CS 182 Lecture 6: Multi-Robot Systems!

Professors: Ariel Procaccia and **Stephanie Gil**

Email: sgil@seas.harvard.edu

Prof. Gil Office hours: Wednesdays 2:30-3:30p

Last Time

- Constraint Satisfaction Problems
    - Search to solve *identification* (or assignment) problems
    - Different types of CSPs
        - Unary constraints
        - Binary constraints (graph coloring)
        - Global constraints (Cryptarithmetic problems)
    - How to make search over CSPs easier?
        - Fail fast!
            - Arc consistency
            - Variable ordering
            - Minimum values remaining
        - Local search
        - Structure (tree CSP solver)

This Time

- A new topic!
  - Start our foray into optimization
  - This will be more of a “fun” lecture – more focused on an overview of multi-robot systems and practical aspects of the problem
- My last lecture before maternity leave
  - My office hours will continue through the end of this month (i.e. I will still have OH next week)
- Prof. Procaccia will continue on the topic of optimization
  - And all of its beautiful theory

Course Topics (Full list on course website)

- Uninformed search
- Informed search
- Motion planning
- Constraint satisfaction problems
- Multi-robot systems

Search and Planning

Lecturer: Gil

- Intro to optimization
- Game theory
- AI game playing
- Stackelberg security games
- Bayesian networks

Optimization and Games

Lecturer: Proccacia

- Markov Decision Processes
- Reinforcement learning
- Decision trees
- Linear classification
- Neural networks
- Ethics

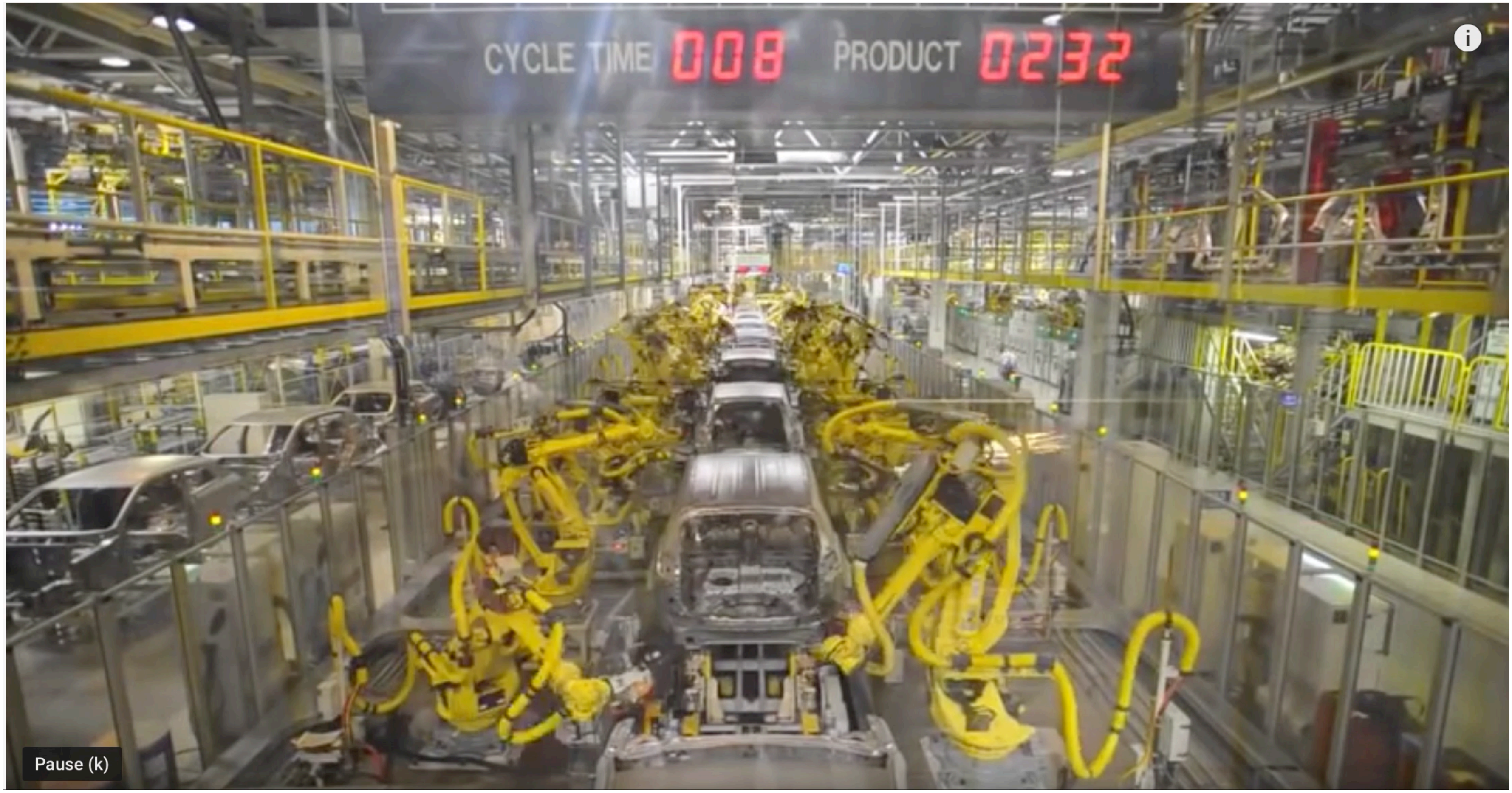
Learning and Uncertainty

Lecturer: Proccacia

Robots Today

A robot is a **machine** that can carry out **actions** in the physical world using computer **algorithms**. It often uses sensors to sense the world and **base its actions on sensory input**.

