Control and Coordination of Multi-Agent Systems

Magnus Egerstedt
Institute for Robotics and Intelligent Machines, Georgia Tech
http://www.robotics.gatech.edu
A (Swiss) Mood Picture

Courtesy of Alcherio Martinoli
Why Multi-Robot Systems?

- Strength in numbers
- Lots of (potential) applications
- Confluence of technology and algorithms
- Scientifically interesting!
But How?

- Local (distributed)
- Scalable (decentralized)
- Safe and Reactive
- Emergent (but not too much)

Application Domains

- Multi-agent robotics
- Sensor and communications networks
- Biological networks
- Coordinated control
Application Domains

Multi-agent robotics

“There is nothing more practical than a good theory” - James C. Maxwell (Lewin? Pauling?)

“In theory, theory and practice are the same. In practice, they are not” – Yogi Berra
1. GRAPH-BASED ABSTRACTIONS

2. FORMATION CONTROL

3. INTERACTING WITH NETWORKS
A True Swarm

“They look like ants.”
– Stephen Pratt, Arizona State University

Magnus Egerstedt, 2017
Graphs as Network Abstractions

- A networked sensing and actuation system consists of
  - **NODES** - physical entities with limited resources (computation, communication, perception, control)
  - **EDGES** - virtual entities that encode the flow of information between the nodes

- The “right” mathematical object for characterizing such systems at the network-level is a **GRAPH**
  - Purely *combinatorial* object (no *geometry* or *dynamics*)
  - The characteristics of the information flow is abstracted away through the (possibly weighted and directed) edges
Graphs as Network Abstractions

- The connection between the combinatorial graphs and the geometry of the system can for instance be made through geometrically defined edges.
- Examples of such proximity graphs include disk-graphs, Delaunay graphs, visibility graphs, and Gabriel graphs[1].

\[ \mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\} \]
\[ \mathcal{E} = \{(n_1, n_2), (n_2, n_3), (n_3, n_4), (n_2, n_4), (n_4, n_5), (n_4, n_6), (n_5, n_6)\} \]
**The Basic Setup**

- $x_i(k) =$ “state” at node $i$ at time $k$

- $N_i(k) =$ “neighbors” to agent $i$

- Information “available to agent $i$

  \[ I^c_i(k) = \{ x_j(k) \mid j \in N_i(k) \} \]  

  or

  \[ I^r_i(k) = \{ x_i(k) - x_j(k) \mid j \in N_i(k) \} \]

  common ref. frame (comms.)

  relative info. (sensing)

- Update rule:

  \[ x_i(k+1) = F_i(x_i(k), I_i(k)) \]  

  or

  \[ \dot{x}_i(t) = F_i(x_i(t), I_i(t)) \]  

  discrete time

  continuous time

- **How pick the update rule?**
Rendezvous – A Canonical Problem

- Given a collection of mobile agents who can only measure the relative displacement of their neighbors (no global coordinates)

- Problem: Have all the agents meet at the same (unspecified) position

- If there are only two agents, it makes sense to have them drive towards each other, i.e.

\[
\begin{align*}
\dot{x}_1 &= -\gamma_1(x_1 - x_2) \\
\dot{x}_2 &= -\gamma_2(x_2 - x_1)
\end{align*}
\]

- If \( \gamma_1 = \gamma_2 \) they should meet halfway
Rendezvous – A Canonical Problem

- If there are more than two agents, they should probably aim towards the centroid of their neighbors (or something similar)

\[ \dot{x}_i = - \sum_{j \in \mathcal{N}_i} (x_i - x_j) \]

The “consensus protocol” drives all states to the same value if the interaction topology is “rich enough”

