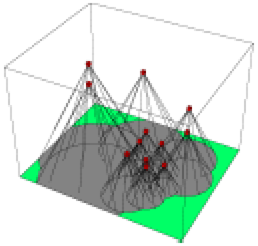
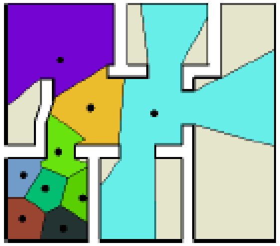
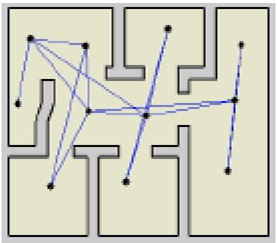
Sustainable, portable, adaptive, coupled observation/modeling system.

Learn how to deploy, direct and utilize autonomous vehicles most efficiently to sample the ocean, assimilate the data into numerical models in real time, and predict future conditions with minimal error.



Cooperative Multi-agent Systems

- What kind of systems?
  - Groups of agents with control, sensing, communication and computing
- Each individual
  - senses its immediate environment
  - communicates with others
  - processes information gathered
  - takes local action in response

Motivation: Decision Making in Animals

- Able to
  - deploy over a given region
  - assume specified pattern
  - rendezvous at a common point
  - jointly initiate motion/change direction in a synchronized way

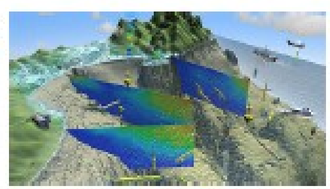
- Species achieve synchronized behavior
  - with limited sensing/communication between individuals
  - without apparently following group leader

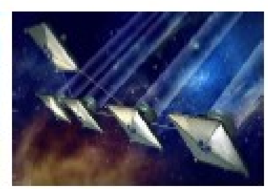
(Couzin et al, Nature 05; Conradt et al, Nature 03)

Motivation: Engineered Multi-agent Systems

- Embedded robotic systems and sensor networks for
  - high-stress, rapid deployment — e.g., disaster recovery networks
  - distributed environmental monitoring — e.g., portable chemical and biological sensor arrays detecting toxic pollutants
  - autonomous sampling for biological applications — e.g., monitoring of species in risk, validation of climate and oceanographic models
  - science imaging — e.g., multi-spacecraft distributed interferometers flying in formation to enable imaging at micro-arcsecond resolution


Sandia National Labs


MBARI AOSN


NASA Terrestrial Planet Finder

Challenges

- What useful engineering tasks can be performed with limited-sensing/communication agents?
  - Feedback: rather than open-loop computation for known/static setup
  - Information flow: who knows what, when, why, how, dynamically changing
  - Reliability/performance: robust, efficient, predictable behavior

- How to coordinate individual agents into coherent whole?

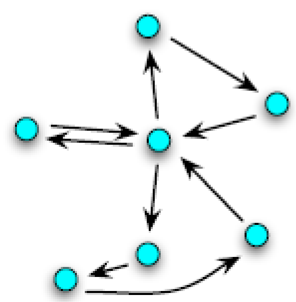
Objective: systematic methodologies to design and analyze cooperative strategies to control multi-agent systems

Integration of control, communication, sensing, computing

Notion of Distributed Algorithms

- Simplest distributed iteration is linear averaging:
  - you are given a graph
  - each node contains a value x_i
  - each node repeatedly executes:

$$x^+_i := average(x_i, \{x_j \text{ , for all neighboring } j\})$$



Why does this algorithm converge and to what?

Idea for Why Distributed Averaging Works

- If you define the pair-wise disagreement as a contributor to an ‘energy’
- then the iterations act to minimize the energy

$$\Phi_G(x) = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(x_j - x_i)^2$$

- Many ways to understand such processes:
  - Matrix Analysis (Perron-Frobenius theory)
  - Algebraic Graph Theory (Laplacians and flows)

Recap: Potential Function

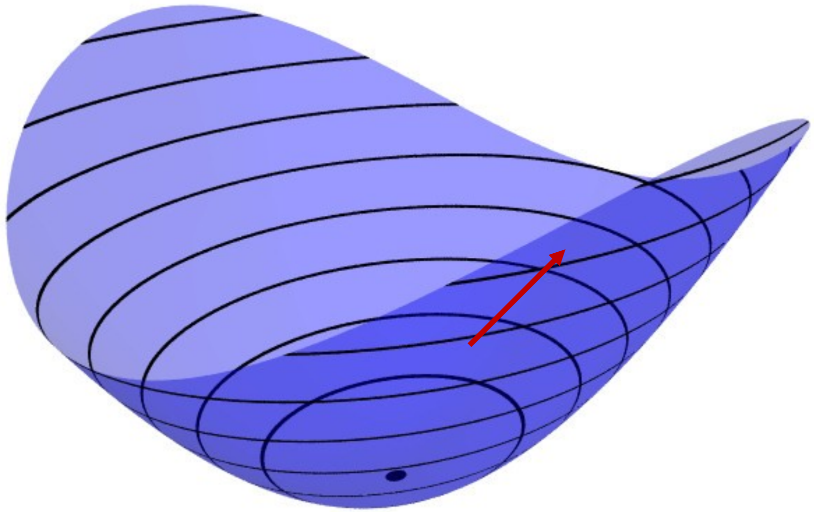
- Differentiable real-valued function,

$$U : \mathbb{R}^m \mapsto \mathbb{R}$$

- Treat the value as 'energy'
- Then, gradient is the vector,

$$\nabla U(q) = DU(q)' = [\frac{\partial U}{\partial q_1}(q), \ldots, \frac{\partial U}{\partial q_m}(q)]'$$

- The gradient points in the direction that locally maximally increases U

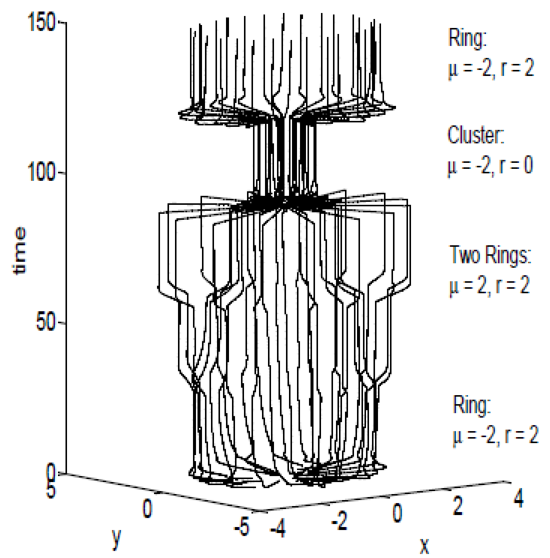


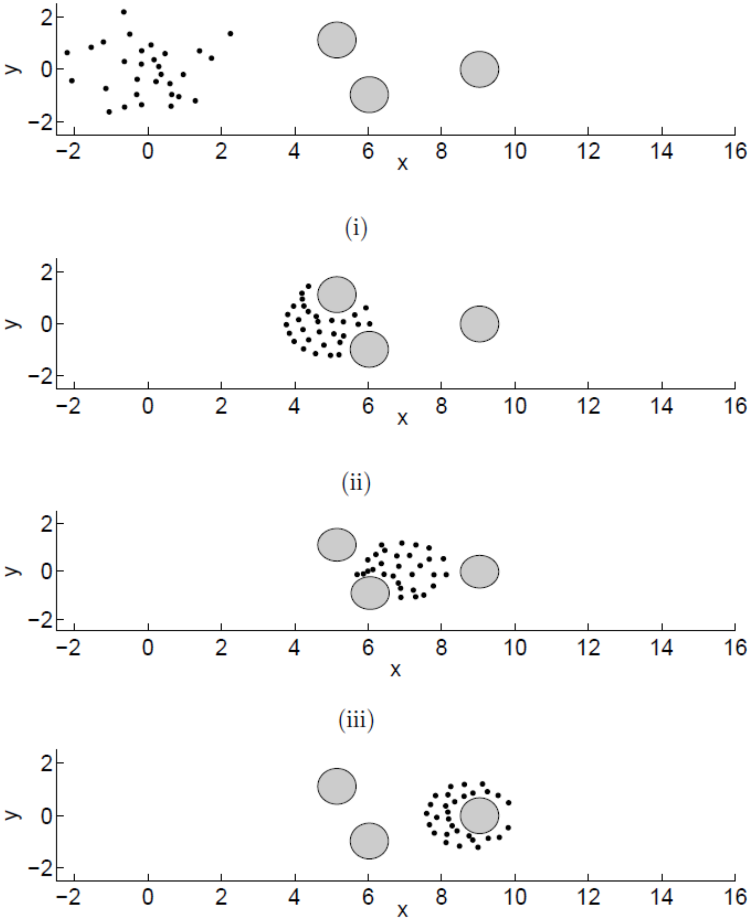
We could use such an encoding for the motion of the group. How exactly do we encode?

