Distributed motion coordination of robotic networks

Lecture 2 – models and complexity notions

Jorge Cortés

Applied Mathematics and Statistics
Baskin School of Engineering
University of California at Santa Cruz
http://www.ams.ucsc.edu/~jcortes

Summer School on Geometry Mechanics and Control
Centro Internacional de Encuentros Matemáticos
Castro Urdiales, June 25-29, 2007

Roadmap

Today

Cooperative robotic network model

1. **Proximity graphs** as interaction topology
2. Control and communication **laws**, coordination **tasks**
3. **Complexity** notions
4. Analysis of **agree and pursue** coordination algorithm

Modeling theme

Broad aim: optimal trade-offs in sensing, control, communication

Given two coordination algorithms that achieve the same task, which one is better?

Specific objectives

- formalize execution of coordination algorithms
- characterize performance, costs – complexity
- rigorously combine strategies to achieve more complex tasks

Long standing tradition in the theory of distributed algorithms and parallel computing – but networks have fixed topology

Outline

1. Proximity graphs
2. Model for cooperative robotic networks
   - Control and communication laws
   - Coordination tasks
   - Complexity notions
3. Agree and pursue coordination law: complexity analysis

Proximity graphs

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

Appear in computational geometry and topology control of wireless networks

Proximity graphs

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

Appear in computational geometry and topology control of wireless networks

Definition (Proximity graph)

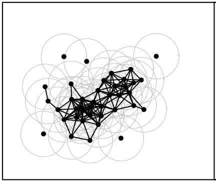
Let X be a d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, with $d_1 + d_2 = d$. Let $\mathbb{G}(X)$ be the set of all undirected graphs whose vertex set is an element of $\mathbb{F}(X)$ (finite subsets of X)

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

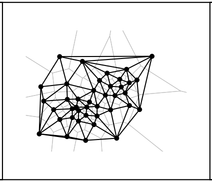
Examples of proximity graphs

On $(\mathbb{R}^d, \mathsf{dist}_2)$, $(\mathbb{S}^d, \mathsf{dist}_g)$, or $(\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}, (\mathsf{dist}_2, \mathsf{dist}_g))$

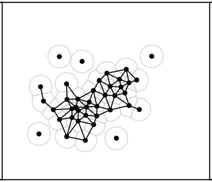
1. the **r-disk graph** $\mathcal{G}_{\text{disk}}(r)$, for $r \in \mathbb{R}_{>0}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{disk}}(r)}(\mathcal{P})$ if $\mathsf{dist}(p_i, p_j) \leq r$
2. the **Delaunay graph** $\mathcal{G}_{\text{D}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{D}}}(\mathcal{P})$ if $V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset$ ▸ Definition
3. the **r-limited Delaunay graph** $\mathcal{G}_{\text{LD}}(r)$, for $r \in \mathbb{R}_{>0}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{LD}(r)}}(\mathcal{P})$ if $V_{i,\frac{r}{2}}(\mathcal{P}) \cap V_{j,\frac{r}{2}}(\mathcal{P}) \neq \emptyset$ ▸ Definition
4. the **relative neighborhood graph** $\mathcal{G}_{\text{RN}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\text{RN}}}(\mathcal{P})$ if $p_k \notin B(p_i, \mathsf{dist}(p_i, p_j)) \cap B(p_j, \mathsf{dist}(p_i, p_j))$ for all $p_k \in \mathcal{P}$



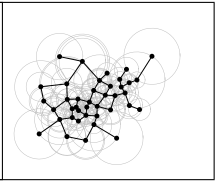
$\mathcal{G}_{\text{disk}}(r)$



$\mathcal{G}_{\text{D}}$



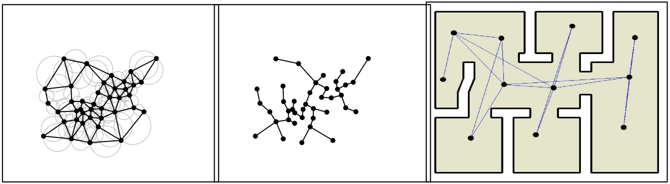
$\mathcal{G}_{\text{LD}}(r)$



$\mathcal{G}_{\text{RN}}$

More examples of proximity graphs on Euclidean space

1. the *Gabriel graph* $\mathcal{G}_{\mathrm{G}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{G}}}(\mathcal{P})$ if $p_k \notin B\big(\frac{p_i+p_j}{2}, \frac{\mathrm{dist}(p_i,p_j)}{2}\big)$ for all $p_k \in \mathcal{P}$
2. the *Euclidean minimum spanning tree* $\mathcal{G}_{\mathrm{EMST}}$, that assigns to each $\mathcal{P}$ a minimum-weight spanning tree of the complete weighted digraph $(\mathcal{P}, \{(p,q) \in \mathcal{P} \times \mathcal{P} \mid p \neq q\}, \mathcal{A})$, with weighted adjacency matrix $a_{ij} = \|p_i - p_j\|_2$, for $i, j \in \{1, \dots, n\}$
3. given a simple polygon Q in $\mathbb{R}^2$, the *visibility graph* $\mathcal{G}_{\mathrm{vis,Q}}$, with $(p_i, p_j) \in \mathcal{E}_{\mathcal{G}_{\mathrm{vis,Q}}}(\mathcal{P})$ if the closed segment $[p_i, p_j]$ from p_i to p_j is contained in Q



$\mathcal{G}_{\mathrm{G}}$ $\mathcal{G}_{\mathrm{EMST}}$ $\mathcal{G}_{\mathrm{vis,Q}}$

Set of neighbors map

To each proximity graph $\mathcal{G}$, each $p \in X$ and each $\mathcal{P} = \{p_1, \dots, p_n\} \in \mathbb{F}(X)$

associate **set of neighbors** map $\mathcal{N}_{\mathcal{G},p} \colon \mathbb{F}(X) \to \mathbb{F}(X)$ defined by

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

Typically, p is a point in $\mathcal{P}$, but this works for any $p \in X$

When does a proximity graph provide sufficient information to compute another proximity graph?