Fact [2-4]: If and only if the graph* is connected, the consensus equation drives all agents to the same state value

$$\lim_{t \to \infty} x_i(t) = \bar{x} = \frac{1}{N} \sum_{j=1}^{N} x_j(0)$$

*static and undirected graphs
Consensus/Rendezvous

Pickem, Squires, Egerstedt, 2015
Algebraic Graph Theory

- To show this, we need some tools…
- Algebraic graph theory provides a bridge between the combinatorial graph objects and their matrix representations
  - **Degree matrix:**
    \[ D = \text{diag}(\text{deg}(n_1), \ldots, \text{deg}(n_N)) \]
  - **Adjacency matrix:**
    \[ A = [a_{ij}], \quad a_{ij} = \begin{cases} 1 & \text{if } \quad \begin{array}{c} n_i \\ \rightarrow \end{array} n_j \\ 0 & \text{o.w.} \end{cases} \]
  - **Incidence matrix** (directed graphs):
    \[ \mathcal{I} = [\nu_{ij}], \quad \nu_{ij} = \begin{cases} 1 & \text{if } \quad \begin{array}{c} e_j \\ \rightarrow \end{array} n_i \\ -1 & \text{if } \quad \begin{array}{c} n_i \\ \rightarrow \end{array} e_j \\ 0 & \text{o.w.} \end{cases} \]
  - **Graph Laplacian:**
    \[ \mathcal{L} = D - A = \mathcal{I} \mathcal{I}^T \]
One reason why the graph Laplacian is so important is through the already seen “consensus equation”

\[ \dot{x}_i = - \sum_{j \in \mathcal{N}_i} (x_i - x_j), \; i = 1, \ldots, N \]

or equivalently (W.L.O.G. scalar agents)

\[ \begin{align*}
\dot{x}_i &= -\text{deg}(n_i)x_i + \sum_{j=1}^{N} a_{ij}x_j \\
x &= \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T
\end{align*} \Rightarrow \dot{x} = -\mathcal{L}x \]

This is an autonomous LTI system whose stability properties depend purely on the spectral properties of the Laplacian.
Graph Laplacians: Useful Properties

- It is orientation independent
- It is symmetric and positive semi-definite
- If the graph is connected then

\[
\text{eig}(\mathcal{L}) = \{\lambda_1, \ldots, \lambda_N\}, \quad \text{with } 0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N
\]

\[
\text{eigv}(\mathcal{L}) = \{\nu_1, \ldots, \nu_N\}, \quad \text{with } \text{null}(\mathcal{L}) = \text{span}\{\nu_1\} = \text{span}\{1\}
\]

\[
\begin{bmatrix}
1 \\
1 \\
\vdots \\
1
\end{bmatrix}
\]
Stability: Basics

- The stability properties (what happens as time goes to infinity?) of a linear, time-invariant system is completely determined by the eigenvalues of the system matrix

\[ \dot{x} = Ax \quad (\dot{x} = -Lx) \]

- Eigenvalues \( \lambda(A) = \lambda_1, \ldots, \lambda_n \)

- Asymptotic stability: \( \text{Re}(\lambda_i) < 0, \ i = 1, \ldots, n \Rightarrow \lim_{t \to \infty} x(t) = 0 \)
Stability: Basics

- Unstable: \( \exists i \text{ s.t. } \mathrm{Re}(\lambda_i) > 0 \Rightarrow \exists x(0) \text{ s.t. } \lim_{t \to \infty} \|x(t)\| = \infty \)

- (A special case of) Critically stable:
  \[ 0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n, \Rightarrow \lim_{t \to \infty} x(t) \in \text{null}(A) \]

  This is the case for the consensus equation
Static and Undirected Consensus

- If the graph is static and connected, under the consensus equation, the states will reach $null(L)$.

- Fact (again):

  $$null(L) = \text{span}\{1\}, \ x \in null(L) \iff x = \begin{bmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{bmatrix}, \ \alpha \in \mathbb{R}$$

- So all the agents state values will end up at the same value, i.e. the consensus/rendezvous problem is solved!

\[
\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j) \Rightarrow \lim_{t \to \infty} x_i(t) = \frac{1}{n} \sum_{j=1}^{n} x_j(0) = \frac{1}{n} \mathbf{1}^T x(0)
\]
Convergence Rates

• The second smallest eigenvalue of the graph Laplacian is really important!
• Algebraic Connectivity (= 0 if and only if graph is disconnected)
• Fiedler Value (robustness measure)
• **Convergence Rate:**
  \[ \|x(t) - \frac{1}{n} \mathbf{1}\mathbf{1}^T x(0)\| \leq Ce^{-\lambda_2 t} \]

• **Punch-line:** The more connected the network is, the faster it converges (and the more information needs to be shuffled through the network)