Manufacturing Robots



Robots in Space

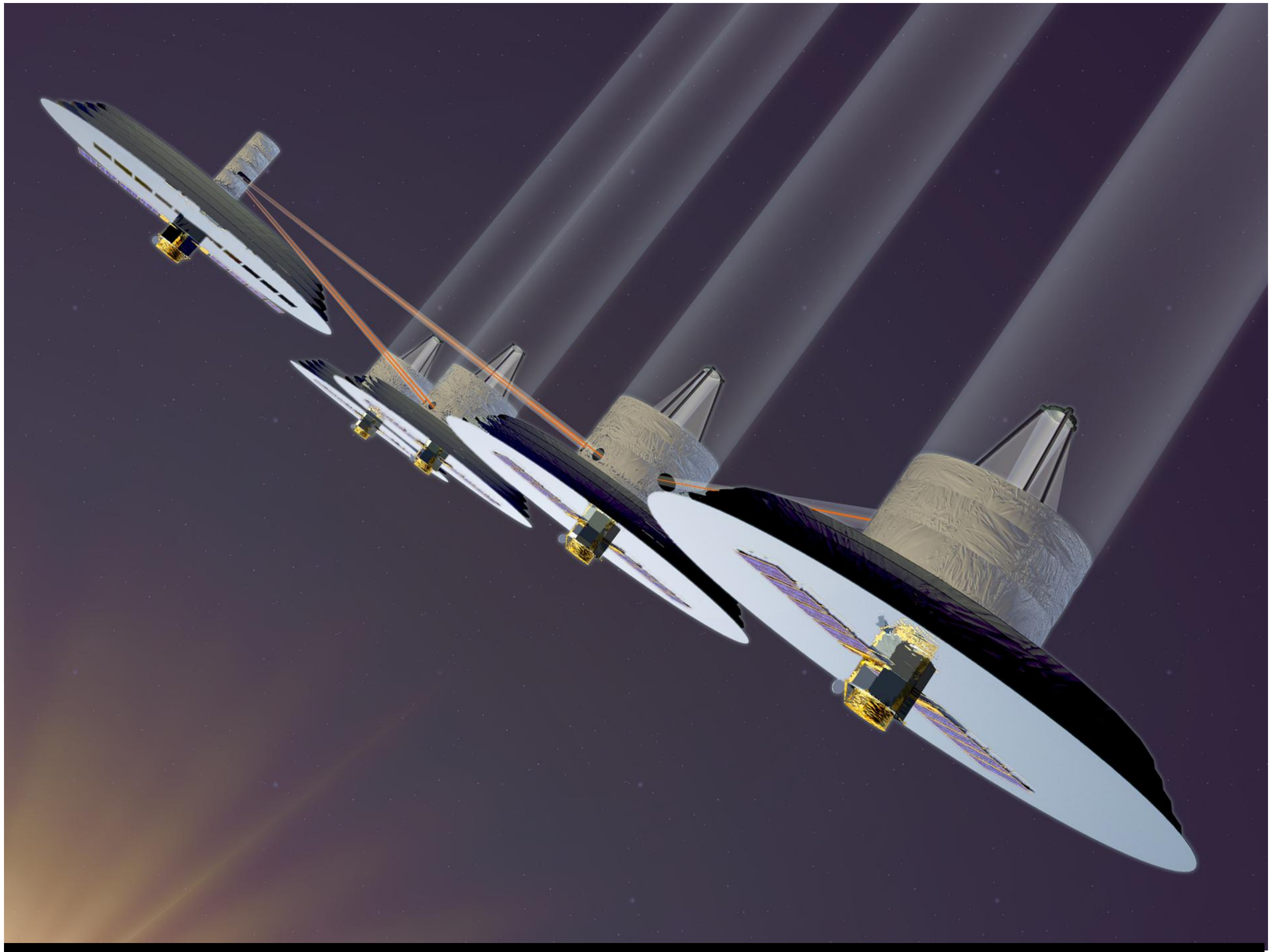


Image credit: JPL NASA Terrestrial Planet Finder mission (TPF)

Disaster Relief Robots

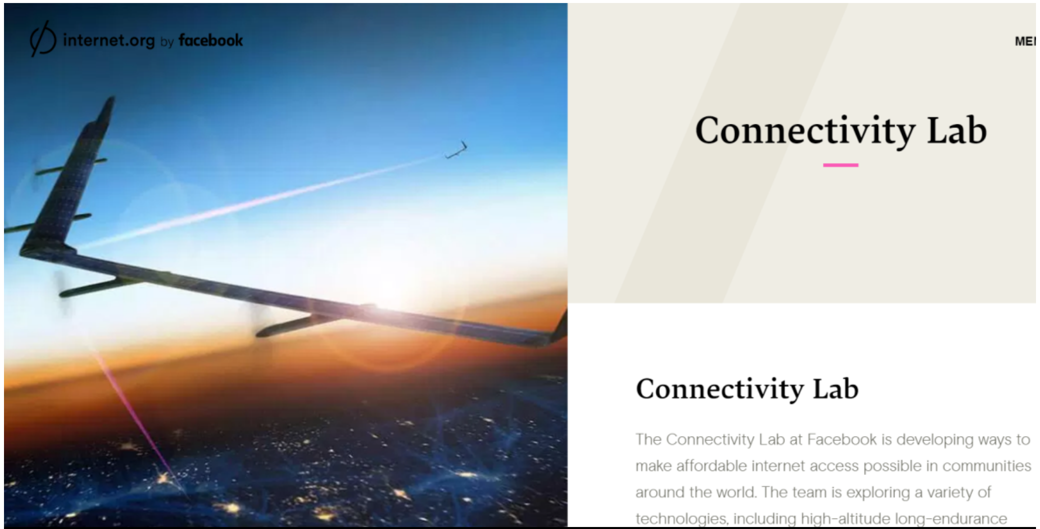
World Trade Center Crisis 2001

Fukushima Nuclear Disaster 2011

Nepal Earthquake 2015

Explosive Ordinance Disposal

On-Demand Communication



Connectivity Lab

Connectivity Lab

The Connectivity Lab at Facebook is developing ways to make affordable internet access possible in communities around the world. The team is exploring a variety of technologies, including high-altitude long-endurance