Spatially distributed graphs

E.g., if a node knows position of its neighbors in the complete graph, then it can compute its neighbors with respect to any proximity graph

Formally, given $\mathcal{G}_1$ and $\mathcal{G}_2$,

1. $\mathcal{G}_1$ is a **subgraph** of $\mathcal{G}_2$, denoted $\mathcal{G}_1 \subset \mathcal{G}_2$, if $\mathcal{G}_1(\mathcal{P})$ is a subgraph of $\mathcal{G}_2(\mathcal{P})$ for all $\mathcal{P} \in \mathbb{F}(X)$
2. $\mathcal{G}_1$ is **spatially distributed over $\mathcal{G}_2$** if, for all $p \in \mathcal{P}$,

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

that is, any node equipped with the location of its neighbors with respect to $\mathcal{G}_2$ can compute its set of neighbors with respect to $\mathcal{G}_1$

Spatially distributed graphs

E.g., if a node knows position of its neighbors in the complete graph, then it can compute its neighbors with respect to any proximity graph

Formally, given $\mathcal{G}_1$ and $\mathcal{G}_2$,

1. $\mathcal{G}_1$ is a **subgraph** of $\mathcal{G}_2$, denoted $\mathcal{G}_1 \subset \mathcal{G}_2$, if $\mathcal{G}_1(\mathcal{P})$ is a subgraph of $\mathcal{G}_2(\mathcal{P})$ for all $\mathcal{P} \in \mathbb{F}(X)$
2. $\mathcal{G}_1$ is **spatially distributed over $\mathcal{G}_2$** if, for all $p \in \mathcal{P}$,

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

   that is, any node equipped with the location of its neighbors with respect to $\mathcal{G}_2$ can compute its set of neighbors with respect to $\mathcal{G}_1$

$\mathcal{G}_1$ spatially distributed over $\mathcal{G}_2 \implies \mathcal{G}_1 \subset \mathcal{G}_2$

Converse not true: $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r) \subset \mathcal{G}_{\mathrm{disk}}$, but $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ not spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$

Illustration

Inclusion relationships among proximity graphs

Theorem

For $r \in \mathbb{R}_{>0}$, the following statements hold:

1. $\mathcal{G}_{\mathrm{EMST}} \subset \mathcal{G}_{\mathrm{RN}} \subset \mathcal{G}_{\mathrm{G}} \subset \mathcal{G}_{\mathrm{D}}$;
2. $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r) \subset \mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$
3. $\mathcal{G}_{\mathrm{RN}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, $\mathcal{G}_{\mathrm{G}} \cap \mathcal{G}_{\mathrm{disk}}(r)$, *and* $\mathcal{G}_{\mathrm{LD}}(r)$ *are spatially distributed over* $\mathcal{G}_{\mathrm{disk}}(r)$

The inclusion $\mathcal{G}_{\mathrm{LD}}(r) \subset \mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ is in general strict

Since $\mathcal{G}_{\mathrm{EMST}}$ is by definition connected, (1) implies that $\mathcal{G}_{\mathrm{RN}}$, $\mathcal{G}_{\mathrm{G}}$ and $\mathcal{G}_{\mathrm{D}}$ are connected

Connectivity properties of $\mathcal{G}_{\text{disk}}(r)$

Theorem

For $r \in \mathbb{R}_{>0}$, the following statements hold:

1. $\mathcal{G}_{\text{EMST}} \subset \mathcal{G}_{\text{disk}}(r)$ *if and only if* $\mathcal{G}_{\text{disk}}(r)$ *is connected;*
2. $\mathcal{G}_{\text{EMST}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{RN}} \cap \mathcal{G}_{\text{disk}}(r)$, $\mathcal{G}_{\text{G}} \cap \mathcal{G}_{\text{disk}}(r)$ *and* $\mathcal{G}_{\text{LD}}(r)$ *have the same connected components as* $\mathcal{G}_{\text{disk}}(r)$ *(i.e., for all point sets* $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$*), all graphs have the same number of connected components consisting of the same vertices).*

Proximity graphs over tuples

Proximity graphs are defined for **sets** of distinct points
We are interested in **tuples** – might contain coincident points

Let $i_{\mathbb{F}} \colon X^n \to \mathbb{F}(X)$ be the natural immersion of X^n into $\mathbb{F}(X)$

$$P = (p_1, \dots, p_n) \mapsto \mathcal{P} = \{p_1, \dots, p_n\}$$

Given a proximity graph $\mathcal{G}$, define

1. $\mathcal{G} = \mathcal{G} \circ i_{\mathbb{F}} \colon X^n \to \mathbb{G}(X)$
2. The set of neighbors map $\mathcal{N}_{\mathcal{G}} \colon X^n \to \mathbb{F}(X)$ is defined by

$$\mathcal{N}_{\mathcal{G},p}(p_1, \dots, p_n) = \mathcal{N}_{\mathcal{G},p}(i_{\mathbb{F}}(p_1, \dots, p_n))$$

According to this definition, coincident points in the tuple $(p_1, \dots, p_n)$ will have the same set of neighbors

Spatially distributed maps

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

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

where T_j denotes the jth-component of T

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

Outline

The physical components of a robotic network

A synchronous robotic network as a group of robots with the ability to exchange messages according to a geometric communication topology, perform local computations and control their motion

The physical components of a robotic network

A synchronous robotic network as a group of robots with the ability to exchange messages according to a geometric communication topology, perform local computations and control their motion

Mobile robot: continuous-time continuous-space dynamical system, that is, tuple (X, U, X_0, f)

1. X is d-dimensional space chosen among $\mathbb{R}^d$, $\mathbb{S}^d$, and the Cartesian products $\mathbb{R}^{d_1} \times \mathbb{S}^{d_2}$, for some $d_1 + d_2 = d$, called the *state space*;
2. U is a compact subset of $\mathbb{R}^m$ containing $\mathbf{0}_n$, called the *input space*;
3. X_0 is a subset of X, called the *set of allowable initial states*;
4. $f\colon X \times U \to \mathbb{R}^d$ is a smooth control vector field on X, that is, f determines the robot motion $x\colon \mathbb{R}_{\geq 0} \to X$ via

$$\dot{x}(t) = f(x(t), u(t)),$$

subject to the control $u\colon \mathbb{R}_{\geq 0} \to U$

Synchronous robotic network

Definition (Robotic network)

The physical components of a *uniform robotic network* $\mathcal{S}$ consist of a tuple $(I, \mathcal{R}, \mathcal{E})$, where

1. $I = \{1, \dots, n\}$; I is called the *set of unique identifiers (UIDs)*;
2. $\mathcal{R} = \{R^{[i]}\}_{i \in I} = \{(X, U, X_0, f)\}_{i \in I}$ is a set of mobile robots;
3. $\mathcal{E}$ is a map from X^n to the subsets of $I \times I$; this map is called the *communication edge map*.