Potential for Steering to a 'Ring'

'Tune' formation on to a ring:

$$U^S(\mathbf{x}_i;\mu,\alpha) = -\frac{1}{2}\mu(\rho_i - r)^2 + \frac{1}{4}(\rho_i - r)^4 + \frac{1}{2}\alpha z_i^2$$





Behaviour of Ring Potential Function

- Define: $$D = \frac{\partial^2 U}{\partial \rho_i^2}\frac{\partial^2 U}{\partial z_i^2} - \left[\frac{\partial^2 U}{\partial \rho_i \partial z_i}\right]^2$$

- Conditions for stability:

(i) $D > 0$, $\partial^2 U/\partial \rho_i^2 > 0 \implies$ equilibrium point is a stable minimum.
(ii) $D > 0$, $\partial^2 U/\partial \rho_i^2 < 0 \implies$ equilibrium point is an unstable maximum.
(iii) $D < 0 \implies$ equilibrium point is a saddle.

$$\frac{\partial^2 U}{\partial \rho_i^2} = -\mu + 3(\rho_i - r)^2$$

$$\frac{\partial^2 U}{\partial z_i^2} = \alpha$$

$$\frac{\partial^2 U}{\partial \rho_i \partial z_i} = 0$$

Dynamics Under Steering Potential

- The induced equations of motion:

$$\begin{pmatrix} \dot{\mathbf{x}}_i \\ \dot{\mathbf{v}}_i \end{pmatrix} = \begin{pmatrix} \mathbf{v}_i \\ -\sigma \mathbf{v}_i - \nabla_i U^S(\mathbf{x}_i) \end{pmatrix} = \begin{pmatrix} f(\mathbf{x}_i, \mathbf{v}_i) \\ g(\mathbf{x}_i, \mathbf{v}_i) \end{pmatrix}$$

- And its local linearization:

$$\begin{pmatrix} \delta\dot{\mathbf{x}}_i \\ \delta\dot{\mathbf{v}}_i \end{pmatrix} = \mathbf{J} \begin{pmatrix} \delta\mathbf{x}_i \\ \delta\mathbf{v}_i \end{pmatrix}$$

where,

$$\mathbf{J} = \left.\begin{pmatrix} \frac{\partial}{\partial \mathbf{x}_i}(f(\mathbf{x}_i, \mathbf{v}_i)) & \frac{\partial}{\partial \mathbf{v}_i}(f(\mathbf{x}_i, \mathbf{v}_i)) \\ \frac{\partial}{\partial \mathbf{x}_i}(g(\mathbf{x}_i, \mathbf{v}_i)) & \frac{\partial}{\partial \mathbf{v}_i}(g(\mathbf{x}_i, \mathbf{v}_i)) \end{pmatrix}\right|_{\mathbf{x}_o, \mathbf{v}_o}$$

$$\mathbf{J} = \left.\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{\partial^2 U}{\partial \rho_i^2} & -\frac{\partial^2 U}{\partial \rho_i \partial z_i} & -\sigma & 0 \\ -\frac{\partial^2 U}{\partial \rho_i \partial z_i} & -\frac{\partial^2 U}{\partial z_i^2} & 0 & -\sigma \end{pmatrix}\right|_{\mathbf{x}_o, \mathbf{v}_o}$$

Analyze eigenvalues to ensure stability

Dynamics Under Steering Potential

- To understand 'nonlinear' stability, consider a Lyapunov function based on total energy

$$L = \sum_i \left(\frac{1}{2} |\mathbf{v}_i|^2 + U^S(\mathbf{x}_i) \right)$$

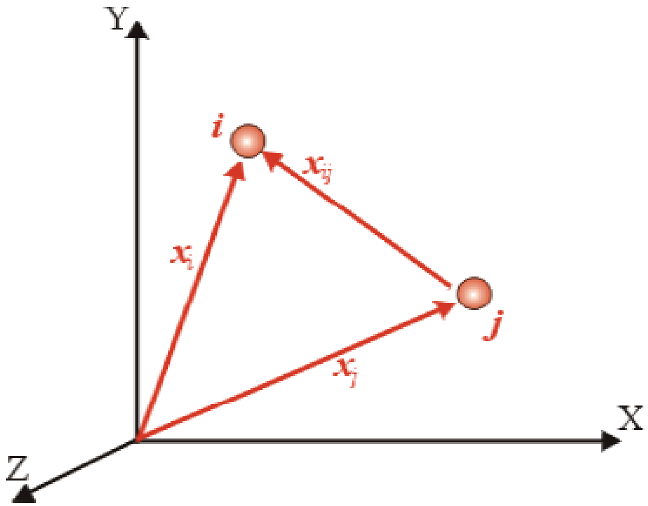
- If this function is monotone decreasing, one can be assured of stability over time, towards a fixed point

$$\frac{dL}{dt} = \left(\frac{\partial L}{\partial \mathbf{x}_i} \right) \dot{\mathbf{x}}_i + \left(\frac{\partial L}{\partial \mathbf{v}_i} \right) \dot{\mathbf{v}}_i$$

$$\frac{dL}{dt} = -\sigma \sum_i |\mathbf{v}_i|^2 \leqslant 0$$

We only discussed the base case. More details for obstacles but same technique.

Coordinated Control in the Ocean Robots

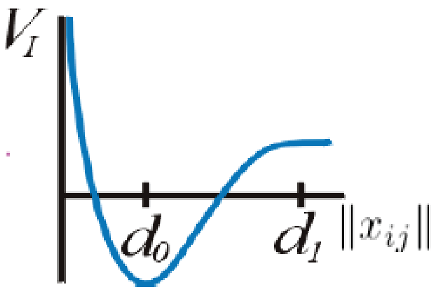


N vehicles with fully actuated dynamics:

$$\ddot{x}_i = u_i, \quad x_i \in \mathbb{R}^3$$
$$x_{ij} = x_i - x_j$$

Define potential $V_I(x_{ij})$ to reflect attraction between individuals when distant and repulsion when close (and little or no influence outside neighborhood).

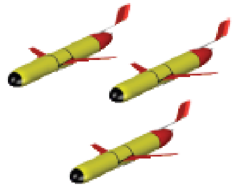
For example:



$$V_I(x_{ij}) = \frac{1}{2}k_s(||x_{ij}|| - d_0)^2 \quad \text{or} \quad V_I(x_{ij}) = k_s\left(\ln ||x_{ij}|| + \frac{d_0}{||x_{ij}||^2}\right).$$

$$u_i = -\sum_{j=1, j\neq i}^{N} F_I(x_{ij}) - k_d\dot{x}_i, \qquad -F_I(x_{ij}) = -\nabla V_I(x_{ij})$$

Stability of Formation



$$\ddot{x}_i = - \sum_{j=1, j\neq i}^{N} F_I(x_{ij}) - k_d \dot{x}_i.$$

Lyapunov function:

$$V = \frac{1}{2}\sum_{i=1}^{N} \dot{x}_i \cdot \dot{x}_i + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} V_I(x_{ij}).$$

$$\dot{V} \quad = \quad \sum_{i=1}^{N} \dot{x}_i \cdot \ddot{x}_i + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} F_I(x_{ij}) \cdot \dot{x}_{ij} = -k_d \sum_{i=1}^{N} ||\dot{x}_i||^2.$$

Implies stability of group in absence of additional external forces.
Equilibria have to satisfy