Facebook Connectivity Lab

Google Project Loon

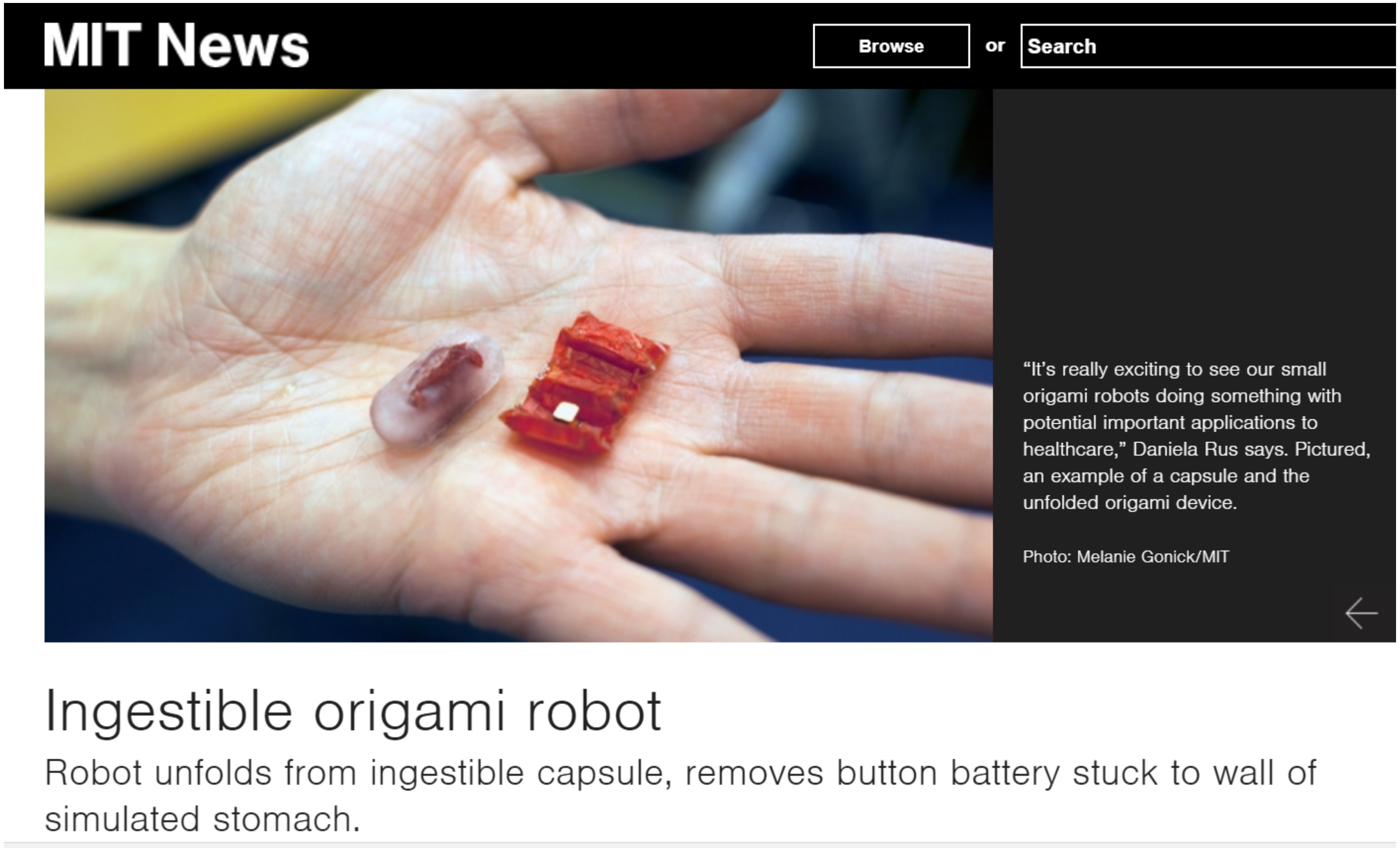
MIT Lincoln Labs Perdix Swarm

Delivery Robots

Image credit: DHL delivery services

Autonomous Vehicles

Medical Robots



"It's really exciting to see our small origami robots doing something with potential important applications to healthcare," Daniela Rus says. Pictured, an example of a capsule and the unfolded origami device.

Photo: Melanie Gonick/MIT

Ingestible origami robot

Robot unfolds from ingestible capsule, removes button battery stuck to wall of simulated stomach.

So how does it work?

- How can we make these robots **do** what we want them to do?
- Formulate as an *optimization problem* that we can encode!

Theory of Multi-Robot Systems

- A mathematical review (see optional reading “Distributed Control of Robotic Networks” by Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo, Sections 1.1-1.4, *available for free online*)

- The evolution of a multi-robot system described as a mathematical function

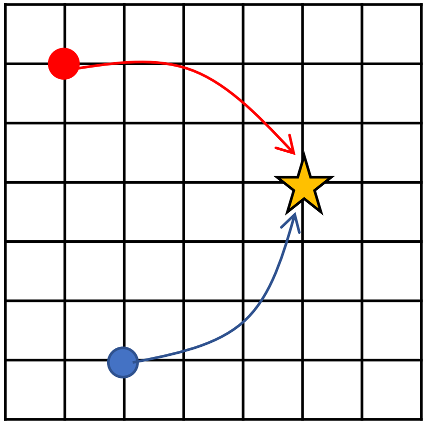
$f: X \times U \rightarrow X$ A *mapping*, describes the state evolution

$x(l+1) = f(x(l), u(l))$ State dynamics (discrete)

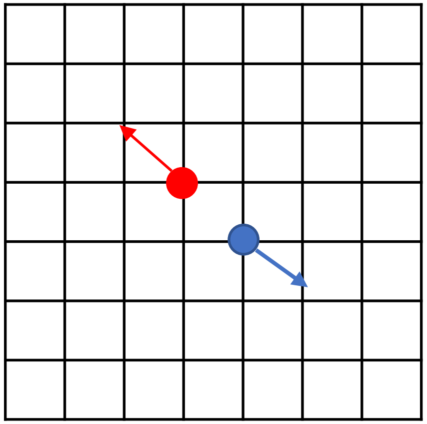
$\dot{x}(t) = f(t, x(t), u(t))$ State dynamics (continuous)

Theory of Multi-Robot Systems

- We are interested in how the state x evolves over *time* according to *f(x)*



Consenus (rendezvous)

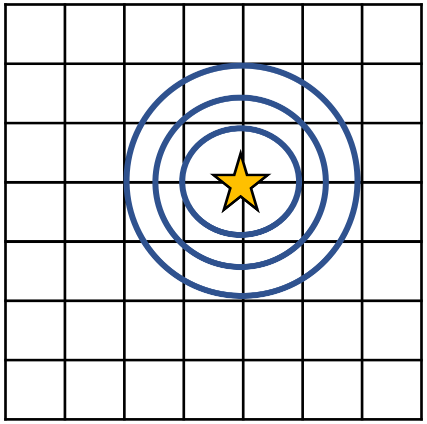


Collision avoidance

- *Equilibrium point* $f(x^*) = x^*$
- *Conditions under which* x^* *exists?*
- *Conditions under which* x^* *can be characterized and/or controlled*
- *These are called **Performance Guarantees***

Theory of Multi-Robot Systems

- Performance guarantees depend on many things:
  - *the function f*



Convex *f* ?
Nonconvex *f* ?

  - *connectivity (how agent i’s update affects agent j’s update)*

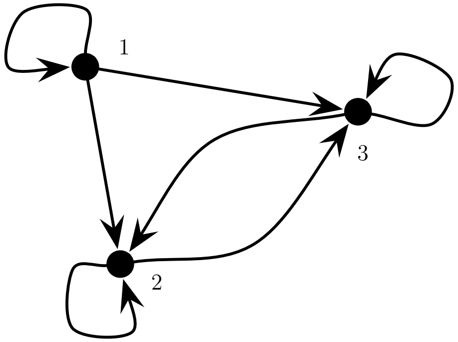


Fig 1.13 “Distributed Control of Robotic Networks”

  - *and other factors that we will discuss…*

How to Encode Robot State?

$f: X \times U \to X$ is the evolution map

State space Input space

- Discrete, no time dependence

$$x(l+1) = f\big(x(l), u(l)\big), \qquad l = \{1,2,..\}$$

- Discrete, time dependent

$$x(l+1) = f\big(l, x(l), u(l)\big), \qquad l = \{1,2,..\}$$

- Continuous, time dependent

$$\dot{x}(t) = f\big(t, x(t), u(t)\big), \qquad t \in R_{>0}$$

Types of Systems

Collective Swarm

- Act independently
- Minimal need for knowledge about other members of the system

Intentionally Cooperative

- Have knowledge of the presence of other robots in the environment
- Act together based on the **state, actions or capabilities** of their teammates in order to accomplish the same goal

How to Encode Relationships?