The map $x \mapsto (I, \mathcal{E}(x))$ models the topology of the communication service among the robots. As communication graphs, we will adopt the proximity graph determined by network capabilities

A couple of examples

Locally-connected first-order robots in $\mathbb{R}^d$: $\mathcal{S}_{\mathrm{disk}}$
n points $x^{[1]}, \dots, x^{[n]}$ in $\mathbb{R}^d$, $d \geq 1$, obeying $\dot{x}^{[i]}(t) = u^{[i]}(t)$, with $u^{[i]} \in [-u_{\max}, u_{\max}]$. These are identical robots of the form

$$(\mathbb{R}^d, [-u_{\max}, u_{\max}]^d, \mathbb{R}^d, (\mathbf{0}, \boldsymbol{e}_1, \dots, \boldsymbol{e}_d))$$

Each robot can communicate to other robots within r, $\mathcal{G}_{\mathrm{disk}}(r)$ on $\mathbb{R}^d$

A couple of examples

Locally-connected first-order robots in $\mathbb{R}^d$: $\mathcal{S}_{\text{disk}}$
n points $x^{[1]}, \dots, x^{[n]}$ in $\mathbb{R}^d$, $d \geq 1$, obeying $\dot{x}^{[i]}(t) = u^{[i]}(t)$, with $u^{[i]} \in [-u_{\max}, u_{\max}]$. These are identical robots of the form

$$(\mathbb{R}^d, [-u_{\max}, u_{\max}]^d, \mathbb{R}^d, (\mathbf{0}, \boldsymbol{e}_1, \dots, \boldsymbol{e}_d))$$

Each robot can communicate to other robots within r, $\mathcal{G}_{\text{disk}}(r)$ on $\mathbb{R}^d$

Locally-connected first-order robots in $\mathbb{S}^1$: $\mathcal{S}_{\text{circle,disk}}$
n robots $\theta^{[1]}, \dots, \theta^{[n]}$ in $\mathbb{S}^1$, moving along on the unit circle with angular velocity equal to the control input. Each robot is described by

$$(\mathbb{S}^1, [-u_{\max}, u_{\max}], \mathbb{S}^1, (0, \boldsymbol{e}))$$

($\boldsymbol{e}$ describes unit-speed counterclockwise rotation). Each robot can communicate to other robots within r along the circle, $\mathcal{G}_{\text{disk}}(r)$ on $\mathbb{S}^1$

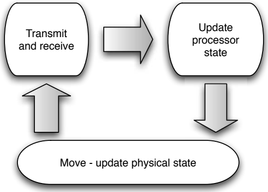
Outline

Uniform control and communication law

1. **communication schedule** — $\mathbb{T} = \{t_\ell\}_{\ell \in \mathbb{N}_0} \subset \mathbb{R}_{\geq 0}$
2. **communication alphabet** — L including the `null` message
3. **processor state space** — W, with initial allowable $W_0^{[i]}$

4. **message-generation function** — $\text{msg}\colon \mathbb{T} \times X \times W \times I \to L$
5. **state-transition functions** — $\text{stf}\colon X \times W \times L^n \to W$
6. **control function** — $\text{ctrl}\colon \mathbb{T} \times X \times W \times L^n \to U$

Execution: discrete-time communication
discrete-time computation
continuous-time motion



Evolution of a robotic network – formal definition

Evolution of $(\mathcal{S}, \mathcal{CC})$ from $x_0^{[i]} \in X_0{}^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, is the collection of curves $x^{[i]} : \mathbb{R}_{\geq 0} \to X^{[i]}$ and $w^{[i]} : \mathbb{T} \to W^{[i]}$, $i \in I$

$$\dot{x}^{[i]}(t) = f\Big(x^{[i]}(t), \operatorname{ctrl}^{[i]}\big(t, x^{[i]}(\lfloor t \rfloor), w^{[i]}(\lfloor t \rfloor), y^{[i]}(\lfloor t \rfloor)\big)\Big),$$

where $\lfloor t \rfloor = \max\{\ell \in \mathbb{T} \mid \ell < t\}$, and

$$w^{[i]}(\ell) = \operatorname{stf}^{[i]}(x^{[i]}(\ell), w^{[i]}(\ell - 1), y^{[i]}(\ell)),$$

with $x^{[i]}(0) = x_0^{[i]}$, and $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$

Here, $y^{[i]} : \mathbb{T} \to L^n$ (describing the messages received by processor i) has components $y_j^{[i]}(\ell)$, for $j \in I$, given by

$$y_j^{[i]}(\ell) = \begin{cases} \operatorname{msg}^{[j]}(x^{[j]}(\ell), w^{[j]}(\ell - 1), i), & \text{if } (i, j) \in \mathcal{E}\big(x^{[1]}(\ell), \dots, x^{[n]}(\ell)\big) \\ \texttt{null}, & \text{otherwise} \end{cases}$$

Processor state set and alphabet quantization

We allow the processor state set and the communication alphabet to contain an infinite number of symbols. This is equivalent to assuming that we neglect any inaccuracies due to quantization

It is convenient to allow messages to contain real numbers because, in many control and communication laws, the robots exchange their states, including both their processor and their physical states

For such laws, we identify the communication alphabet with $L = (X \times W) \cup \{\texttt{null}\}$; corresponding message generation function $\mathrm{msg}_{\mathrm{std}}(x, w, j) = (x, w)$ is the **standard message-generation function**

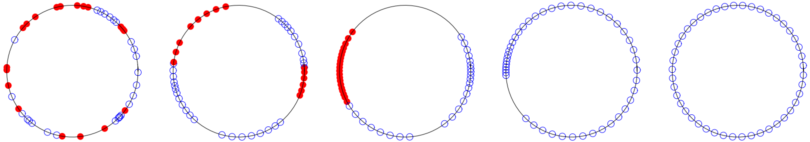
Toy[1] example: agree and pursue

Uniform network of locally-connected first-order robots on $\mathbb{S}^1$

- agents $\{1, \dots, n\}$ moving in $\mathbb{S}^1$
- each robot described by $(\mathbb{S}^1, [-u_{\max}, u_{\max}], \dot{\theta} = u)$
- interaction graph is r-disk graph on $\mathbb{S}^1$

Objectives

- Agree on a common direction of motion
- Uniformly deploy over circle



Combines **leader election** with robotic **deployment**

[1] Toy does not imply easy!