  - Complete graph: \( \lambda_2 = n \)
  - Star graph: \( \lambda_2 = 1 \)
  - Path graph: \( \lambda_2 < 1 \)
Cheeger’s Inequality

\[ \phi(S) = \frac{\epsilon(S)}{\min\{|S|, |S^c|\}} \]

(measures how many edges need to be cut to make the two subsets disconnected as compared to the number of nodes that are lost)

**isoperimetric number:**

\[ \phi(G) = \min_S \phi(S) \]

(measures the robustness of the graph)

\[ \phi(G) \geq \lambda_2 \geq \frac{\phi(G)^2}{2\Delta(G')} \]
Summary I

- Graphs are natural abstractions (combinatorics instead of geometry)
- Consensus problem (and equation)
- Static Graphs:
  - Undirected: Average consensus iff G is connected
- Need richer network models and more interesting tasks!
1. GRAPH-BASED ABSTRACTIONS
2. FORMATION CONTROL
3. INTERACTING WITH NETWORKS
Formation Control v.1

- Being able to reach consensus goes beyond solving the rendezvous problem.
- Formation control:

\[ x_1, \ldots, x_N \quad \text{agent positions} \quad \rightarrow \quad y_1, \ldots, y_N \quad \text{target positions} \]

- But, formation achieved if the agents are in any translated version of the targets, i.e.,

\[ x_i = y_i + \tau, \quad \forall i, \quad \text{for some } \tau \]

- Enter the consensus equation [5]:

\[
\begin{align*}
e_i &= x_i - y_i \\
\dot{e}_i &= -\sum_{j \in N_i} (e_i - e_j) \\
e_i(\infty) &= e_j(\infty) = \tau \\
\end{align*}
\]

\[
\begin{align*}
\dot{x}_i &= \sum_{j \in N_i} [(x_i - x_j) - (y_i - y_j)] \\
x_i(\infty) &= y_i + \tau, \quad \forall i
\end{align*}
\]
Formation Control v.1
Beyond Static and Undirected Consensus

- So far, the consensus equation will drive the node states to the same value if the graph is static and connected.
- But, this is clearly not the case for mobile agents in general:
  - **Edges = communication links**
    - Random failures
    - Dependence on the position (shadowing,…)
    - Interference
    - Bandwidth issues
  - **Edges = sensing**
    - Range-limited sensors
    - Occlusions
    - Weirdly shaped sensing regions
Directed Graphs

- Instead of connectivity, we need directed notions:
  - **Strong connectivity** = there exists a directed path between any two nodes
  - **Weak connectivity** = the disoriented graph is connected

- Directed consensus:

\[
\dot{x}_i = - \sum_{j \in N_i^{in}} (x_i - x_j)
\]
Directed Consensus

- Undirected case: Graph is connected = sufficient information is flowing through the network
- Clearly, in the directed case, if the graph is strongly connected, we have the same result

- Theorem: If $G$ is strongly connected, the consensus equation achieves
  \[
  \lim_{t \to \infty} (x_i - x_j) = 0, \ \forall i, j
  \]

- This is an unnecessarily strong condition! Unfortunately, weak connectivity is too weak.
Spanning, Outbranching Trees

- Consider the following structure

- Seems like all agents should end up at the root node

- Theorem [2]: Consensus in a static and directed network is achieved if and only if G contains a *spanning, outbranching tree*. 
Where Do the Agents End Up?

- Recall: Undirected case
  \[
  \lim_{t \to \infty} x_i(t) = \bar{x}(0) = \frac{1}{N} \sum_{j=1}^{N} x_j(0), \quad \forall i
  \]

- How show that?
- The centroid is invariant under the consensus equation
  \[
  \dot{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \in N_i} (x_j - x_i) = 0
  \]

- And since the agents end up at the same location, they must end up at the static centroid (average consensus).
Where Do the Agents End Up?