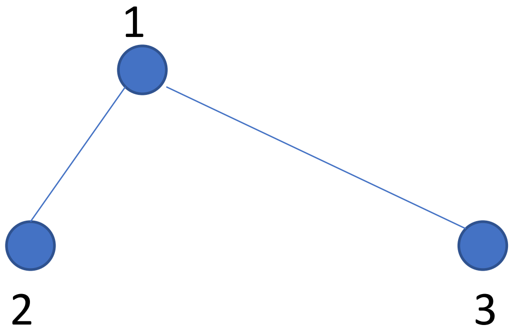
- A graph!

$$G = (V, E), \qquad E \subseteq V \times V$$

$V(G)$ – vertices of G

$E(G)$ – edges of G

(u, v)- $u, v \in V$ is the edge from u to v



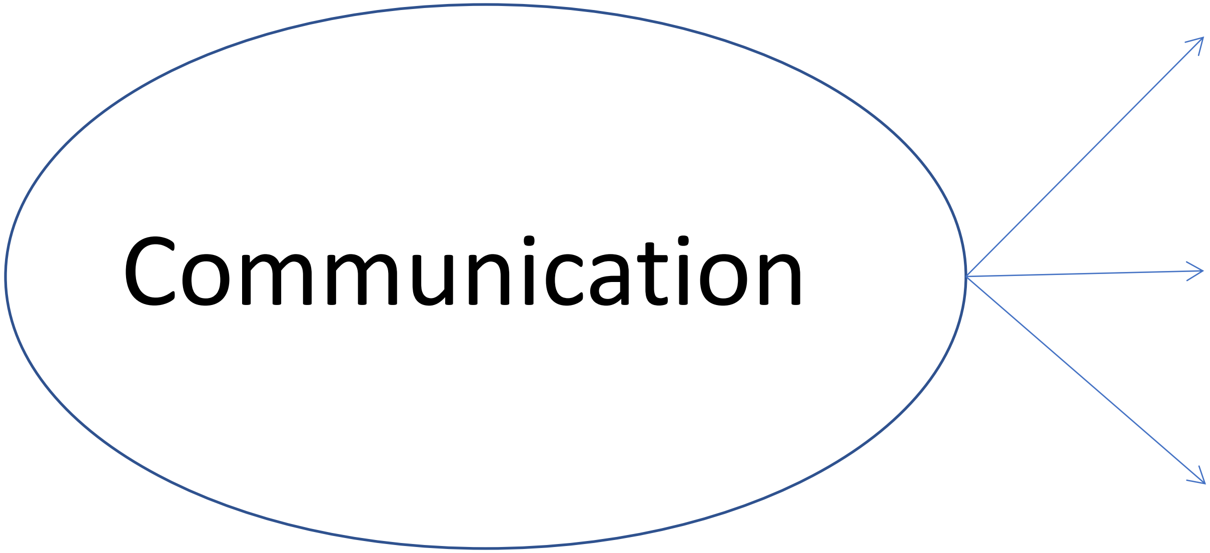
Different Control Architectures

- Centralized
- Hierarchical
- Decentralized

(examples of directed/undirected?)

- Hybrid

Different Communication Architectures

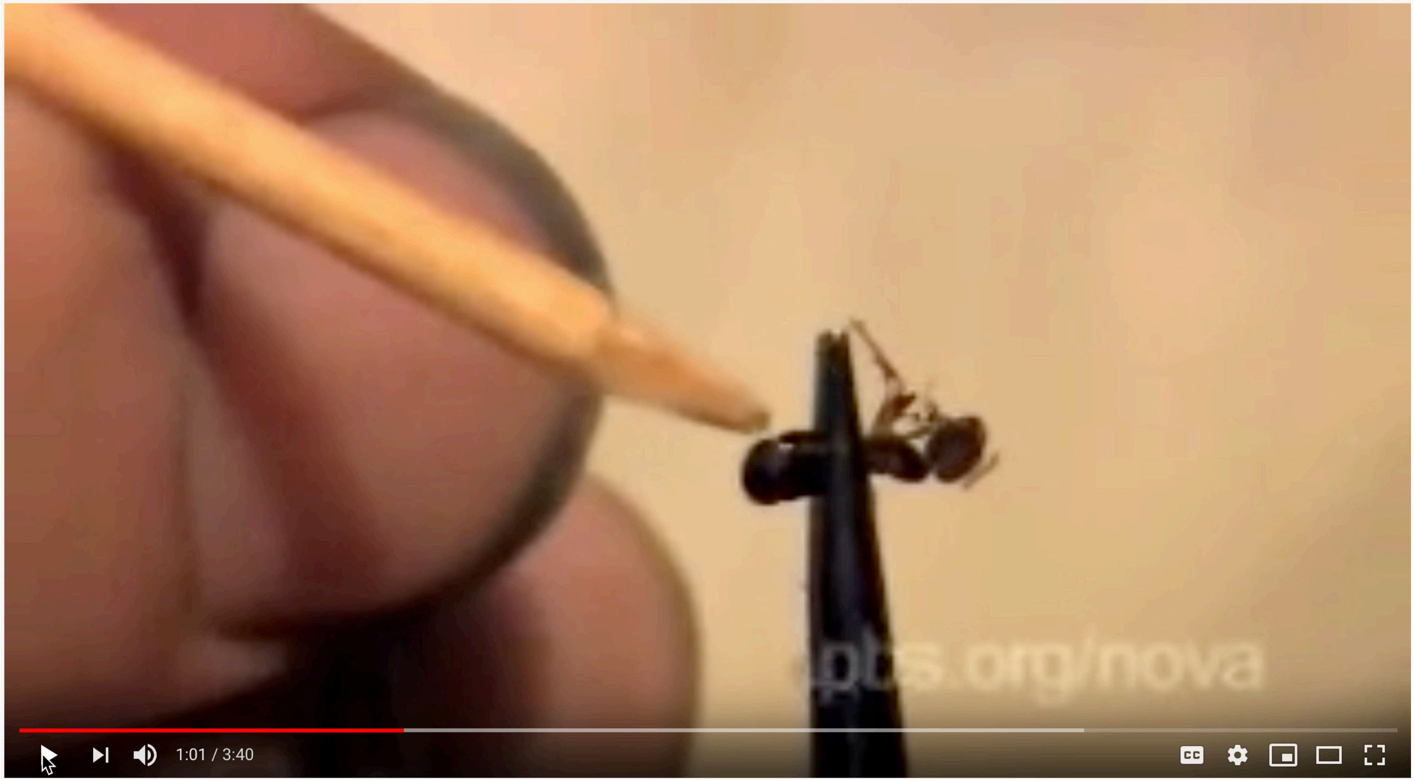


Communication

- Stigmergy
- Passive
- Explicit

Communication Architectures

- Stigmergy



Communication Architectures

- **Q1:** Stigmergy (as a graph?)

Communication Architectures

- Stigmergy

- Passive (as a graph?)

Communication Architectures

- Stigmergy
- Passive
- Explicit (can we make the previous graph “explicit” communication?)

Communication Architectures

- Stigmergy

- Passive

- Explicit

Architecture to Application

What control/comms architecture is needed in order to achieve different applications?