Algorithm: informal description

The **processor state** consists of

- `dir` $\in \{\texttt{c}, \texttt{cc}\}$ (the robot's direction of motion)
- `max-uid` $\in I$ (the largest UID received by the robot, initially set to the robot's UID)

Algorithm: informal description

The **processor state** consists of

- `dir` $\in \{\mathtt{c}, \mathtt{cc}\}$ (the robot's direction of motion)
- `max-uid` $\in I$ (the largest UID received by the robot, initially set to the robot's UID)

At each communication round

- each robot transmits its position and its processor state
- among the messages received from agents moving towards its position, agent picks message with largest value of `max-uid`. If this value is larger than its own, agent resets its processor state with the selected message

Algorithm: informal description

The **processor state** consists of

- `dir` $\in \{\texttt{c}, \texttt{cc}\}$ (the robot's direction of motion)
- `max-uid` $\in I$ (the largest UID received by the robot, initially set to the robot's UID)

At each communication round

- each robot transmits its position and its processor state
- among the messages received from agents moving towards its position, agent picks message with largest value of `max-uid`. If this value is larger than its own, agent resets its processor state with the selected message

Between communication rounds

- each robot moves in the clockwise or counterclockwise direction (according to $\texttt{dir} = \texttt{c}$ or $\texttt{dir} = \texttt{cc}$)
- robot moves k_{prop} times the distance to the immediately next neighbor in the chosen direction, or, if no neighbors are detected, k_{prop} times the communication range r

Algorithm: formal description

$\texttt{Alphabet:}\ L = \mathbb{S}^1 \times \{\texttt{c}, \texttt{cc}\} \times I \cup \{\texttt{null}\}$

$\texttt{Processor State:}\ w = (\texttt{dir}, \texttt{max-uid})$, where

$\texttt{dir}$	$\in \{\texttt{c}, \texttt{cc}\}$,	initially:	$\texttt{dir}^{[i]}$ unspecified
$\texttt{max-uid}$	$\in I$,	initially:	$\texttt{max-uid}^{[i]} = i$ for all i

---

$\texttt{function}$ msg(θ, w, i)

return (θ, w)

$\texttt{function}$ stf(w, y)

for each non-$\texttt{null}$ message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-uid}_{\text{rcvd}}))$ **do**
    if $(\texttt{max-uid}_{\text{rcvd}} > \texttt{max-uid})$ AND $(\mathsf{dist}_{\mathsf{cc}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{c})$ OR $(\mathsf{dist}_{\mathsf{c}}(\theta, \theta_{\text{rcvd}}) \leq r$ AND $\texttt{dir}_{\text{rcvd}} = \texttt{cc})$ **then**
        $\texttt{new-dir} := \texttt{dir}_{\text{rcvd}}$
        $\texttt{new-uid} := \texttt{max-uid}_{\text{rcvd}}$
return $(\texttt{new-dir}, \texttt{new-uid})$

$\texttt{function}$ ctrl$(\theta_{\text{smpld}}, w, y)$

$d_{\text{tmp}} := r$
for each non-$\texttt{null}$ message $(\theta_{\text{rcvd}}, (\texttt{dir}_{\text{rcvd}}, \texttt{max-uid}_{\text{rcvd}}))$ **do**
    if $(\texttt{dir} = \texttt{cc})$ AND $(\mathsf{dist}_{\mathsf{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
        $d_{\text{tmp}} := \mathsf{dist}_{\mathsf{cc}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$ and $u_{\text{tmp}} := k_{\text{prop}} d_{\text{tmp}}$ $(k_{\text{prop}} \in (0, \frac{1}{2}))$
    if $(\texttt{dir} = \texttt{c})$ AND $(\mathsf{dist}_{\mathsf{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}}) < d_{\text{tmp}})$ **then**
        $d_{\text{tmp}} := \mathsf{dist}_{\mathsf{c}}(\theta_{\text{smpld}}, \theta_{\text{rcvd}})$ and $u_{\text{tmp}} := -k_{\text{prop}} d_{\text{tmp}}$
return u_{tmp}

Outline

Coordination tasks

What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time, space, and communication complexity?

Coordination tasks

What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time, space, and communication complexity?

A **coordination task** for a robotic network $\mathcal{S}$ is a map $\mathcal{T}\colon X^n \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$

- Logic-based: agree, synchronize, form a team, elect a leader
- Motion: deploy, gather, flock, reach pattern
- Sensor-based: search, estimate, identify, track, map

Coordination tasks

What is a coordination task for a robotic network? When does a control and communication law achieve a task? And with what time, space, and communication complexity?

A **coordination task** for a robotic network $\mathcal{S}$ is a map $\mathcal{T}\colon X^n \times \mathcal{W}^n \to \{\texttt{true}, \texttt{false}\}$

- Logic-based: agree, synchronize, form a team, elect a leader
- Motion: deploy, gather, flock, reach pattern
- Sensor-based: search, estimate, identify, track, map

A control and communication law $\mathcal{CC}$ **achieves** the task $\mathcal{T}$ if, for all initial conditions $x_0^{[i]} \in X_0^{[i]}$ and $w_0^{[i]} \in W_0^{[i]}$, $i \in I$, the corresponding network evolution $t \mapsto (x(t), w(t))$ has the property that there exists $T \in \mathbb{R}_{>0}$ such that $\mathcal{T}(x(t), w(t)) = \texttt{true}$ for all $t \geq T$

Task definitions via temporal logic

Loosely speaking, achieving a task means obtaining and maintaining a specified pattern in the robot physical or processor state

In other words, the task is achieved if **at some time** and **for all subsequent times** the predicate evaluates to true along system trajectories

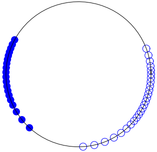
More general tasks based on more expressive predicates on trajectories can be defined through temporal and propositional logic, e.g.

periodically visiting a desired set of configurations

Direction agreement and equidistance tasks

Direction agreement task $\mathcal{T}_{\texttt{dir}} \colon (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$