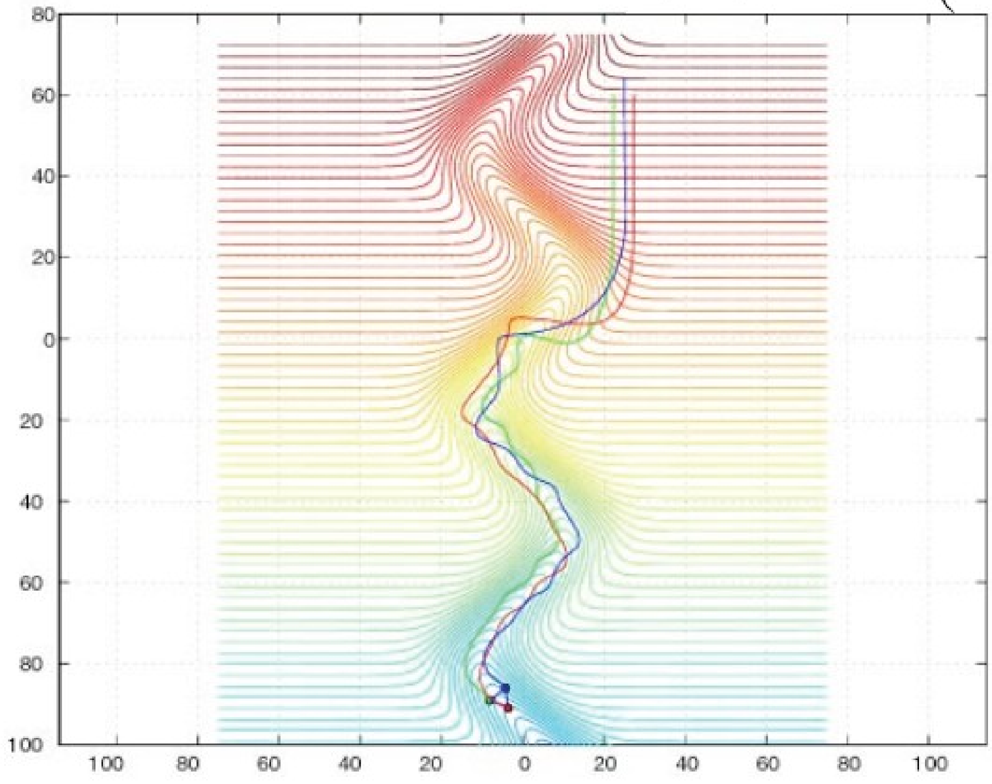
$$\sum_{j=1,\, j\neq i}^{N} F_I(x_{ij}) = 0, \quad i = 1, \ldots, N.$$

Guiding Vehicles through a Field: Coordinated Control + Projected Gradient

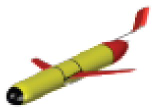
Feedback from artificial potentials and from measurements of environment:

$$T(x,y) = ay + b\left(1 - e^{-(x-\sin(y/10))^2}\right)$$

Vehicle group in descending Gaussian valley



What is going on?
Gradient Descent with 1 Vehicle



Gradient field is $T : \mathbb{R}^2 \to \mathbb{R}$

For single vehicle case with $\ddot{x} = u$, let

$$u = -k_d \dot{x} - \kappa \nabla T(x)$$

$k_d > 0$ and κ are constant scalar gains.

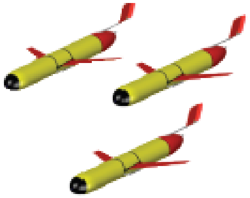
If $T(x)$ has strict minimum at origin, asymptotic stability of origin is proved with Lyapunov function

$$V(x, \dot{x}) = \frac{1}{2} \dot{x} \cdot \dot{x} + \kappa T(x)$$

with $\kappa > 0$.

If origin is unique, global minimum, then global asymptotic stability results possible.

Gradient Descent with Many Vehicles



Consider this control law plus interaction forces $F_I(x_{ij})$ on group of N vehicles:

$$\boxed{\ddot{x}_i = -\kappa \nabla T(x_i) - k_d \dot{x}_i - \sum_{j=1, j\neq i}^{N} F_I(x_{ij}).}$$

Lyapunov function:

$$V = \frac{1}{2} \sum_{i=1}^{N} (\dot{x}_i \cdot \dot{x}_i + \kappa T(x_i)) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V_I(x_{ij}).$$

$$\dot{V} \quad = \quad \sum_{i=1}^{N} \dot{x}_i \cdot \ddot{x}_i + \kappa \sum_{i=1}^{N} \nabla T(x_i) \cdot \dot{x}_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} F_I(x_{ij}) \cdot \dot{x}_{ij}. = -k_d \sum_{i=1}^{N} ||\dot{x}_i||^2.$$

Prop. An equilibrium corresponding to a located formation that is a strict minimizer of $\kappa \sum_{i=1}^{N} T(x_i) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V_I(x_{ij})$ will be a stable equilibrium for the coupled dynamics.

What **kinds** of problems do we have?

- Deployment: area coverage
- Rendezvous: coordinate steering across the group
- Target Tracking: of an entity in the environment
- Pursuit: Jointly go after some entity in the environment
- Boundary Estimation: Where am I in the formation?
- Averaging, Leader Election, etc. etc.

Is there a common framework for all such problems?

Note: We are not <u>yet</u> interested in multi-agent strategic interactions – happy to assume cooperation (but insist on decentralization)

Encoding Deployment

- Objective: maximize area within close range of mobile nodes
  - Let us assume a convex polygonal environment for now
  - Each node only has local measurements
  - We are often interested in scenarios where there is a weight over coverage (some spots are more important): density ϕ

- This is a distributed computation problem
  - Studied within parallel computing literature but assuming fixed network topology
  - We are interested in **mobile** agents that can fluidly re-deploy
  - So, better models may be obtained from computational geometry

How to Model Interconnection Topology?

Proximity graph

graph whose vertex set is a set of distinct points and whose edge set is a function of the relative locations of the point set

Let X be a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, with $d_1 + d_2 = d$. Let $\mathbb{G}(X)$ be the set of all undirected graphs whose vertex set is an element of $\mathbb{F}(X)$ (finite subsets of X)

A *proximity graph* $\mathcal{G} : \mathbb{F}(X) \to \mathbb{G}(X)$ associates to $\mathcal{P} = \{p_1, \ldots, p_n\} \subset X$ an undirected graph with vertex set $\mathcal{P}$ and edge set $\mathcal{E}_{\mathcal{G}}(\mathcal{P}) \subseteq \{(p, q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}$.

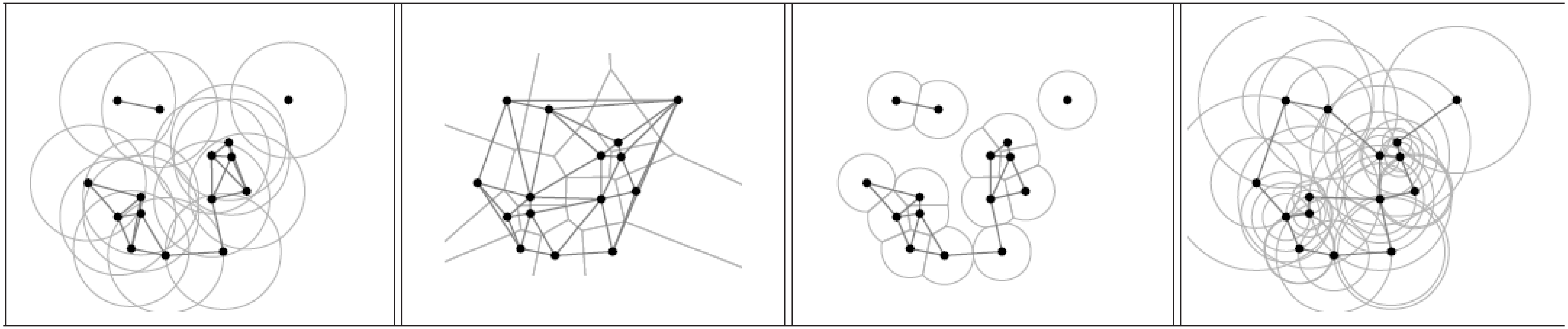
Examples of Proximity Graphs

- R-disk graph: Connect points within a ball
- Delaunay graph: connect if their Voronoi cells intersect

$$(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{D}}}(\mathcal{P}) \text{ if } V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$$

- R-limited Delaunay Graph: limit further by distance
- Relative Neighbourhood Graph: $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{RN}}}(\mathcal{P})$ if