1) **Foraging/coverage**

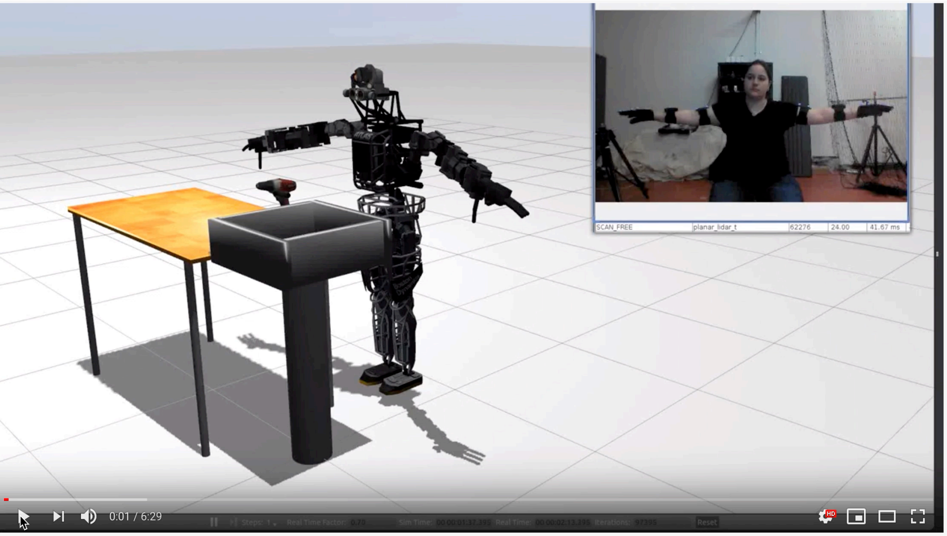
2) **Flocking/formations**

3) **Box pushing and cooperative manipulation**

4) **Traffic control and multi-robot path planning**

Classes of Networked Robots

Teleoperated



Video credit: MIT DRC Atlas Robot

- When is this good vs bad?

Autonomous

Video credit: JPL/NASA Curiosity Mission

Main Research Challenges

The broad challenge of *Autonomous Networked Robots* is to develop a science base that couples **communication**, **perception**, and **control** to enable new capabilities

Main Research Challenges: Networked Robots

- Networking
- Communication
- Control
- Perception
- Decision Making
- Adaptation

What is the difference here?

How do these two relate?

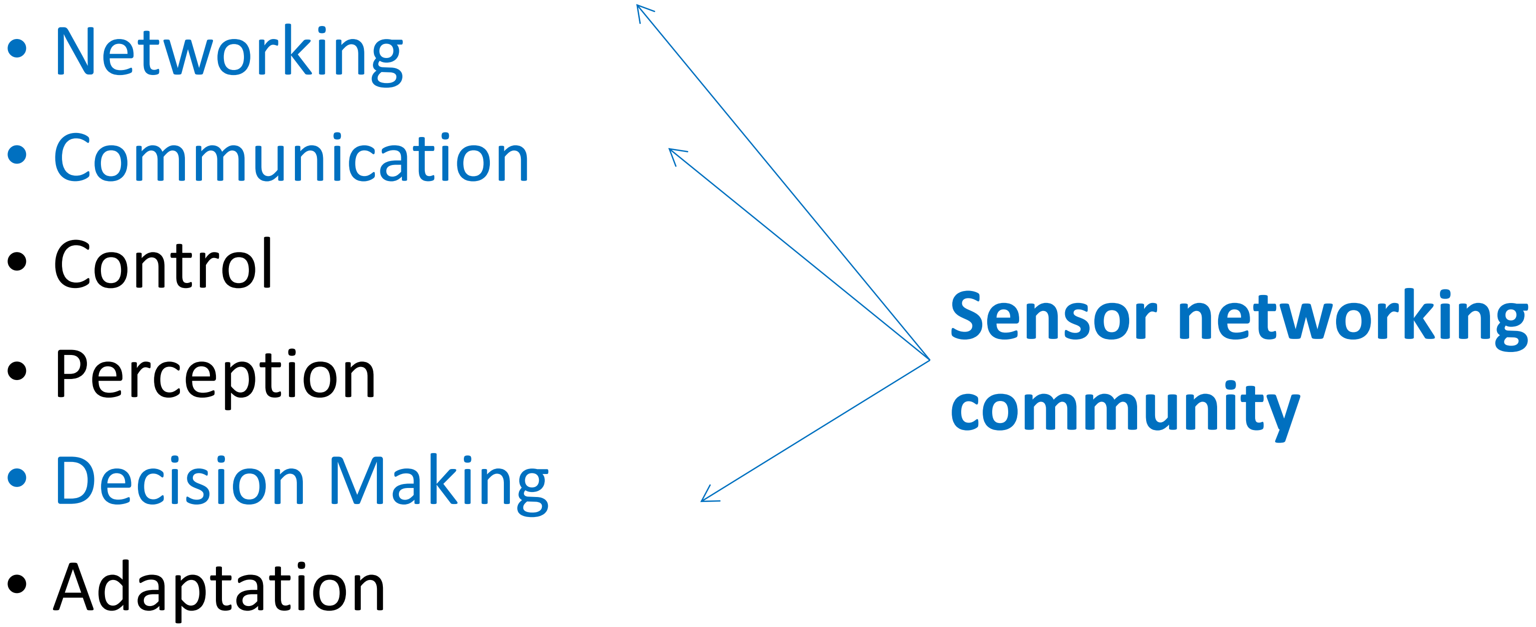
How does perception influence adaptation? State equations?

Main Research Challenges: Networked Robots

- Networking
- Communication
- Control
- Perception
- Decision Making
- Adaptation

Robotics community

Main Research Challenges: Networked Robots



- Networking
- Communication
- Control
- Perception
- Decision Making
- Adaptation

Sensor networking community

Main Research Challenges: Networked Robots

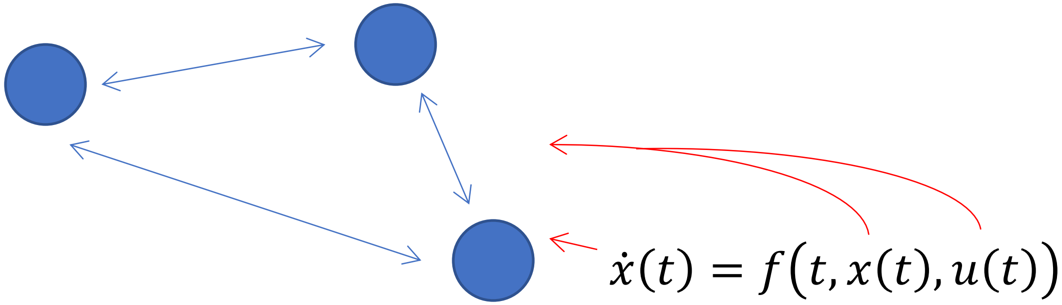
- Networking
- Communication
- Control
- Perception
- Decision Making
- Adaptation

AI community

Q2 **(polls everywhere):** So what is the difference between robot networks and sensor or computer networks?

Dynamics + Graph

- Dynamics allows us to describe individual behavior
- Graphs allow us to go from **individual** motion/behavior to **group behavior**.



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

Controllable Group Behavior

We need:

Individual controls to achieve a specified *aggregate motion* and *shape* of the group

Starlings Murmuration. Video credit: UZoo

Graph Theory

- The adjacency matrix captures the influence of agents’ states on one another
    - Draw the adjacency matrix of the visibility graph example

Connectivity in Multi-Robot Systems

- **Q3:** Why do we care?