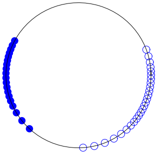
$$\mathcal{T}_{\texttt{dir}}(\theta, w) = \begin{cases} \texttt{true}, & \text{if } \texttt{dir}^{[1]} = \cdots = \texttt{dir}^{[n]} \\ \texttt{false}, & \text{otherwise} \end{cases}$$



Direction agreement and equidistance tasks

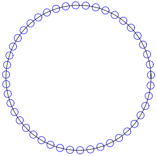
Direction agreement task $\mathcal{T}_{\texttt{dir}} : (\mathbb{S}^1)^n \times W^n \to \{\texttt{true}, \texttt{false}\}$

$$\mathcal{T}_{\texttt{dir}}(\theta, w) = \begin{cases} \texttt{true}, & \text{if } \texttt{dir}^{[1]} = \cdots = \texttt{dir}^{[n]} \\ \texttt{false}, & \text{otherwise} \end{cases}$$



For $\epsilon > 0$, **equidistance** task $\mathcal{T}_{\epsilon\text{-eqdstnc}} : (\mathbb{S}^1)^n \to \{\texttt{true}, \texttt{false}\}$ is $\texttt{true}$ iff

$$\Big| \min_{j \neq i} \mathsf{dist_c}(\theta^{[i]}, \theta^{[j]}) - \min_{j \neq i} \mathsf{dist_{cc}}(\theta^{[i]}, \theta^{[j]}) \Big| < \epsilon, \quad \text{for all } i \in I$$



Outline

Complexity notions for control and communication laws

For network $\mathcal{S}$, task $\mathcal{T}$, and algorithm $\mathcal{CC}$, define costs/complexity

control effort, communication packets, computational cost

Time complexity: maximum number of communication rounds required to achieve $\mathcal{T}$

Space complexity: maximum number of basic memory units required by a robot processor among all robots

Communication complexity: maximum number of basic messages transmitted over entire network

(among all allowable initial physical and processor states until termination)

basic memory unit/message contain $\log(n)$ bits

More formally: time complexity

The **time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$ from** $(x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]}$ is

$$\mathsf{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \inf \{\ell \mid \mathcal{T}(x(t_k), w(t_k)) = \texttt{true}\,, \text{ for all } k \geq \ell\}\,,$$

where $t \mapsto (x(t), w(t))$ is the evolution of $(\mathcal{S}, \mathcal{CC})$ from the initial condition (x_0, w_0)

The **time complexity to achieve $\mathcal{T}$ with $\mathcal{CC}$** is

$$\mathsf{TC}(\mathcal{T}, \mathcal{CC}) = \sup \left\{ \mathsf{TC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) \mid (x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]} \right\}.$$

The **time complexity of $\mathcal{T}$** is

$$\mathsf{TC}(\mathcal{T}) = \inf \{\mathsf{TC}(\mathcal{T}, \mathcal{CC}) \mid \mathcal{CC} \text{ compatible with } \mathcal{T}\}$$

More formally: communication complexity

The set of all non-null messages generated during one communication round from network state (x, w)

$$\mathcal{M}(x, w) = \left\{(i, j) \in \mathcal{E}(x) \mid \mathrm{msg}^{[i]}(x^{[i]}, w^{[i]}, j) \neq \texttt{null}\right\}.$$

The **mean communication complexity** and the **total communication complexity** to achieve $\mathcal{T}$ with $\mathcal{CC}$ from $(x_0, w_0) \in \prod_{i \in I} X_0^{[i]} \times \prod_{i \in I} W_0^{[i]}$ are,

$$\mathsf{MCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = \frac{|L|_{\text{basic}}}{\lambda} \sum_{\ell=0}^{\lambda-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

$$\mathsf{TCC}(\mathcal{T}, \mathcal{CC}, x_0, w_0) = |L|_{\text{basic}} \sum_{\ell=0}^{\lambda-1} |\mathcal{M}(x(\ell), w(\ell))|,$$

where $|L|_{\text{basic}}$ is number of basic messages required to represent elements of L and $\lambda = \mathsf{TC}(\mathcal{CC}, \mathcal{T}, x_0, w_0)$

Variations and extensions

Asymptotic results

Complexities in $O(f(n))$, $\Omega(f(n))$, or $\Theta(f(n))$ as $n \to \infty$

1. **Infinite-horizon mean communication complexity:** mean communication complexity to maintain true the task for all times

$$\mathsf{cc}(\mathcal{CC}, x_0, w_0) = \lim_{\lambda \to +\infty} \frac{|L|_{\text{basic}}}{\lambda} \sum_{\ell=0}^{\lambda} |\mathcal{M}(x(\ell), w(\ell))|$$

2. **Communication complexity in omnidirectional networks:** All neighbors of a to receive the signal it transmits. Makes sense to count the number of transmissions, i.e., a unit cost per node, rather than a unit cost per edge of the network
3. **Energy complexity**
4. **Expected** notions, rather than **worst-case** notions

Outline

Time complexity of agree-and-pursue law

Let $r\colon \mathbb{N} \to \,]0, 2\pi[$ be a monotone non-increasing function of number of agents n – modeling wireless communication congestion

Theorem

In the limit as $n \to +\infty$ and $\epsilon \to 0^+$, the network $\mathcal{S}_{\text{circle,disk}}$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\text{agrmnt}}$ and $\mathcal{T}_{\epsilon\text{-eqdstnc}}$ together satisfy:

1. $\mathsf{TC}(\mathcal{T}_{\text{agrmnt}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(r(n)^{-1})$;
2. *if $\delta(n) = nr(n) - 2\pi$ is lower bounded by a positive constant as $n \to +\infty$, then*

$$\mathsf{TC}(\mathcal{T}_{\epsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^2 \log(n\epsilon)^{-1}),$$
$$\mathsf{TC}(\mathcal{T}_{\epsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 \log(n\epsilon^{-1})).$$

If $\delta(n)$ is lower bounded by a negative constant, then $\mathcal{CC}_{\text{AGREE \& PURSUE}}$ does not achieve $\mathcal{T}_{\epsilon\text{-eqdstnc}}$ in general.