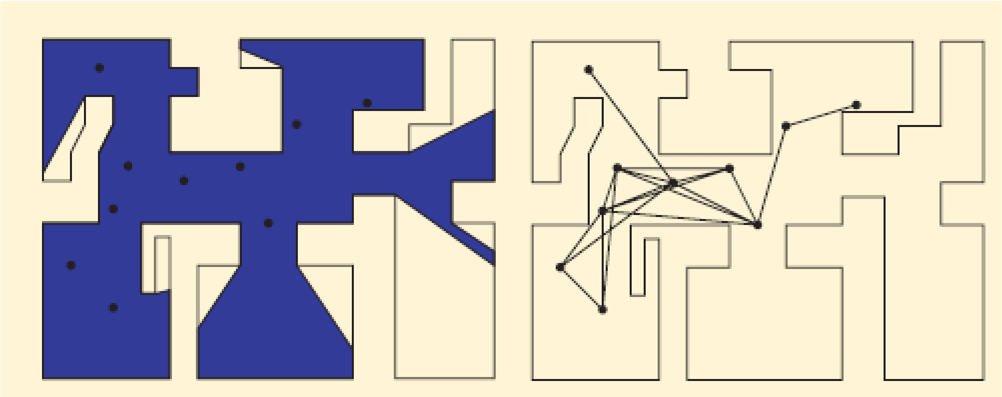
$$p_k \notin B(p_i, \mathrm{dist}(p_i, p_j)) \cap B(p_j, \mathrm{dist}(p_i, p_j)) \text{ for all } p_k \in \mathcal{P}$$



$\mathcal{G}_{\mathrm{disk}}(r)$ $\mathcal{G}_{\mathrm{D}}$ $\mathcal{G}_{\mathrm{LD}}(r)$ $\mathcal{G}_{\mathrm{RN}}$

Proximity Graph based on Visibility

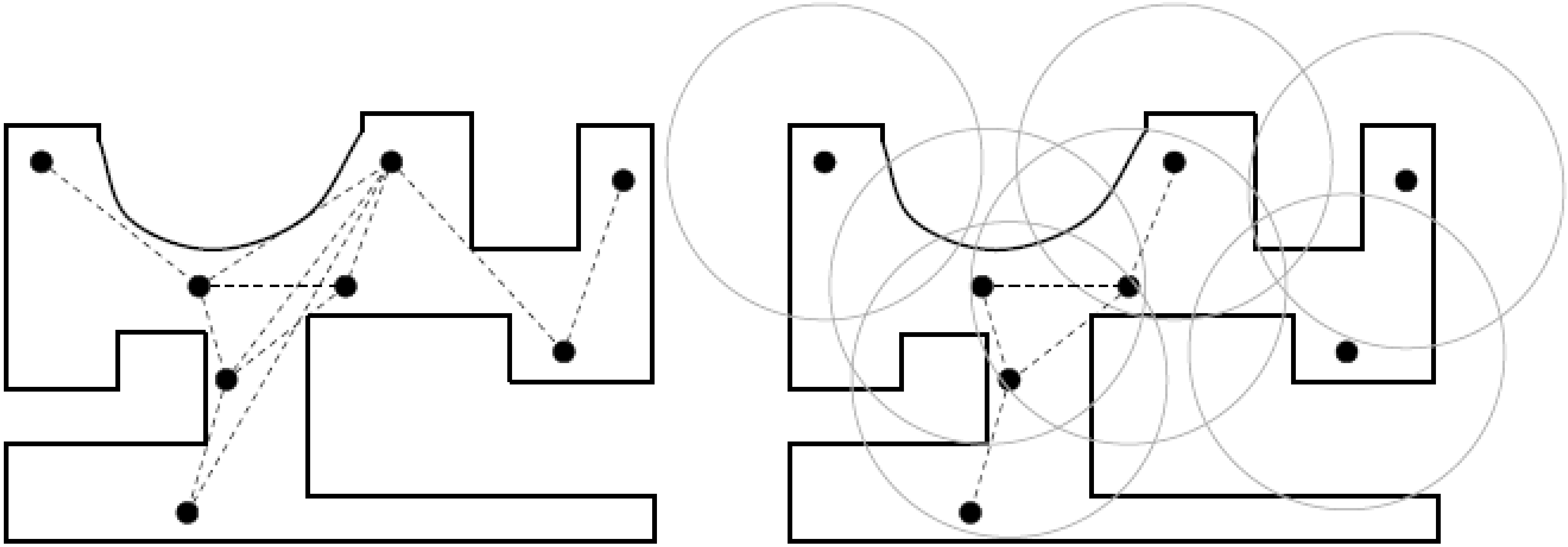
the *visibility graph* $\mathcal{G}_{\mathrm{vis},Q}$ in an allowable environment Q in $\mathbb{R}^2$, where two points are neighbors if they are visible to each other, that is, $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{vis},Q}}(\mathcal{P})$ if the closed segment $[p_i, p_j]$ from p_i to p_j is contained in Q.



This gives an induced neighbourhood structure based on the proximity graph:

$$\mathcal{N}_{\mathcal{G},p_i}(\mathcal{P}) = \{q \in \mathcal{P} \mid \{p_i, q\} \in \mathcal{E}_{\mathcal{G}}(\mathcal{P})\}\,.$$

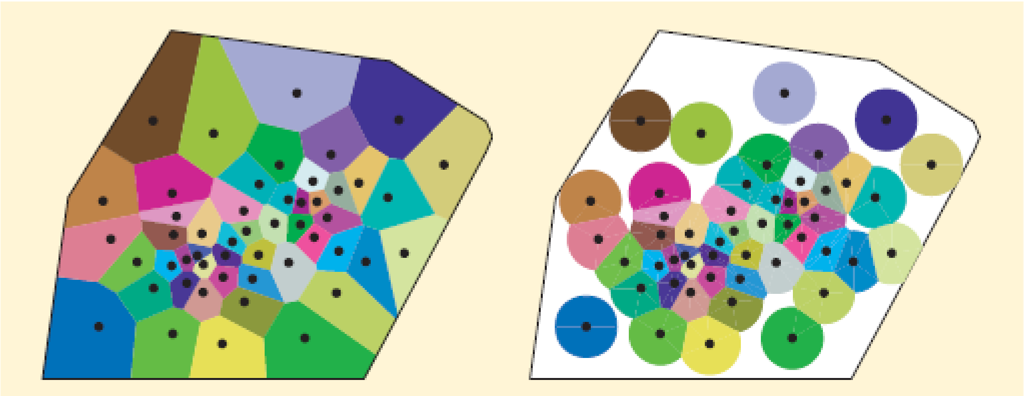
Visibility Graphs



(a) $\mathcal{G}_{\mathrm{vis},Q}$ (b) $\mathcal{G}_{\text{vis-disk},Q}$

The visibility and range-limited visibility graphs for 8 agents in an allowable environment. The geometric objects determining the edge relationship are plotted in light gray.

Partitioning the Polygonal Environment



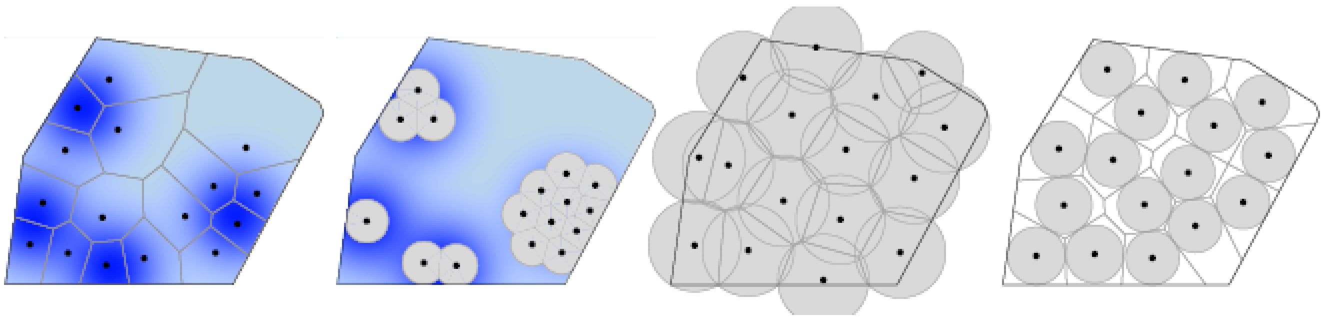
This is a Voronoi decomposition (right side has r-limited version).

$$\mathcal{P} = \{p_1, \ldots, p_n\} \subset \ S \subset \mathbb{R}^2$$

$$\downarrow$$

$$\{V_1(\mathcal{P}), \ldots, V_n(\mathcal{P})\} \text{ defined by } V_i(\mathcal{P}) = \{q \in S \mid \|q - p_i\| \leq \ \|q - p_j\| \text{ for all } p_j \in \mathcal{P}\}.$$

Further Types of Partitions



From left to right, centroidal, r-limited centroidal, circumcenter, and incenter Voronoi configurations composed by 16 points in a convex polygon. Darker blue-colored areas correspond to higher values of the density ϕ.