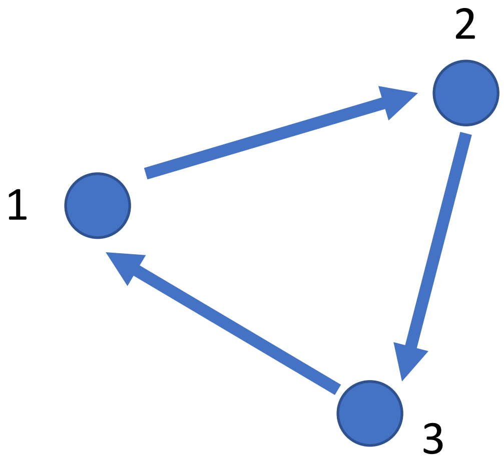
Graph Theory

- The adjacency matrix captures the influence of agents’ states on one another
    - Draw the adjacency matrix of the visibility graph example

- The adjacency matrix tells us a lot about the underlying graph structure and topology
    - The (i,j)th entry of A^k equals the number of directed paths of length k from node i to node j [see Section 1.3.5 of “Distributed Control of Robotics Networks”]

Adjacency Matrix and Connectivity (cont.)

- $A^{k+1} = AA^k$
- $\left(A^{k+1}\right)_{ij} = \sum_{l=1}^{n} A_{il}\left(A^k\right)_{lj}$
- $A_{dir}^2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
- $A_{dir}^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$



- Show that the entry $\left(A^k\right)_{ij}$ equals the # of directed paths from i to j of length k?

Notions of Connectivity

- Globally reachable – A vertex of a directed graph is globally reachable if it can be reached from any other vertex by traversing a directed path

- Strongly connected – A directed graph is strongly connected if every vertex is globally reachable
    - What is the difference with an undirected graph here?

Graph Laplacian

- $L(G) = D_{out}(G) - A(G)$
- $L(G)\mathbf{1}_n = \mathbf{0}_n$
- If G is strongly connected, then $rank(L(G)) = n - 1$, that is, 0 is a simple eigenvalue of L(G)
- G is undirected *iff* L(G) is symmetric

Group Behavior from Individual Controllers

Starlings Murmuration. Video credit: UZoo

Controlling a Large Group of Robot Agents

- Motivation

Source: “Rescue Robotics: An Introduction,” iRevolutions https://irevolutions.org/2015/08/10/rescue-robotics-introduction/

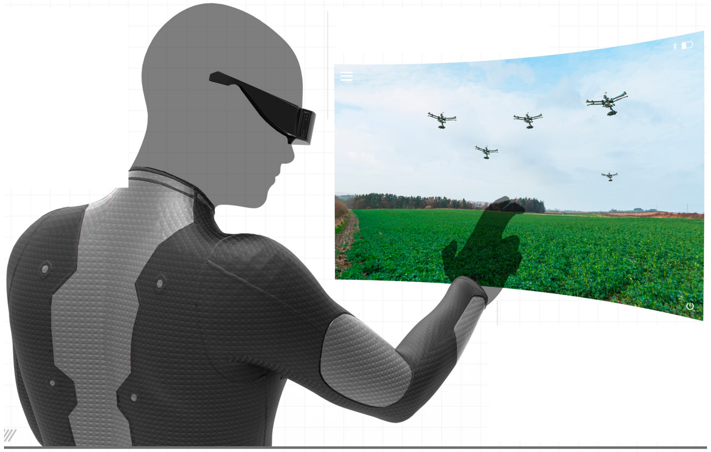


Image credit: https://www.behance.net/gallery/54081423/Wearable-Human-Swarm-Interaction-Technology

Controlling a Large Group of Robot Agents

- **Q4:** How might we do this? What are some inherent challenges here?

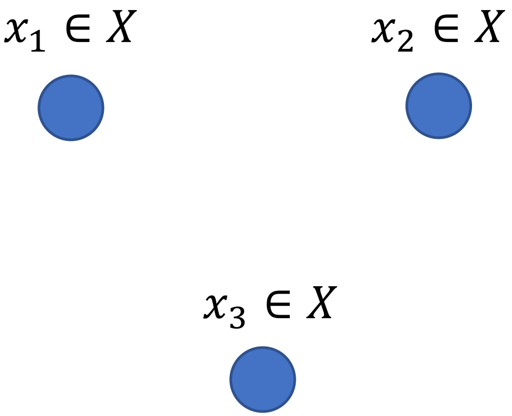
- Asymmetric Broadcast Control (ABC)
  - State space as Cartesian product

How to Encode Robot State?