Proof sketch - O bound for $\mathcal{T}_{\mathrm{agrmnt}}$

Claim: $\mathsf{TC}(\mathcal{T}_{\mathrm{agrmnt}}, \mathcal{CC}_{\textsc{agree \& pursue}}) \leq 2\pi/(k_{\mathrm{prop}} r(n))$

By contradiction, assume there exists initial condition such that execution has time complexity $> 2\pi/(k_{\mathrm{prop}} r(n))$
Without loss of generality, $\mathtt{dir}^{[n]}(0) = \mathtt{c}$. For $\ell \leq 2\pi/(k_{\mathrm{prop}} r(n))$, let

$$k(\ell) = \mathsf{argmin}\{\mathsf{dist}_{\mathtt{cc}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \mathtt{dir}^{[i]}(\ell) = \mathtt{cc},\ i \in I\}$$

Agent $k(\ell)$ is agent moving counterclockwise that has smallest counterclockwise distance from the initial position of agent n

Proof sketch - O bound for $\mathcal{T}_{\text{agrmnt}}$

Claim: $\mathsf{TC}(\mathcal{T}_{\text{agrmnt}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \leq 2\pi/(k_{\text{prop}} r(n))$

By contradiction, assume there exists initial condition such that execution has time complexity $> 2\pi/(k_{\text{prop}} r(n))$
Without loss of generality, $\mathtt{dir}^{[n]}(0) = \mathtt{c}$. For $\ell \leq 2\pi/(k_{\text{prop}} r(n))$, let

$$k(\ell) = \mathsf{argmin}\{\mathsf{dist}_{\mathsf{cc}}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \mathtt{dir}^{[i]}(\ell) = \mathtt{cc},\ i \in I\}$$

Agent $k(\ell)$ is agent moving counterclockwise that has smallest counterclockwise distance from the initial position of agent n

Recall that according to $\mathcal{CC}_{\text{AGREE \& PURSUE}}$

- messages with $\mathtt{dir} = \mathtt{cc}$ can only travel counterclockwise
- messages with $\mathtt{dir} = \mathtt{c}$ can only travel clockwise

Therefore, position of agent $k(\ell)$ at time ℓ can only belong to the counterclockwise interval from the position of agent $k(0)$ at time 0 to the position of agent n at time 0

Proof sketch - O bound for $\mathcal{T}_{\text{agrmnt}}$

How fast the message from agent n travels clockwise?

For $\ell \leq 2\pi/(k_{\text{prop}} r(n))$, define

$$j(\ell) = \mathsf{argmax}\{\mathsf{dist_c}(\theta^{[n]}(0), \theta^{[i]}(\ell)) \mid \texttt{prior}^{[i]}(\ell) = n,\ i \in I\}$$

Agent $j(\ell)$

- has `prior` equal to n
- is moving clockwise

and is the agent furthest from the initial position of agent n in the clockwise direction with these two properties

Initially, $j(0) = n$. Additionally, for $\ell \leq 2\pi/(k_{\text{prop}} r(n))$, we claim

$$\mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \geq k_{\text{prop}} r(n)$$

This happens because either (1) there is no agent clockwise-ahead of $\theta^{[j(\ell)]}(\ell)$ within clockwise distance r and, therefore, the claim is obvious, or (2) there are such agents. In case (2), let m denote the agent whose clockwise distance to agent $j(\ell)$ is maximal within the set of agents with clockwise distance r from $\theta^{[j(\ell)]}(\ell)$. Then,

$$\begin{aligned}
&\mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[j(\ell+1)]}(\ell+1)) \\
&\quad = \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad = \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + \mathsf{dist_c}(\theta^{[m]}(\ell), \theta^{[m]}(\ell+1)) \\
&\quad \geq \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) + k_{\text{prop}}\big(r - \mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell))\big) \\
&\quad = k_{\text{prop}} r + (1 - k_{\text{prop}})\,\mathsf{dist_c}(\theta^{[j(\ell)]}(\ell), \theta^{[m]}(\ell)) \;\geq k_{\text{prop}} r
\end{aligned}$$

Therefore, after $2\pi/(k_{\text{prop}} r(n))$ communication rounds, the message with $\texttt{prior} = n$ has traveled the whole circle in the clockwise direction, and must therefore have reached agent $k(\ell)$ **Contradiction**

Proof sketch - O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Assume $\mathcal{T}_{\text{agrmnt}}$ has been achieved and all agents are moving clockwise

At time $\ell \in \mathbb{N}_0$, let $H(\ell)$ be the union of all the empty "circular segments" of length at least r,

$$H(\ell) = \{x \in \mathbb{S}^1 \mid \min_{i \in I} \mathsf{dist}_{\mathsf{c}}(x, \theta^{[i]}(\ell)) + \min_{j \in I} \mathsf{dist}_{\mathsf{cc}}(x, \theta^{[j]}(\ell)) > r\}.$$

$H(\ell)$ does not contain any point between two agents separated by a distance less than r, and each connected component has length at least r

Proof sketch - O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Assume $\mathcal{T}_{\text{agrmnt}}$ has been achieved and all agents are moving clockwise
At time $\ell \in \mathbb{N}_0$, let $H(\ell)$ be the union of all the empty "circular segments" of length at least r,

$$H(\ell) = \{x \in \mathbb{S}^1 \mid \min_{i \in I} \mathrm{dist_c}(x, \theta^{[i]}(\ell)) + \min_{j \in I} \mathrm{dist_{cc}}(x, \theta^{[j]}(\ell)) > r\}.$$

$H(\ell)$ does not contain any point between two agents separated by a distance less than r, and each connected component has length at least r

Let $n_H(\ell)$ be number of connected components of $H(\ell)$,

- if $H(\ell)$ is empty, then $n_H(\ell) = 0$
- $n_H(\ell) \leq n$
- if $n_H(\ell) > 0$, then $t \mapsto n_H(\ell + t)$ is non-increasing

Proof sketch- O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Number of connected components is strictly decreasing

Claim: if $n_H(\ell) > 0$, then $\exists t > \ell$ such that $n_H(t) < n_H(\ell)$

By contradiction, assume $n_H(\ell) = n_H(t)$ for all $t > \ell$. Without loss of generality, let $\{1, \dots, m\}$ be a set of agents with the properties

- $\mathrm{dist_{cc}}\left(\theta^{[i]}(\ell), \theta^{[i+1]}(\ell)\right) \leq r$, for $i \in \{1, \dots, m\}$
- $\theta^{[1]}(\ell)$ and $\theta^{[m]}(\ell)$ belong to the boundary of $H(\ell)$
- there is no other set with the same properties and more agents