Why do this?

This gives us a sensible notion of neighbourhood.

A Question about Decentralization

When does a given proximity graph encode sufficient information to compute another proximity graph?

For instance, if a node knows the position of its neighbors in the complete graph (i.e., of every other node in the graph), then it is clear that the node can compute its neighbors with respect to any proximity graph.

Spatially Distributed Proximity Graph

- A proximity graph is spatially distributed w.r.t. another if,

$$\mathcal{N}_{\mathcal{G}_1,p}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}_1,p}\big(\mathcal{N}_{\mathcal{G}_2,p}(\mathcal{P})\big)$$

- i.e., any node informed about the location of its neighbors with respect to $\mathcal{G}_2$ can compute its set of neighbors with respect to $\mathcal{G}_1$.

- Follow-up concept:

  When studying coordination tasks and coordination algorithms, it will be relevant to characterize the spatially distributed features of functions, vector fields, and set-valued maps with respect to suitable proximity graphs.

Notion of a Spatially Distributed Map

Given a set Y and a proximity graph $\mathcal{G}$, a map $T : X^n \to Y^n$ is *spatially distributed over* $\mathcal{G}$ if there exists a map $\tilde{T} : X \times \mathbb{F}(X) \to Y$, with the property that, for all $(p_1, \dots, p_n) \in X^n$ and for all $j \in \{1, \dots, n\}$,

$$T_j(p_1, \dots, p_n) = \tilde{T}(p_j, \mathcal{N}_{\mathcal{G},p_j}(p_1, \dots, p_n)),$$

where T_j denotes the jth component of T. In other words, the jth component of a spatially distributed map at $(p_1, \dots, p_n)$ can be computed with only knowledge of the vertex p_j and the neighboring vertices in the undirected graph $\mathcal{G}(P)$.

the jth component of a spatially distributed map at $(p_1, \dots, p_n)$ can be computed with the knowledge of the vertex p_j and the neighboring vertices in the undirected graph $\mathcal{G}(P)$

Encoding Coordination Tasks

- The aggregate behavior of the entire mobile network is evaluated by means of appropriate objective functions
- Achieving a coordination task corresponds to moving the agents and changing their state to maximize or minimize the objective function.
- Since maximizers or minimizers must be critical points, we seek to characterize the critical points of the aggregate objective function.

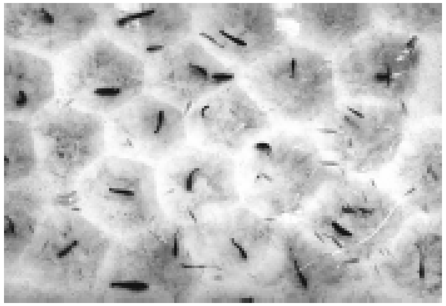
Problem: Coverage Optimization

DESIGN of performance metrics

1. how to cover a region with n minimum-radius overlapping disks?
2. how to design a minimum-distortion (fixed-rate) vector quantizer? (Lloyd '57)
3. where to place mailboxes in a city / cache servers on the internet?

ANALYSIS of cooperative distributed behaviors

4. how do animals share territory? what if every fish in a swarm goes toward center of own dominance region?



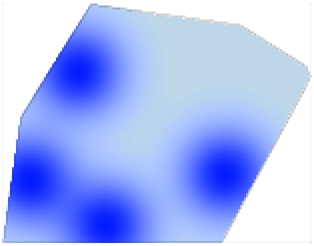
Barlow, Hexagonal territories, *Animal Behavior*, 1974

5. what if each vehicle goes to center of mass of own Voronoi cell?
6. what if each vehicle moves away from closest vehicle?

Objective Functions for Deployment

Deployment problem consists of placing a network of mobile agents inside a given environment to achieve maximum coverage.

Given a density function ϕ and a performance function f, we are interested in maximizing the expected value of the coverage performance provided by the group of agents for points in the convex polytope $Q \subset \mathbb{R}^d$. We thus define $\mathcal{H} : Q^n \to \mathbb{R}$ by



$$\mathcal{H}(P) = \int_Q \max_{i \in \{1,\ldots,n\}} f(\|q - p_i\|)\phi(q)dq \,,$$

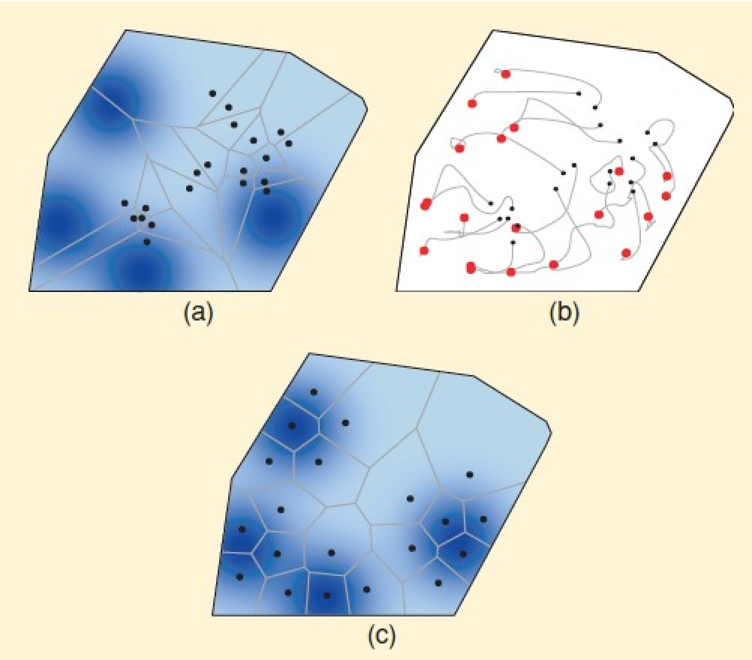
Other Formulations of Deployment: Distortion

If $f(x) = -x^2$,

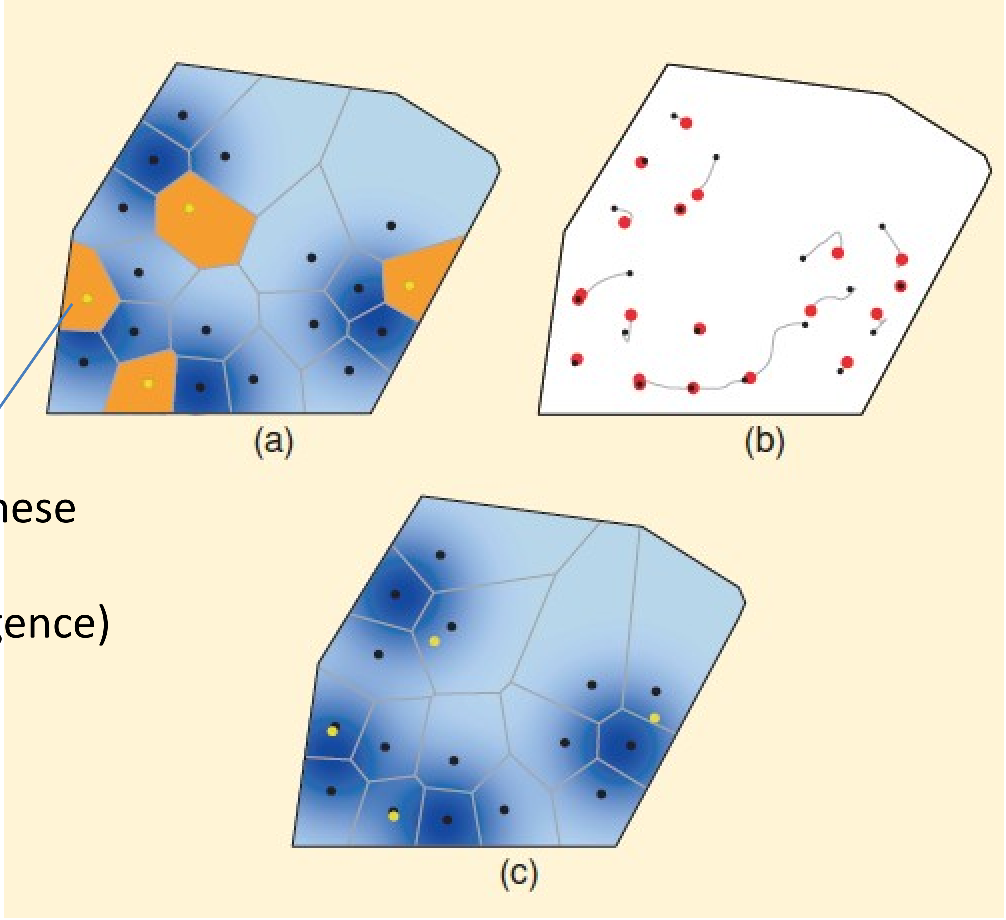
$$\mathcal{H}_{\mathrm{C}}(P) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|^2 \phi(q) dq$$

$$= -\sum_{i=1}^{n} J(V_i(P), p_i) ,$$

...like moment of inertia



Adaptation to Failure



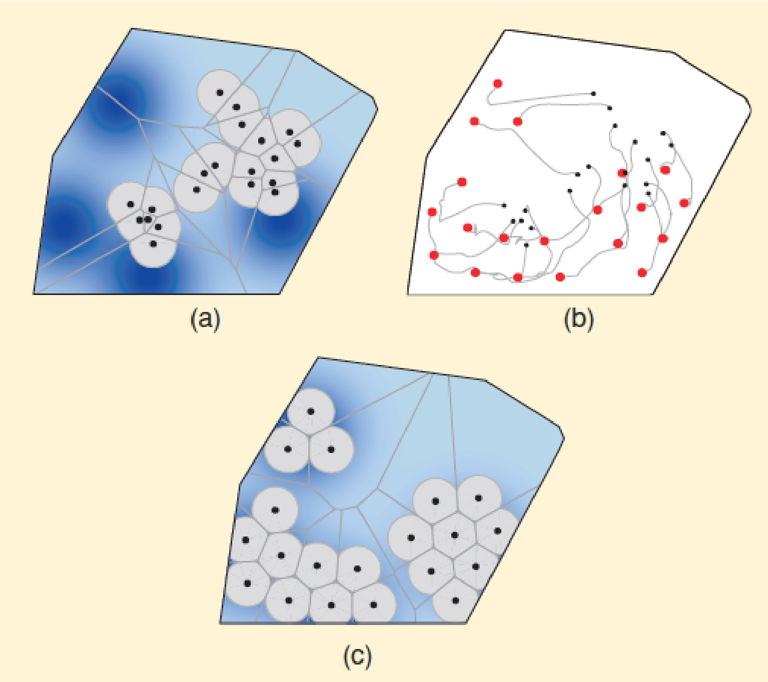
Adjustment when these units go down
(after initial convergence)

Other Formulations of Deployment: Area

If $f = 1_{[0,R]}$, where $R > 0$, then $\mathcal{H}$ corresponds to the area, weighted according to ϕ, of the union of the n balls $B(p_1, R), \ldots, B(p_n, R)$; that is,

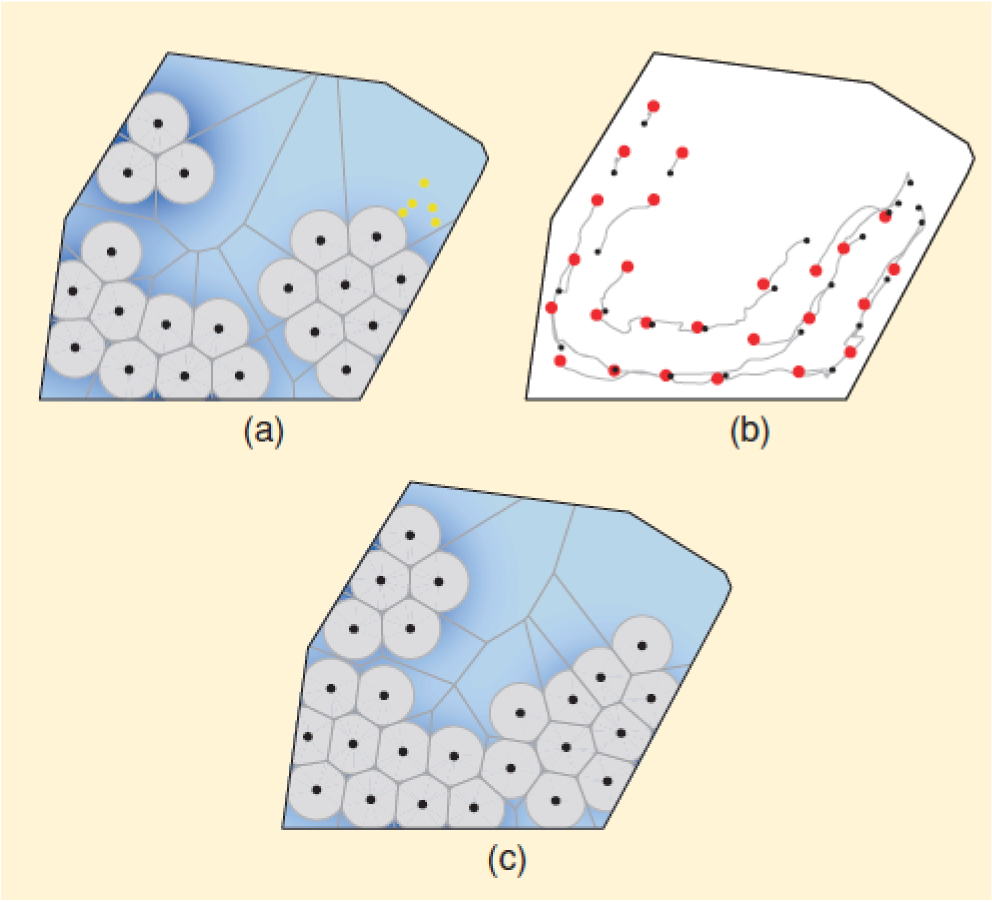
$$\mathcal{H}_{\text{area},R}(P) = \text{area}_\phi \left(\cup_{i=1}^{n} B(p_i, R)\right) ,$$

where $\text{area}_\phi(S) = \int_S \phi(q)dq$.



Adaptation to Agent Arrivals

5 new agents
(in yellow) join
the group after
initial convergence



Consensus on a Graph

For this specific case, let us simplify to a fixed graph $G = (\{1, \ldots, n\}, E)$

The **Laplacian** of this graph is defined as:

$$L_{ij} = \begin{cases} -1, & \text{if } \{i, j\} \in E, \\ \text{degree}(i), & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$$

This is a symmetric, positive definite and singular matrix with rank (n-1) only in the case of a single connected component.