$f: X \times U \rightarrow X$ is the evolution map

State space

Input space

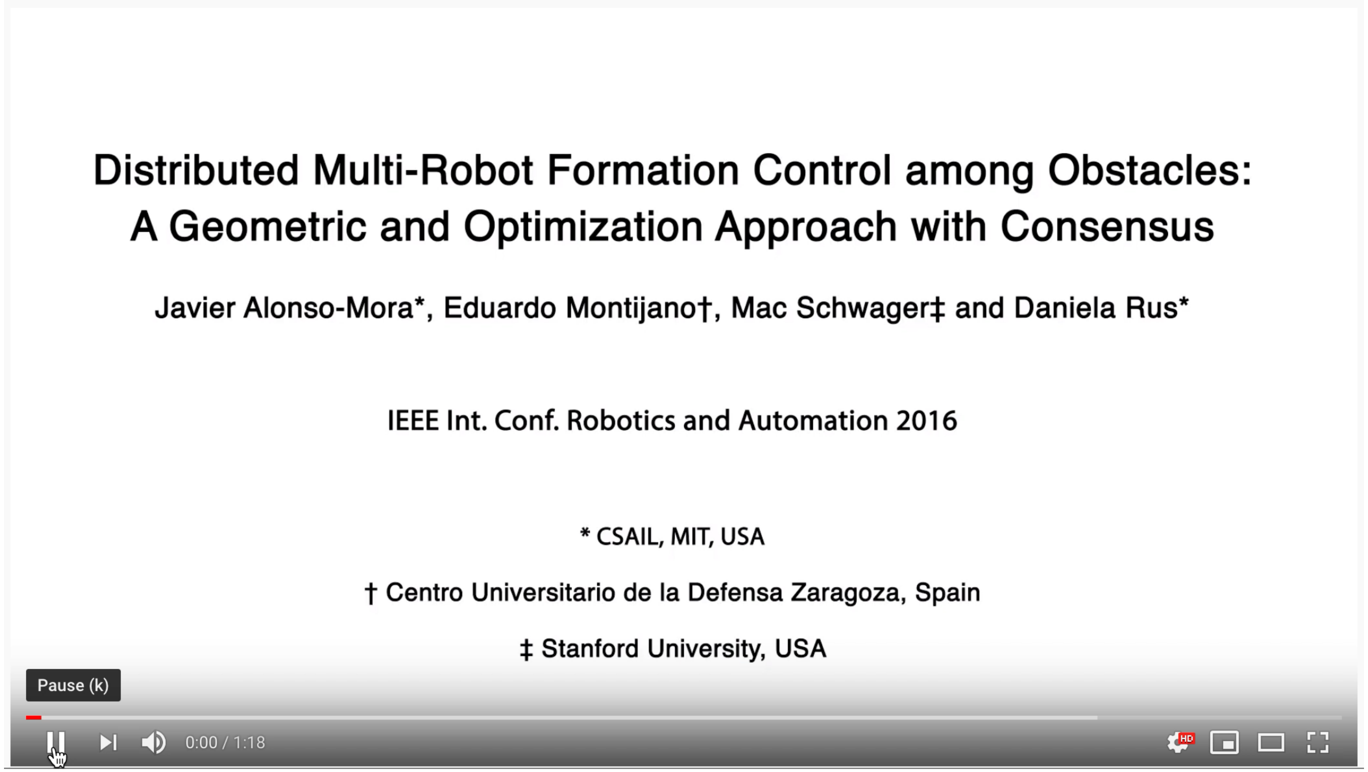


Controlling a Large Group of Robot Agents

- Asymmetric Broadcast Control (ABC)
  - State space as Cartesian product
  - Mapping: abstraction from high-dimensional to low-dimensional space

Mappings

- Example: an average
    - What if I want to control the *position* of a group?
    - Use the centroid



Distributed Multi-Robot Formation Control among Obstacles:
A Geometric and Optimization Approach with Consensus

Javier Alonso-Mora*, Eduardo Montijano†, Mac Schwager‡ and Daniela Rus*

IEEE Int. Conf. Robotics and Automation 2016

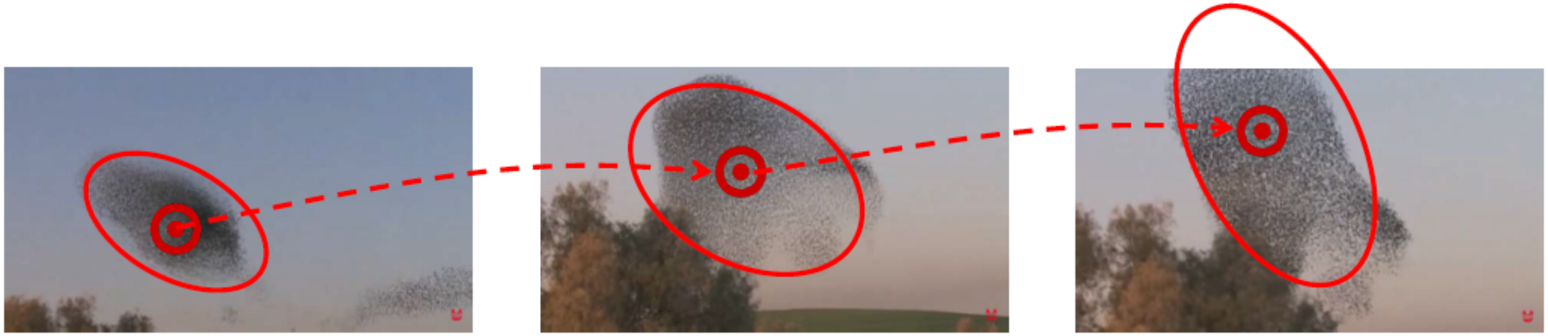
* CSAIL, MIT, USA

† Centro Universitario de la Defensa Zaragoza, Spain

‡ Stanford University, USA

Mappings (cont.)

- Example: a shape
  - What if I want to control the *position* and *shape* of a group?
  - Use an ellipsoid



t=1

t=2

t=3

Gradient Based Control

- Now that we know how to represent a state of the robot (or an aggregate state of the team), how do we *control it?*