Proof sketch- O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Number of connected components is strictly decreasing

Claim: if $n_H(\ell) > 0$, then $\exists t > \ell$ such that $n_H(t) < n_H(\ell)$

By contradiction, assume $n_H(\ell) = n_H(t)$ for all $t > \ell$. Without loss of generality, let $\{1, \dots, m\}$ be a set of agents with the properties

- $\mathsf{dist}_{\mathsf{cc}}\left(\theta^{[i]}(\ell), \theta^{[i+1]}(\ell)\right) \leq r$, for $i \in \{1, \dots, m\}$
- $\theta^{[1]}(\ell)$ and $\theta^{[m]}(\ell)$ belong to the boundary of $H(\ell)$
- there is no other set with the same properties and more agents

One can show that, for $\tau \geq \ell$ and $i \in \{2, \dots, m\}$

$$\theta^{[1]}(\tau + 1) = \theta^{[1]}(\tau) - k_{\text{prop}} r$$

$$\theta^{[i]}(\tau + 1) = \theta^{[i]}(\tau) - k_{\text{prop}} \, \mathsf{dist}_{\mathsf{c}}(\theta^{[i]}(\tau), \theta^{[i-1]}(\tau))$$

Tridiagonal and circulant linear dynamical systems

$$\mathsf{Trid}_n(a,b,c) = \begin{bmatrix} b & c & 0 & \dots & 0 \\ a & b & c & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a & b & c \\ 0 & \dots & 0 & a & b \end{bmatrix}, \quad \mathsf{Circ}_n(a,b,c) = \begin{bmatrix} b & c & 0 & \dots & a \\ a & b & c & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a & b & c \\ c & \dots & 0 & a & b \end{bmatrix}$$

Linear dynamical systems

$$y(\ell+1) = Ay(\ell), \quad \ell \in \mathbb{N}_0$$

Rates of convergence to set of equilibria can be characterized – carefully look at eigenvalues. Statements of the form

> *if* $a \geq 0$, $c \geq 0$, $b > 0$, *and* $a + b + c = 1$, *then* $\lim_{\ell \to +\infty} y(\ell) = y_{\text{ave}} \mathbf{1}$, *where* $y_{\text{ave}} = \frac{1}{n}\mathbf{1}^T y_0$, *and maximum time required (over all initial conditions* $y_0 \in \mathbb{R}^n$*) for* $\|y(\ell) - y_{\text{ave}}\mathbf{1}\|_2 \leq \epsilon \|y_0 - y_{\text{ave}}\mathbf{1}\|_2$ *is* $\Theta(n^2 \log \epsilon^{-1})$

For $d(\tau) = \left(\mathsf{dist}_{\mathsf{cc}}(\theta^{[1]}(\tau), \theta^{[2]}(\tau)), \dots, \mathsf{dist}_{\mathsf{cc}}(\theta^{[m-1]}(\tau), \theta^{[m]}(\tau))\right)$,

$$d(\tau+1) = \mathsf{Trid}_{m-1}(k_{\text{prop}}, 1-k_{\text{prop}}, 0)\, d(\tau) + r[k_{\text{prop}},\ 0,\ \cdots,\ 0]^T$$

Unique equilibrium point is $r(1,\dots,1)$. For $\eta_1 \in]0,1[$, $\tau \mapsto d(\tau)$ reaches ball of radius η_1 centered at equilibrium in $O(m\log m + \log \eta_1^{-1})$

This implies that $\tau \mapsto \sum_{i=1}^{m} d_i(\tau)$ is larger than $(m-1)(r-\eta_1)$ in time $O(m\log m + \log\eta_1^{-1}) = O(n\log n + \log\eta_1^{-1})$. After this time,

$$\begin{aligned} 2\pi &\geq n_H(\ell) r + \sum_{j=1}^{n_H(\ell)} (r-\eta_1)(m_j - 1) \\ &= n_H(\ell) r + (n - n_H(\ell))(r-\eta_1) = n_H(\ell)\eta_1 + n(r-\eta_1) \end{aligned}$$

Proof sketch- O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Take $\eta_1 = (nr - 2\pi)n^{-1} = \delta(n)n^{-1}$, and the contradiction follows from

$$
\begin{aligned}
2\pi &\geq n_H(\ell)\eta_1 + nr - n\eta_1 \\
&= n_H(\ell)\eta_1 + nr + 2\pi - nr = n_H(\ell)\eta_1 + 2\pi
\end{aligned}
$$

Therefore $n_H(\ell)$ decreases by one in time $O(n \log n)$

Proof sketch- O bound for $\mathcal{T}_{\epsilon\text{-eqdstnc}}$

Take $\eta_1 = (nr - 2\pi)n^{-1} = \delta(n)n^{-1}$, and the contradiction follows from

$$\begin{aligned} 2\pi &\geq n_H(\ell)\eta_1 + nr - n\eta_1 \\ &= n_H(\ell)\eta_1 + nr + 2\pi - nr = n_H(\ell)\eta_1 + 2\pi \end{aligned}$$

Therefore $n_H(\ell)$ decreases by one in time $O(n \log n)$

Iterating this argument n times, in time $O(n^2 \log n)$ the set H becomes empty. At that time, resulting network obeys

$$d(\tau+1) = \mathsf{Circ}_n(k_{\text{prop}}, 1 - k_{\text{prop}}, 0)\, d(\tau)$$

In time $O\big(n^2 \log \epsilon^{-1}\big)$, the error 2-norm satisfies the contraction inequality $\|d(\tau) - d_*\big\|_2 \leq \epsilon \|d(0) - d_*\|_2$, for $d_* = \frac{2\pi}{n}\mathbf{1}$

The conversion of this inequality into an appropriate inequality on ∞-norms yields the result

Communication complexity of agree-and-pursue law

Theorem

Total communication complexity of agree-and-pursue law In the limit as $n \to +\infty$ and $\epsilon \to 0^+$, the network $\mathcal{S}_{\text{circle,disk}}$, the law $\mathcal{CC}_{\text{AGREE \& PURSUE}}$, and the tasks $\mathcal{T}_{\text{agrmnt}}$ and $\mathcal{T}_{\epsilon\text{-eqdstnc}}$ together satisfy:

1. *if $\delta(n) \geq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then*

$$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\text{agrmnt}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Theta(n^2 r(n)^{-1}),$$

otherwise if $\delta(n) \leq \pi(1/k_{\text{prop}} - 2)$ as $n \to +\infty$, then

$$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\text{agrmnt}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 + nr(n)^{-1}),$$
$$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\text{agrmnt}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^2 r(n)^{-1});$$

2. *if $\delta(n)$ is lower bounded by a positive constant as $n \to +\infty$, then*

$$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\epsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in \Omega(n^3 \delta(n) \log(n\epsilon)^{-1}),$$
$$\mathsf{TCC}_{\mathsf{unidir}}(\mathcal{T}_{\epsilon\text{-eqdstnc}}, \mathcal{CC}_{\text{AGREE \& PURSUE}}) \in O(n^4 \log(n\epsilon^{-1})).$$

Comparison with leader election

Leader election task (electing a unique agent among all agents)

- different from, but closely related to, $\mathcal{T}_{\text{agrmnt}}$
- **LCR algorithm** (seen in lecture 1!) operates on a static ring network, and achieves leader election with time and total communication complexity, respectively, $\Theta(n)$ and $\Theta(n^2)$
- **Agree-and-pursue** law operates on robotic network with $r(n)$-disk communication topology, and achieves $\mathcal{T}_{\text{agrmnt}}$ with time and total communication complexity, respectively, $\Theta(r(n)^{-1})$ and $O(n^2 r(n)^{-1})$

Comparison with leader election

Leader election task (electing a unique agent among all agents)

- different from, but closely related to, $\mathcal{T}_{\text{agrmnt}}$
- **LCR algorithm** (seen in lecture 1!) operates on a static ring network, and achieves leader election with time and total communication complexity, respectively, $\Theta(n)$ and $\Theta(n^2)$
- **Agree-and-pursue** law operates on robotic network with $r(n)$-disk communication topology, and achieves $\mathcal{T}_{\text{agrmnt}}$ with time and total communication complexity, respectively, $\Theta(r(n)^{-1})$ and $O(n^2 r(n)^{-1})$

If wireless communication congestion is modeled by $r(n)$ of order $1/n$, then identical time complexity and the LCR algorithm has better communication complexity

Computations on a possibly disconnected, dynamic network are more complex than on a static ring topology

Summary and conclusions

Cooperative robotic network model

- proximity graphs
- control and communication law, task, execution
- time, space, and communication complexity
- agree and pursue

Complexity analysis is **challenging** even in 1 dimension!

Plenty of **open problems**

- What is best algorithm to achieve a task?
- What tools are useful to characterize complexity?
- How does combination of algorithms affect individual complexities?

References

Proximity graphs:

- J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502--1517, 1992
- J. Cortés, S. Martínez, and F. Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM. Control, Optimisation & Calculus of Variations*, 11:691--719, 2005

Robotic network model:

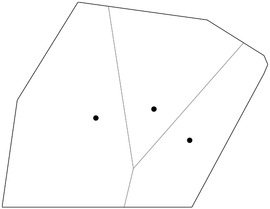
- S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli. On synchronous robotic networks -- part i: Models, tasks, and complexity. *IEEE Transactions on Automatic Control*, 53(1), 2008. To appear

Voronoi partitions

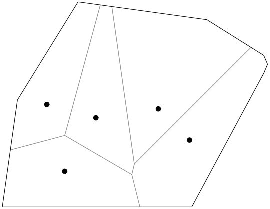
Let $(p_1, \ldots, p_n) \in Q^n$ denote the positions of n points

The **Voronoi partition** $\mathcal{V}(P) = \{V_1, \ldots, V_n\}$ generated by $(p_1, \ldots, p_n)$

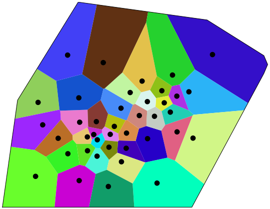
$$\begin{aligned} V_i &= \{q \in Q | \; \|q - p_i\| \le \|q - p_j\| \,,\; \forall j \neq i\} \\ &= Q \cap_j \mathcal{HP}(p_i, p_j) \qquad \text{where } \mathcal{HP}(p_i, p_j) \text{ is half plane } (p_i, p_j) \end{aligned}$$



3 generators



5 generators



50 generators

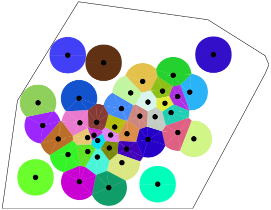
Return

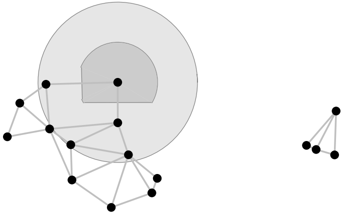
Range-limited Voronoi graph computation

Let $(p_1, \ldots, p_n) \in Q^n$ denote the positions of n points

The **r-limited Voronoi partition** $\mathcal{V}_r(P) = \{V_{1,r}, \ldots, V_{n,r}\}$ generated by $(p_1, \ldots, p_n)$

$$V_{i,r}(\mathcal{P}) = V_i(\mathcal{P}) \cap \overline{B}(p_i, r), \quad i \in \{1, \ldots, n\}$$

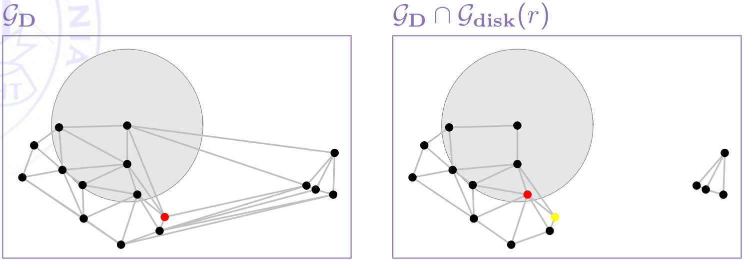




$\mathcal{G}_{\text{LD}}(r)$ is **spatially distributed** over $\mathcal{G}_{\text{disk}}(r)$

Return

$\mathcal{G}_{\mathrm{D}}$ and $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ computation



$\mathcal{G}_{\mathrm{D}}$ and $\mathcal{G}_{\mathrm{D}} \cap \mathcal{G}_{\mathrm{disk}}(r)$ are **not** spatially distributed over $\mathcal{G}_{\mathrm{disk}}(r)$

Return