Using this, disagreement can be described by a potential:

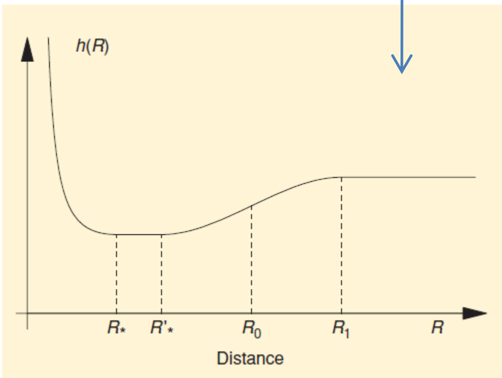
$$\Phi_G(x) = x^T L x = \frac{1}{2} \sum_{\{i,j\} \in E} (x_j - x_i)^2 .$$

Spatial Versions of Consensus

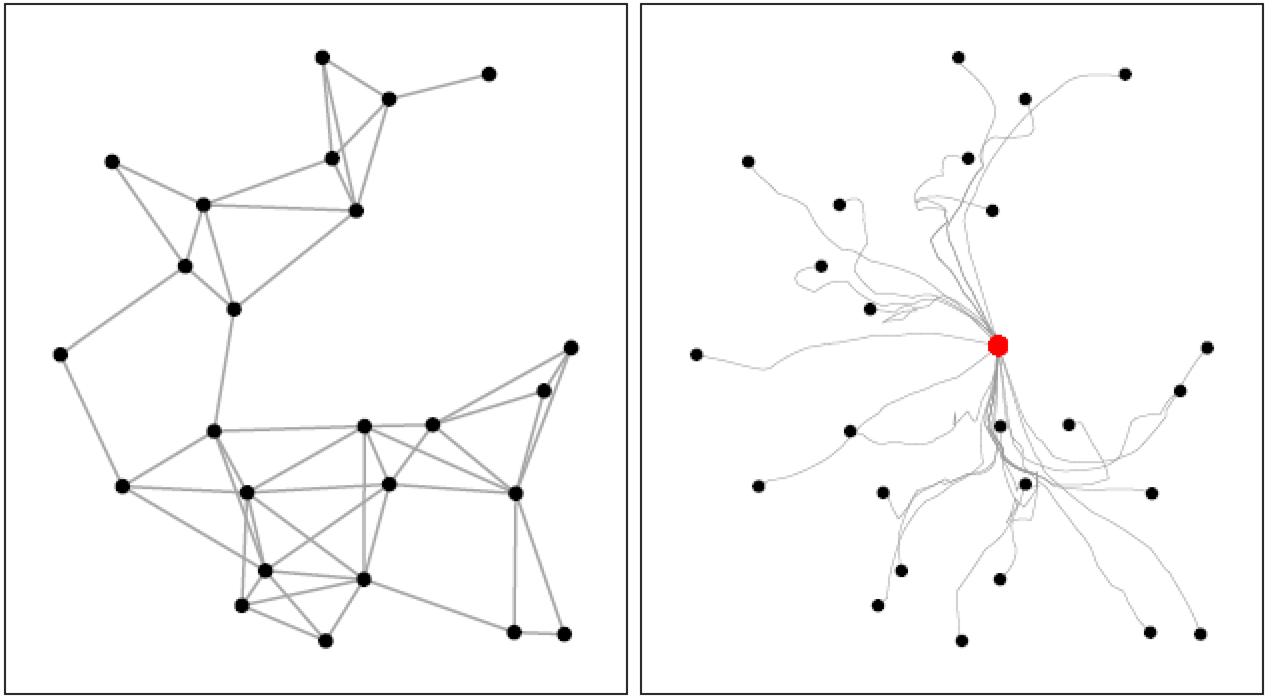
- Rendezvous means agreement over the location of the agents in a network. An objective function that is useful for the purpose of rendezvous is

$$V_{\text{diam}}(P) = \max\{\|p_i - p_j\| \mid i, j \in \{1, \dots, n\}\}.$$

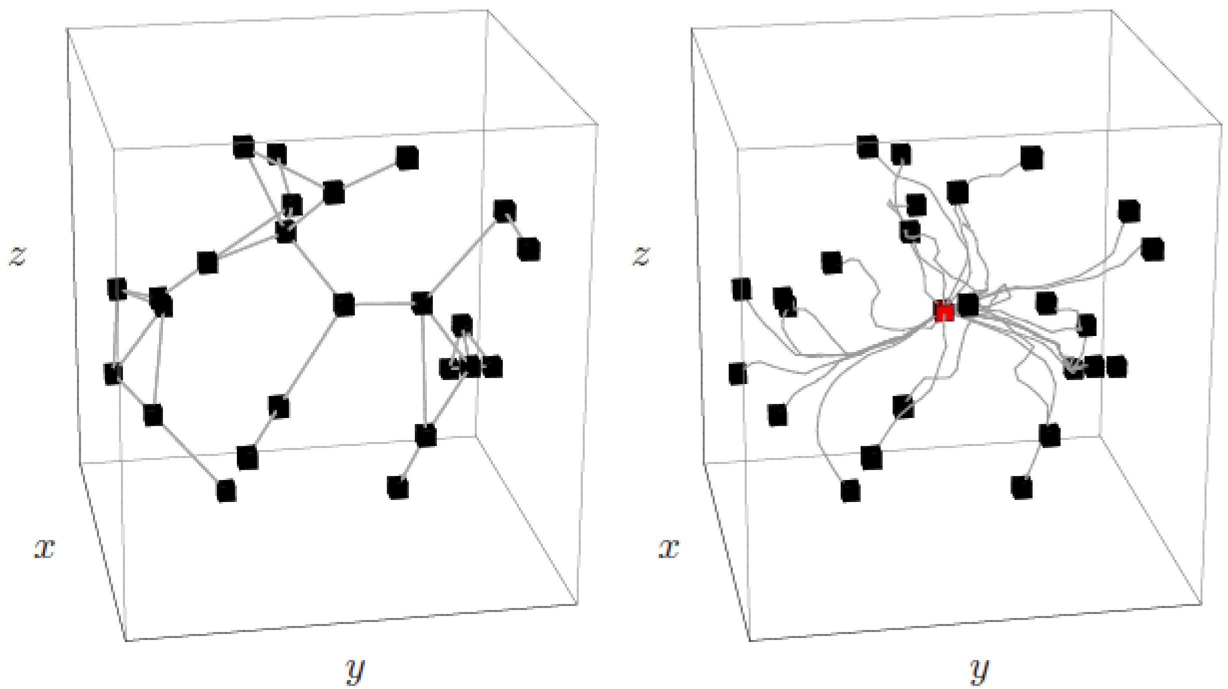
- Cohesiveness: $$\mathcal{H}_{\text{cohe},\mathcal{G}}(P) = \sum_{\{p_i, p_j\} \in \mathcal{E}_{\mathcal{G}}(P)} h(\|p_i - p_j\|)$$



Rendezvous Task

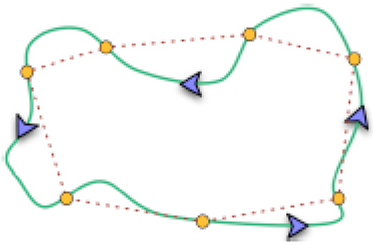


Rendezvous in 3-dim



Boundary Estimation

- Objective: detection and estimation of an evolving boundary in two dimensions by a robotic sensor network
- This type of operation can be of interest in the validation of oceanographic and atmospheric models, as well as for the demarcation of hazardous environments
- Can we have an algorithm that is distributed, in the sense that it does not require the use of a central station or "fusion center"?

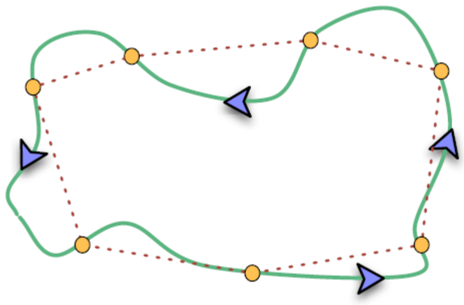


Boundary Estimation

- Assume that the unknown set Q is the planar subset where a certain environmental quantity, e.g., heat or chemical concentration, is above a given threshold.
- The objective is to estimate the boundary ∂Q by means of an array of sensors able to locally detect ∂Q and to move towards and along it.
- Consider a basic task: how to place the robots along ∂Q in such a way that the polygon, whose vertices are the robots' positions, is a good approximation of Q.
- Optimal estimation problem is equivalent to finding the "best" N-vertices polytope inscribed inside Q that best approximates Q according to some metric