- f: some function we wish to minimize

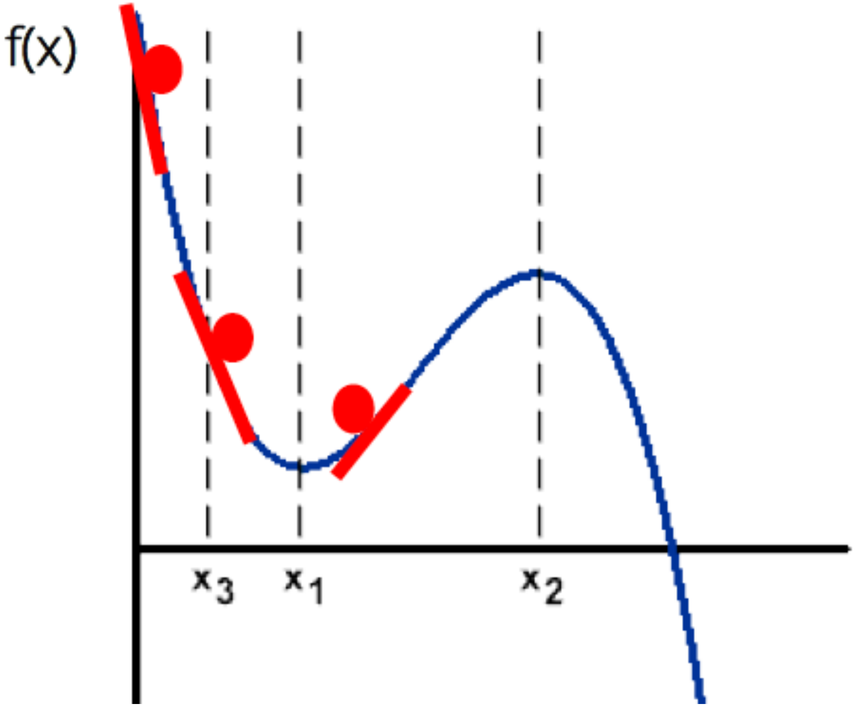
- One idea: gradient-based control

$$x(t+1) = x(t) + \overset{\text{step size}}{\overset{\swarrow}{\alpha}} u(t)$$

$$u(t) = -\frac{\partial f(x(t))}{\partial x(t)}$$

Potential Fields

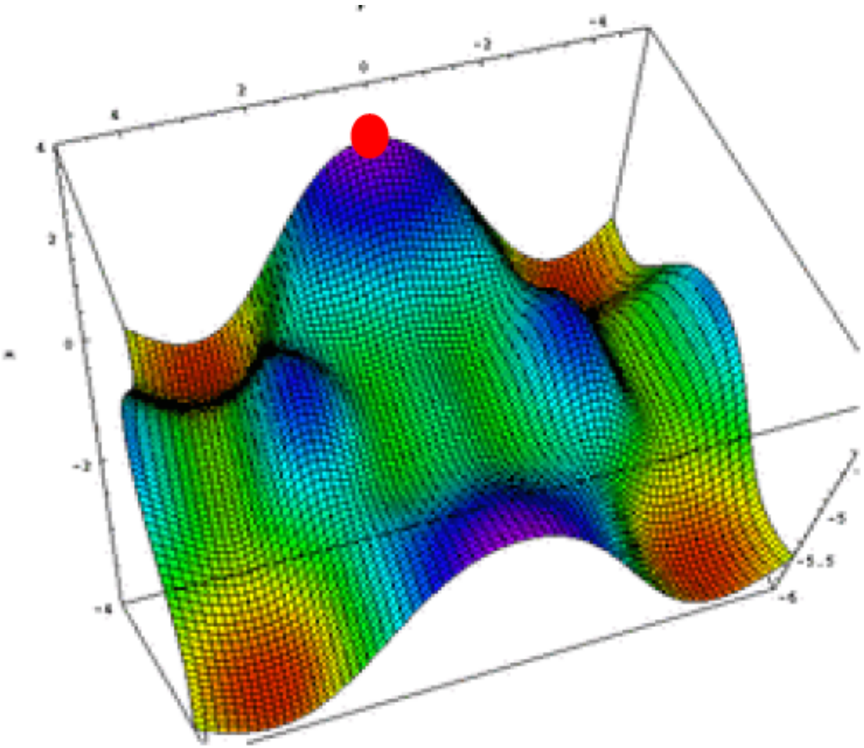
- What about a potential function?



- For n agents in d dimensional space?

Potential Fields for Higher Dimensions

- What about a potential function?



Potential Function for Navigation

- Obstacles and environmental boundaries are assigned high potential

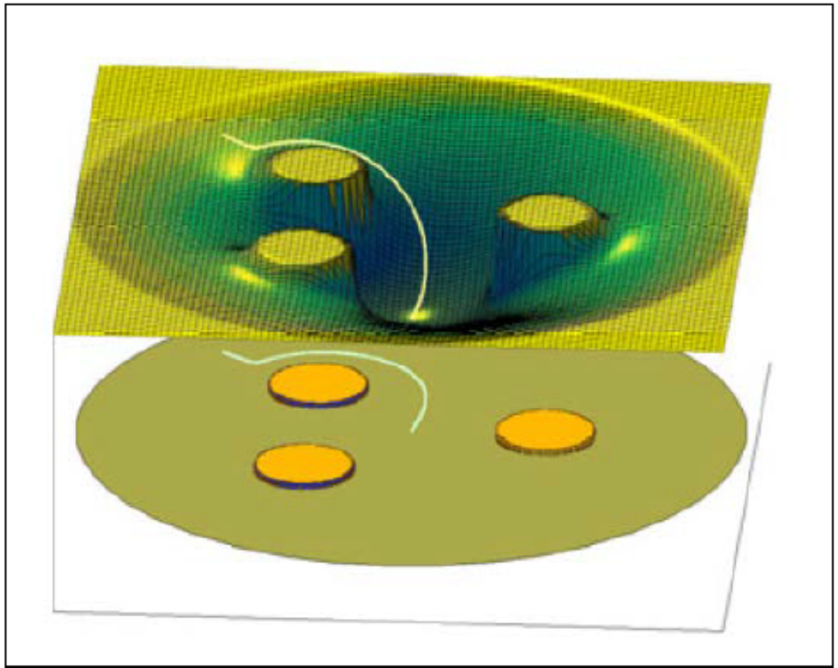
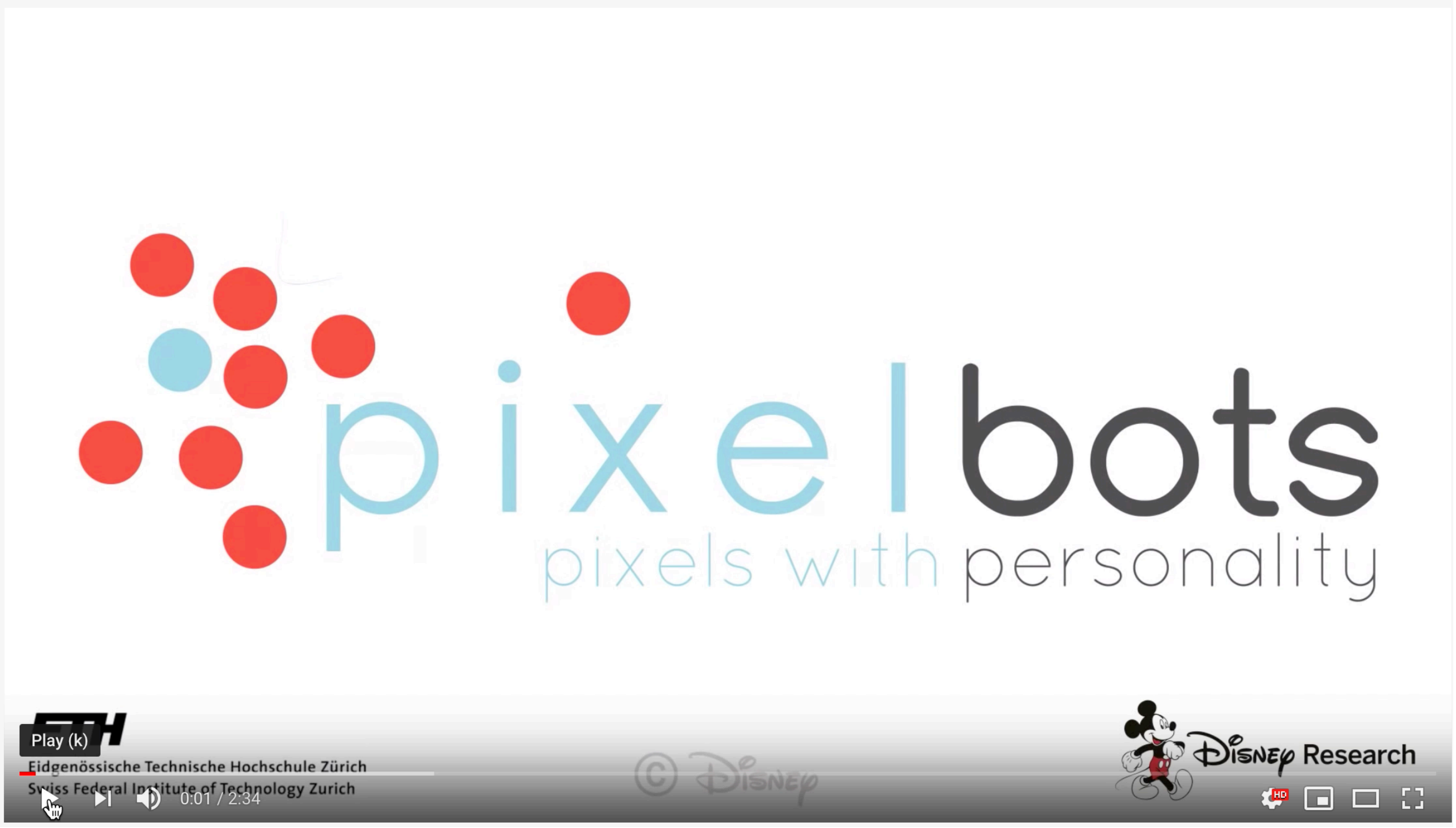


Image credit: Nathan Michael et.al.

Example: Motion Planning & Collision Avoidance



Next Time…

- Convex optimization