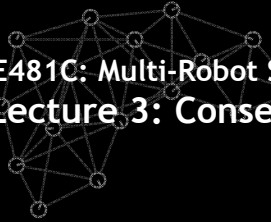


CSE481C: Multi-Robot Systems Lecture 3: Consensus



2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

1

intro

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

3

Misc

Announcements

- on reading papers
- on packet communications

Paper review

- convergecast aggregation
- database queries

Finish Lec2 Content

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

4

multi-robot computation model

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

6

Model: Robot State

We can describe the **state**, s , of a single robot as a tuple of its ID, pose, and private and public variables:



2008-01-06

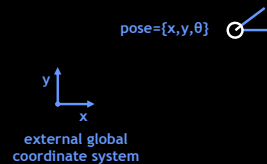
CSE481C wi09 - Robotics Capstone, Lec3: Consensus

7

Model: Robot State

We can describe the **state**, s , of a single robot as a tuple of its ID, pose, and private and public variables:

$$s = \langle \text{ID, pose, private vars, public vars} \rangle$$



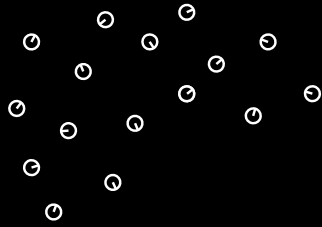
2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

8

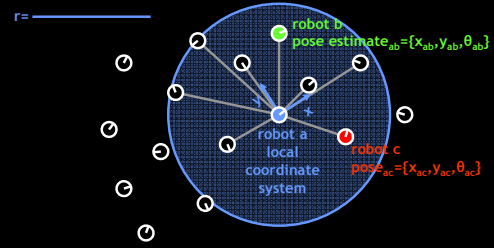
Model: Configuration

We define a **configuration**, C , as the states of n robots
All the robots use the same software and hardware



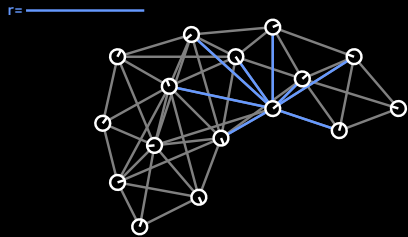
Model: Local Network Geometry

Each robot can communicate with and localize neighboring robots within radius r



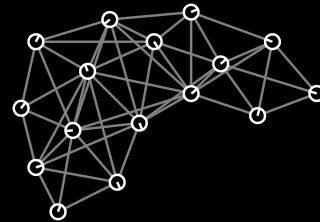
Model: Configuration Graph

A configuration C and communication radius r produces a **configuration graph** G
 C is **valid** iff G is connected



Model: Periodic Communications

Each robot broadcasts its **public vars** every τ seconds
We assume local communications are reliable
This creates a synchronizer, giving us global **rounds**



definition of terms

Self-Stabilizing Algorithm

Assume:

- Any initial configuration (state, position)
- That robots operate properly
- communications are reliable (perfect)

Provide:

- Proof that the system will stabilize to a desired configuration
- Show time and communications complexity

Complexity Measures

Computation:

- computation per round
- number of rounds
- time for robots to achieve final configuration

Communication:

- total number of messages
- messages per robot per round (bandwidth)

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

16

Errors

Two Types:

- process (robot) failures
- communications failures

Two Flavors:

- bounded quantity:
"At most one robot will fail"
- probabilistic
"Messages arrive with probability p "

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

17

leader election

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

19

Leader Election

Requirements:

- one process becomes leader
- other processes become not-leader

Bonus Requirement:

- all processes know which one is the leader

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

20

approaches

1. All processes start with same initial state

- If you have two identical processes, design an algorithm to elect one of them a leader.
- But only one execution possible on both processes
- Can't break symmetry \rightarrow impossibility proof - not possible to elect leader

2. Randomized Algorithm

- 1 random bit
- 50/50 change of electing leader on each flip
- How long will it take if graph is fully connected?
- How long will it take if graph is not fully connected?

3. Unique IDs

- break symmetry with deterministic algorithm
- can elect leader in bounded time
- how long will it take?

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

21

Problems

How to deal with removal of leader?

How to deal with multiple leaders?

How to elect two leaders?

- Running time and communications complexity?

k leaders?

- Running time and communications complexity?

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

22

consensus

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

24

What is consensus?

Simple:

- All processes agree on a quantity
- All processes know that they agree

Formal:

- Agreement:
 - no two processes decide on different values
- Validity
 1. If all processes start with 0, then 0 is the only possible decision value
 2. If all processes start with 1 and all messages are delivered, then 1 is the only possible decision value
- Termination
 - All processes eventually decide

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

25

the consensus card game

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

26

Whoa... Another Impossibility Result!

Consensus is not possible with faulty communications

- One of the most famous results in distributed algorithms
- (How do you get anything done with these systems, anyway?)

The proof uses the concept of "indistinguishable executions"

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

27

agreement algorithms

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

29

Agreement Algorithms

It's like consensus for real-valued quantities

- Processors share real-valued quantities
- All processors converge to the same quantity.

The papers this week are "hard"

- So I will introduce this content with two fun activities...

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

30

calculator agreement

2008-01-06 CSE4B1C wi09 - Robotics Capstone, Lec3: Consensus 32

Instructions:

- Enter your starting number into your calculator.
- Pick another person and average your two numbers. (Add theirs to yours and divide by two) Don't round off, keep all the digits. Both people should end up with the same number.
- Repeat 12 times. Try to visit different people.

a. $(30+10)/2 = 20$
 $(10+30)/2 = 20$

b. 20 and 20 average to 20 .
 20 and 90 average to 20 .

The answer is

65

2008-01-06 CSE4B1C wi09 - Robotics Capstone, Lec3: Consensus 34

Partial Proof

$$\sigma^2 = \frac{1}{2} [(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2]$$

$$\sigma^2 = \frac{1}{2} [x_1^2 - 2x_1\bar{x} + \bar{x}^2 + x_2^2 - 2x_2\bar{x} + \bar{x}^2]$$

$$\sigma^2 = \frac{1}{2} [x_1^2 + x_2^2 - 2\bar{x}(x_1 + x_2) + 2\bar{x}^2]$$

$$\sigma^2 = \frac{1}{2} \left[\left(\frac{x_1 + x_2}{2} \right)^2 + \left(\frac{x_1 - x_2}{2} \right)^2 \right]$$

$$= \left(\frac{x_1 + x_2}{2} \right)^2 + \frac{1}{4} (x_1 - x_2)^2$$

$$\sigma^2 < \bar{x}^2$$

$$\frac{x_1^2 + 2x_1x_2 + x_2^2}{4} - \bar{x}(x_1 + x_2) + \bar{x}^2 < \frac{1}{4} (x_1^2 - 2x_1x_2 + x_2^2) + \frac{1}{4} (x_1^2 + 2x_1x_2 + x_2^2)$$

$$\frac{1}{4} (x_1^2 + 2x_1x_2 + x_2^2) - \bar{x}(x_1 + x_2) + \bar{x}^2 < \frac{1}{4} (x_1^2 - 2x_1x_2 + x_2^2) + \frac{1}{4} (x_1^2 + 2x_1x_2 + x_2^2)$$

$$x_1^2 + 2x_1x_2 + x_2^2 < 2x_1^2 + 2x_2^2$$

$$2x_1x_2 < x_1^2 + x_2^2$$

Reference

2008-01-06 CSE4B1C wi09 - Robotics Capstone, Lec3: Consensus 36

Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms

JOSEPH N. D'ERSTOFF, ANDREW A. CHAN, ANDREW A. MANTON, ANDREW A. MANTON, ANDREW A. MANTON, ANDREW A. MANTON

The paper presents a family of distributed optimization algorithms that are asynchronous, deterministic, and stochastic. The algorithms are designed to solve a class of distributed optimization problems over a network of nodes. The algorithms are shown to converge to the optimal solution of the problem under a wide range of conditions. The algorithms are also shown to be robust to network changes and node failures.

Simulation

Legend:
 person 1 (blue)
 person 2 (orange)
 person 3 (green)
 person 4 (red)
 person 5 (purple)
 person 6 (brown)
 person 7 (pink)
 person 8 (grey)

2008-01-06 CSE4B1C wi09 - Robotics Capstone, Lec3: Consensus 37

Who Would Compute an Average Using this Crazy Technique?



Honeybees! Workers share food all the time, computing a global average. This lets an individual worker know when the *hive* is hungry by measuring when *she* is hungry.

Will This Work on the Robots?



poker chip agreement

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

41

lab 3: leader election and agreement

2008-01-06

CSE481C wi09 - Robotics Capstone, Lec3: Consensus

43