- When is the centroid invariant in the directed case?
  
  \[ q^T L = 0, \quad w = q^T x \quad \Rightarrow \quad \dot{w} = q^T \dot{x} = -q^T L x = 0 \]

- \( w \) is invariant under the consensus equation
- The centroid is given by
  
  \[ \bar{x} = \frac{1}{N} 1^T x \]
  
  which thus is invariant if
  
  \[ 1^T L = 0 \]

- Def: \( G \) is balanced if
  
  \[ \text{deg}^\text{in}(i) = \text{deg}^\text{out}(i), \quad \forall i \in V \quad \Leftrightarrow \quad 1^T L = 0 \]

- **Theorem** [2]: If \( G \) is balanced and consensus is achieved then average consensus is achieved!
Dynamic Graphs

• In most cases, edges correspond to available sensor or communication data, i.e., the edge set is time varying

• We now have a switched system and spectral properties do not help for establishing stability

• Need to use Lyapunov functions
Lyapunov Functions

• Given a nonlinear system
  \[ \dot{x} = f(x) \]

• \( V \) is a (weak) Lyapunov function if
  
  \[ (i) \quad V(x) > 0, \quad \forall x \neq 0 \]
  
  \[ (ii) \quad \dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0, \quad \forall x \neq 0 \quad (\leq 0) \]

• The system is asymptotically stable if and only if there exists a Lyapunov function

• [LaSalle’s Invariance Principle] If it has a weak Lyapunov function the system converges asymptotically to the largest set with \( f=0 \) s.t. the derivative of \( V \) is 0
Similarly, consider a switched system

\[ \dot{x} = f_\sigma(x), \quad \sigma(t) \in \{1, \ldots, q\} \]

The system is *universally asymptotically stable* if it is asymptotically stable for all switch sequences.

A function \( V \) is a common Lyapunov function if it is a Lyapunov function to all subsystems

\[ V > 0, \quad \frac{\partial V}{\partial x} f_i < 0, \quad i = 1, \ldots, q \]

Theorem [9]: Universal stability if and only if there exists a common Lyapunov function. (Similar for LaSalle.)
Switched Networked Systems

- Switched consensus equation
  \[ \dot{x} = -L_\sigma x \]
- Consider the following candidate Lyapunov function
  \[ V(x) = \frac{1}{2} x^T x, \quad \dot{V}(x) = x^T \dot{x} = -x^T L_\sigma x \]
- This is a common (weak) Lyapunov function as long as \( G \) is connected for all times
- Using LaSalle’s theorem, we know that in this case, it ends up in the null-space of the Laplacians
Switched (Undirected) Consensus

**Theorem [2-4]:** As long as the graph stays connected, the *consensus equation* drives all agents to the same state value

\[
\lim_{t \to \infty} x_i(t) = \bar{x} = \frac{1}{N} \sum_{j=1}^{N} x_j(0)
\]
Collisions?
Adding Weights

- too far away
- too close
- just right
Adding Weights
Selecting the Weights

\[ \dot{x}_i = - \sum_{j \in N_i} w_{i,j} (\|x_i - x_j\|) (x_i - x_j) \]

- Formation Control
- Connectivity Maintenance
- Coverage Control
- Flocking and Swarming
- Patrolling
- Pursuit/Evasion

Weights Through Edge-Tensions

- How select appropriate weights?
- Let an edge tension be given by
  \[ \mathcal{E} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j} \mathcal{E}_{i,j}(\|x_i - x_j\|) \]

Weights Through Edge-Tensions

- How select appropriate weights?
- Let an edge tension be given by \( E = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{i,j} E_{i,j}(\|x_i - x_j\|) \)
- We get
  \[
  \frac{\partial E_{i,j}}{\partial x_i} = w_{i,j}(\|x_i - x_j\|)(x_i - x_j)
  \]
- Gradient descent
  \[
  \dot{x}_i = -\frac{\partial E}{\partial x_i} = -\sum_{j \in N_i} w_{i,j}(\|x_i - x_j\|)(x_i - x_j)
  \]
  \[
  \frac{dE}{dt} = \frac{\partial E}{\partial x} \dot{x} = -\left\|\frac{\partial E}{\partial x}\right\|^2 \quad \text{Energy is non-increasing!}
  \]

**Examples**

Standard, linear consensus!

\[
\mathcal{E}_{ij} = \frac{1}{2} \| x_i - x_j \|^2 \Rightarrow w_{ij} = 1
\]

\[
\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j)
\]

Examples

Unit vector (biology)

\[ \mathcal{E}_{ij} = ||x_i - x_j|| \Rightarrow w_{ij} = \frac{1}{||x_i - x_j||} \]

\[ \dot{x}_i = - \sum_{j \in N_i} \frac{x_i - x_j}{||x_i - x_j||} \]