Outline of Boundary Estimation Algorithm

- For a single connected component, order them on the border in a ring (counterclockwise)
- Define an error function of the form:



$$\mathcal{H}(g, g_I) := \int_a^b \|g(t) - g_I(t)\| dt$$

Area of convex set
- area of inner approx. polygon

Param. repr.
of boundary

Param. repr.
of interp. lines
between nodes

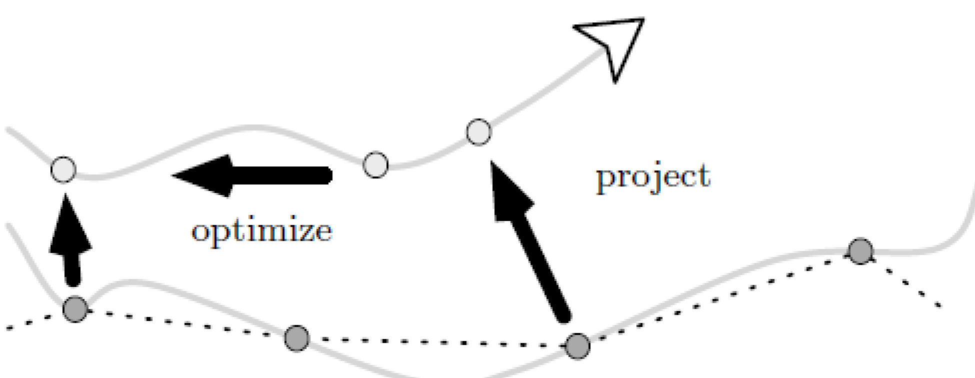
- So, motion coordination requires

$$\min_{p_1,\dots,p_N \in \partial Q} \mathcal{H}(g, g_I) = A(Q) - \max_{p_1,\dots,p_N \in \partial Q} A(\mathrm{co}(p_1, \dots, p_N))$$

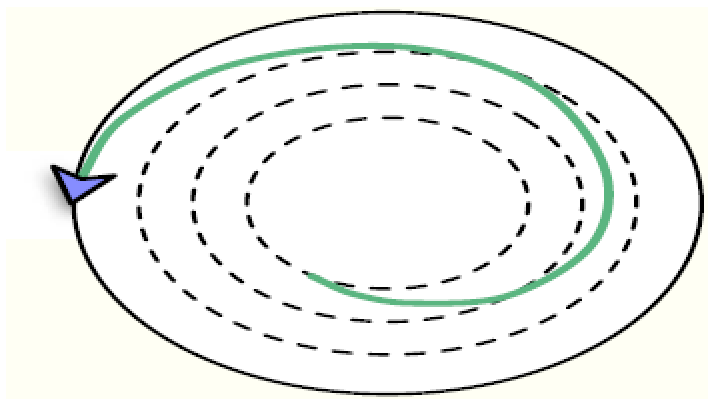
Convex hull

What a Module Does

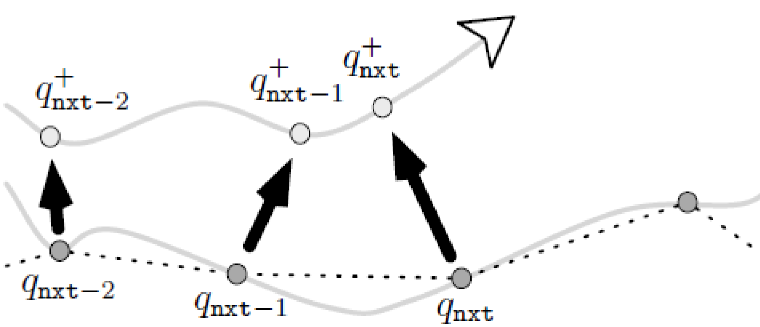
The agent moves counterclockwise along the moving boundary ∂Q, collecting estimates of its tangent and curvature. Using these estimates, the agent executes the following two actions. First, it updates the positions of the interpolation points so that they take value on the estimate of ∂Q. In other words, as sufficient information is available, each interpolation point q_α, $\alpha \in \{1, \ldots, n_{\mathrm{ip}}\}$, is projected onto the estimated boundary. Second, after an interpolation point q_α has been projected, the agent collects sufficient information so that it can locally optimize the location of q_α along the estimate of ∂Q. Here, by an estimate of the time-varying ∂Q, we mean the trajectory of the agent along the moving boundary.



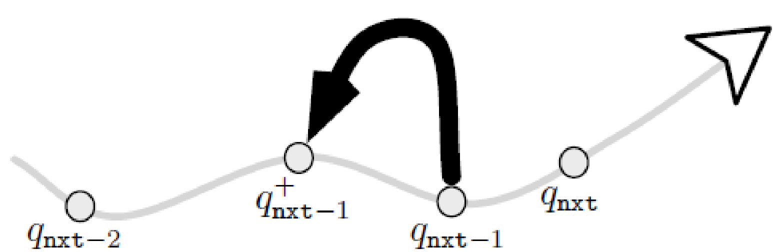
What a Module Does



The agent moves along the time-varying boundary ∂Q, here depicted as a sequence of growing ellipses, and its trajectory is an approximation to ∂Q.



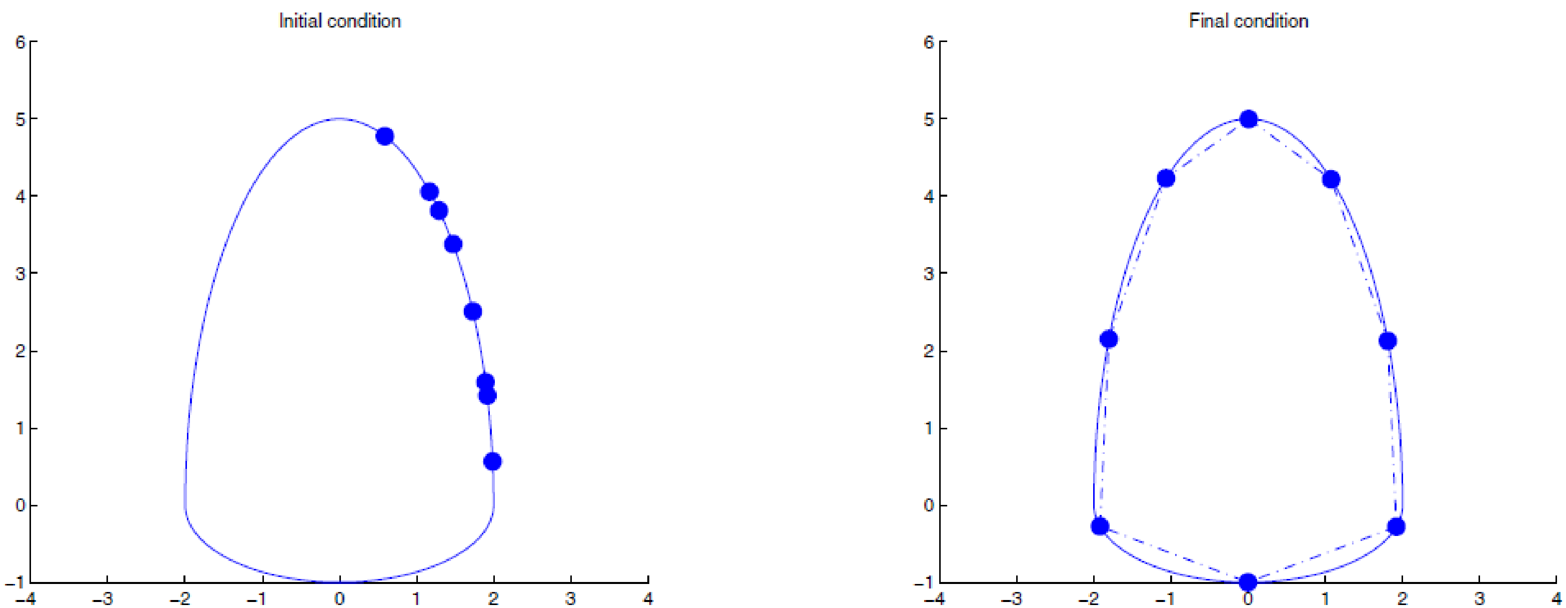
The projection of interpolation point $q_{\mathtt{nxt}}$ onto the curve $\mathtt{path}$.



Optimal placement of the interpolation point $q_{\mathtt{nxt}-1}$ along the curve $\mathtt{path}$.

Boundary Estimation using Gradient Flow

$$\dot{p}_i = \mathrm{proj}_{T\partial Q}\left(\frac{\partial A(\mathrm{co}(p_1,\ldots,p_N))}{\partial p_i}\right) = \frac{1}{2}\,\mathrm{proj}_{T\partial Q}\begin{pmatrix} y_{i+1}-y_{i-1} \\ x_{i-1}-x_{i+1}\end{pmatrix}$$



How much information does any one agent need?

Point to Ponder

What differentiates the constructions of these objective functions from simple minded potential functions? Both do gradient descent…

- The task is more complex (e.g., deployments w.r.t. visibility, consensus)
- We are only working from **local** information – in principle, no global coordinate frame to anchor things to