**Examples**

**Formation control v.2**

\[ \mathcal{E}_{ij} = \frac{1}{2} (\| x_i - x_j \| - d_{ij})^2 \Rightarrow w_{ij} = \frac{\| x_i - x_j \| - d_{ij}}{\| x_i - x_j \|} \]

\[ \dot{x}_i = - \sum_{j \in N_i} \frac{(\| x_i - x_j \| - d_{ij})(x_i - x_j)}{\| x_i - x_j \|} \]

Examples

Connectivity maintenance

\[ E_{ij} = \frac{\|x_i - x_j\|^2}{\Delta - \|x_i - x_j\|} \Rightarrow w_{ij} = \frac{2\Delta - \|x_i - x_j\|}{(\Delta - \|x_i - x_j\|)^2} \]

\[ \dot{x}_i = -\sum_{j \in N_i} \frac{(2\Delta - \|x_i - x_j\|)(x_i - x_j)}{(\Delta - \|x_i - x_j\|)^2} \]

Weighted Consensus: Formation Control

Spatio-Temporal Formations

Chopra, Egerstedt, 2013.
And In the Air…

Wang, Ames, Egerstedt, 2016
Coming to a Toy Store Near You…
Formation Control v.3 ~ Coverage Control

- Objective: Deploy sensor nodes in a distributed manner such that an area of interest is covered

- Idea: Divide the responsibility between nodes into regions
Coverage Control

- The coverage cost:
  \[ J(x, \mathcal{W}) = \frac{1}{2} \sum_{i=1}^{N} \int_{\mathcal{W}_i} \|x_i - q\|^2 dq \]

- Simplify (not optimal):
  \[ \hat{J}(x) = \frac{1}{2} \sum_{i=1}^{N} \int_{\mathcal{V}_i(x)} \|x_i - q\|^2 dq \]

where the Voronoi regions are given by

\[ \mathcal{V}_i(x) = \{ q \in \mathcal{D} \mid \|x_i - q\| \leq \|x_j - q\| \} \]
Deployment

- Using a gradient descent (cost = weak Lyapunov function)

\[
\dot{x}_i = - \frac{\partial \hat{J}}{\partial x_i} \Rightarrow \frac{d}{dt} \hat{J} = - \left\| \frac{\partial \hat{J}}{\partial x} \right\|^2
\]

\[
\dot{x}_i = - \int_{V_i(x)} (x_i - q) dq = - \int_{V_i(x)} dq \left( x_i - \rho_i(x) \right)
\]

- We only care about directions so this can be re-written as Lloyd’s Algorithm [1]

\[
\dot{x}_i = \rho_i(x) - x_i
\]

center of mass of Voronoi cell \( i \)
Deployment

- Lloyd’s Algorithm:
  - Converges to a local minimum to the simplified cost
  - Converges to a Central Voronoi Tessellation

Courtesy of J. Cortes
Summary II

• Static Graphs:
  • Undirected: Average consensus iff G is connected
  • Directed: Consensus iff G contains a spanning, outbranching tree
  • Directed: Average consensus if consensus and G is balanced
• Switching Graphs:
  • Undirected: Average consensus if G is connected for all times
  • Directed: Consensus if G contains a spanning, outbranching tree for all times
  • Directed: Average consensus if consensus and G is balanced for all times
• Additional objectives is achieved by adding weights (edge-tension energies as weak Lyapunov functions)
1. GRAPH-BASED ABSTRACTIONS
2. FORMATION CONTROL
3. INTERACTING WITH NETWORKS
Again: Why Swarming Robots?

- Strength in numbers
- Lots of (potential) applications
- Convergence of technology and algorithms
- *Scientifically interesting!*

People will be part of the mix!
User Study

de la Croix, Egerstedt, 2014.
Results

- Performance “Error”, Difficulty, Workload

[Graph showing performance scores across tasks]

de la Croix, Egerstedt, 2014.
Results

- Performance “Error”, Difficulty, Workload

• PEOPLE ARE REALLY BAD AT CONTROLLING SWARMS OF ROBOTS!

de la Croix, Egerstedt, 2014.
A (Welsh) Mood Picture
Duck Tales
Lagrangian Fluid Dynamics

Lagrangian Swarms:
• Formation Control
• Flocking, Rendezvous, and Swarming
• Coverage Control
• Boundary Protection and Containment
• ...

control volume (CV)
control surface (CS)
Eulerian Fluid Dynamics

Eulerian Swarms?
Key idea: Human operator specifies areas of interest and the robots respond

\[ J(x) = \sum_{i=1}^{N} \int_{V_i(x)} \|x_i - q\|^2 \phi(q) dq \]

Gradient descent (Lloyd’s algorithm)

\[ \dot{x}_i = \rho_i(x) - x_i \]

center of mass of Voronoi cell \(i\)

Achieves a CVT:

\[ x_i(t) - \rho_i(x(t)) \rightarrow 0 \]
Time-Varying Density Functions

- Need time-varying human inputs $\phi(q, t)$

$$\frac{d}{dt} (x - \rho(x)) = 0 \implies \dot{x} = \left( I - \frac{\partial \rho}{\partial x} \right)^{-1} \frac{\partial \rho}{\partial t}$$

- Problem 1: First need to get to a CVT
- Problem 2: Inverse not always defined
- Problem 3: Not distributed
- Problem 4: Messy…

$$\frac{\partial \rho_i^{(k)}}{\partial x_j^{(\ell)}} = \int_{\partial V_{i,j}} \phi q^{(k)} \frac{x_j^{(\ell)} - q^{(\ell)}}{\|x_j - x_i\|} dq \quad - \quad \int_{\partial V_{i,j}} \phi \frac{x_j^{(\ell)} - q^{(\ell)}}{\|x_j - x_i\|} dq \int_{V_i} \phi q^{(k)} dq$$
Time-Varying Density Functions

\[
\frac{d}{dt} \left( x - \rho(x) \right) = 0 \quad \Rightarrow \quad \dot{x} = \left( I - \frac{\partial \rho}{\partial x} \right)^{-1} \frac{\partial \rho}{\partial t}
\]

- Problem 1: First need to get to a CVT
- Problem 2: Inverse not always defined
- Problem 3: Not distributed

\[
\left( I - \frac{\partial \rho}{\partial x} \right)^{-1} = I + \frac{\partial \rho}{\partial x} - \left( \frac{\partial \rho}{\partial x} \right)^2 + \cdots
\]

Solution: Add a Lloyd term and use a truncated Neumann Series:

\[
\dot{x} = \left( I + \frac{\partial \rho}{\partial x} \right) \left( \frac{\partial \rho}{\partial t} + \kappa (\rho - x) \right)
\]

\[x_i(t) - \rho_i(x(t)) \to 0^\ast\]

Lee, Diaz-Mercado, Egerstedt, TRO, 2015
Example 1: Precision Agriculture

Li, Diaz-Mercado, Egerstedt, 2015
Example 2: The Robotarium

MRI: A Shared, Remote-Access Multi-Robot Laboratory
So Far…  [www.robotarium.org]

Distributed Formation Control
K. Fathian, N. Gans, M. Spong

Fault-Tolerant Rendezvous
H. Park, S. Hutchinson

Attitude Synchronization
J. Yamauchi, M. Fujita

Since Jan. 2016:
115 robots, 21 research groups, 105 student projects
Example 2: The Robotarium
Example 3: Mind Control
Eulerian Approached Beyond Density Functions

Kingston, Egerstedt, 2011
Summary III

• Lagrangian swarms at the level of the individual agents
• Eulerian swarms from the users’ perspective:
  – Engage at the level of the team, not at the level of individuals
  – (For small team sizes, leader-follower control still works ok)
• Embedded humans (human-swarm interactions) is still a major area of research!
To Summarize

Open issues:

- formations
- human-swarm interactions
- complex dynamics?
- malicious behaviors?
- beyond geometry?
Thank You!


Thank You!

Lab members:

Smriti Chopra  Philip Twu  JP de la Croix  Musad Haque  Peter Kingston  Meng Ji  Ted Macdonald

Collaborators:

Yasamin Mostofi  Jeff Shamma  George Pappas  Ron Arkin  Mehran Mesbahi

Sponsors: