Distributed flocking control with a consensus algorithm

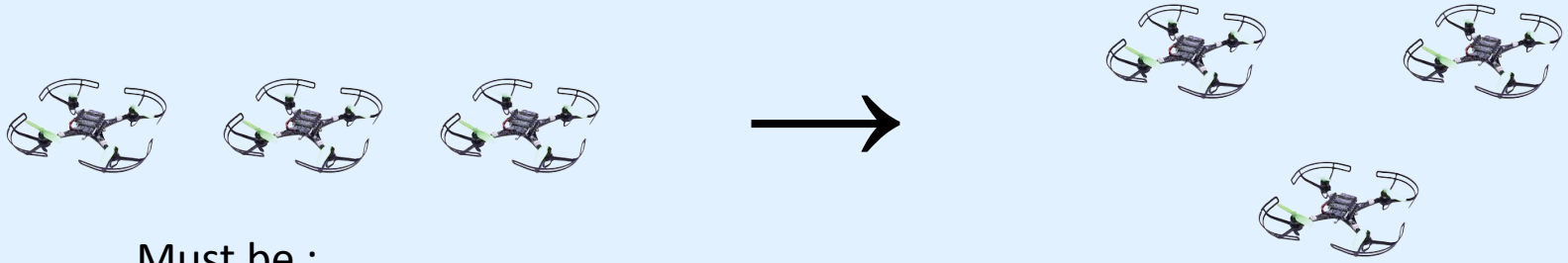
Laboratory of Tatsuya Ibuki at Ikuta campus, Meiji University, Japan
2nd year internship defense, Computer Science, ENSEIRB-MATMECA

Table of contents

1. Overview of the problem

Goal : Achieve a formation

Inputs : Drones positions
Outputs : Speed vectors



Must be :

- Relative to current position
- Maintained even if an agent drifts away from formation

Solution : **Consensus algorithm**

2. What is consensus ?

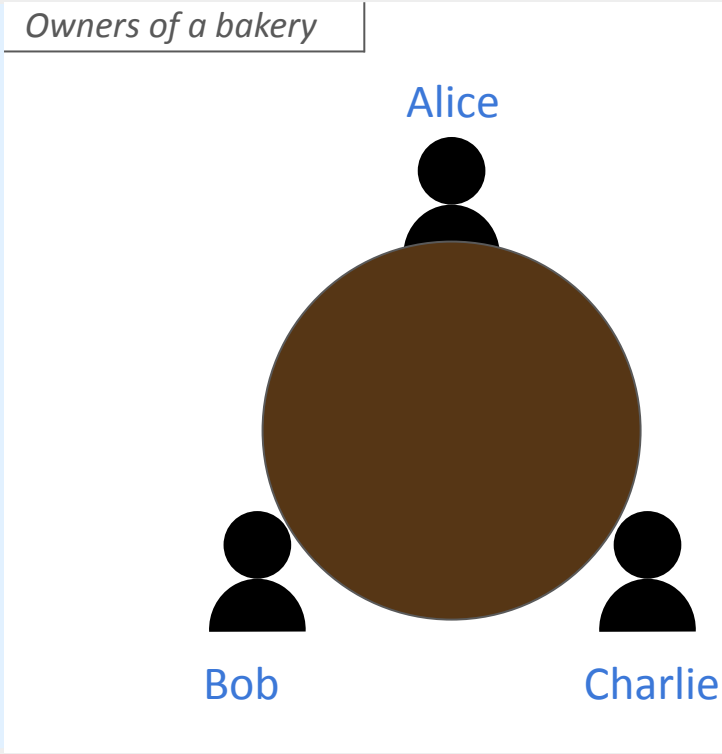
What is consensus ?

Consensus :

Collection of agents that want to agree on a common value over time

Example :

Deciding the sell price of a cookie in a bakery



Preliminary definitions

- A directed influence graph $G = (V, E)$

- Adjacency matrix : $A = [a_{ij}]$

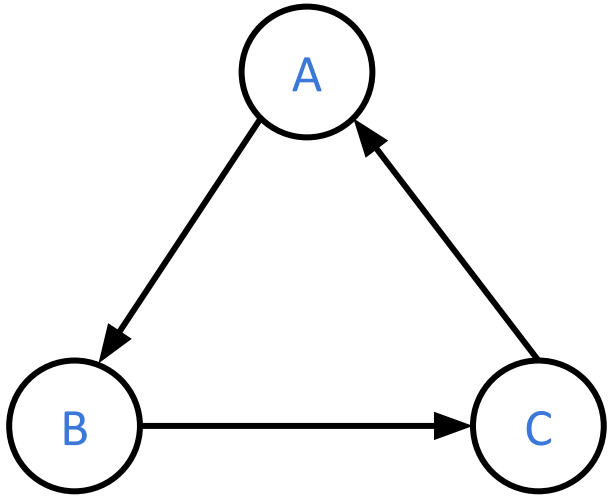
  $\rightarrow$ $a_{AB} = 1$ $\Leftrightarrow$ *Alice will be influenced by Bob*

- Neighbours of node i : $N_i = \{j \in V \mid a_{ij} \neq 0\}$

- State $x_i \in \mathbb{R}^n$ of an agent $i \in V$

  $\rightarrow$ *In the bakery,* $n = 1$ *and* x_i *is the current price of agent* i

Owners of a bakery

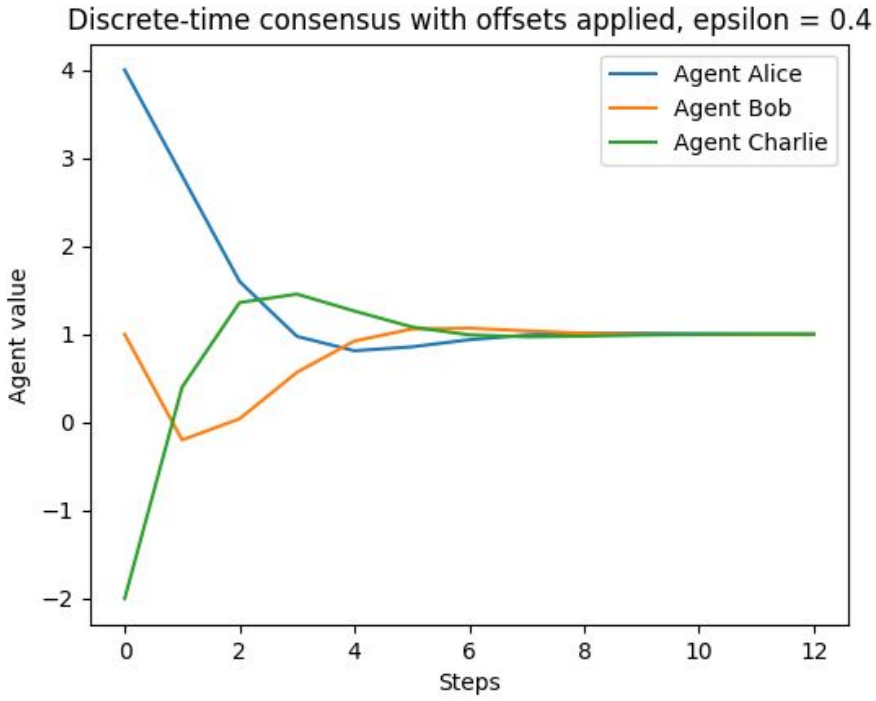


Consensus simulation

Discrete time consensus algorithm [1]

$$x_i[k+1] = x_i[k] + \epsilon \sum_{j \in N_i} x_j[k] - x_i[k]$$

Question : **How to leverage this algorithm to achieve formations ?**



[1] R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," in Proceedings of the IEEE, vol. 95, no. 1, pp. 215-233, Jan. 2007, doi: 10.1109/

Introducing offsets in consensus

Add new relative offset term

d_{ij} Relative offset from i to j

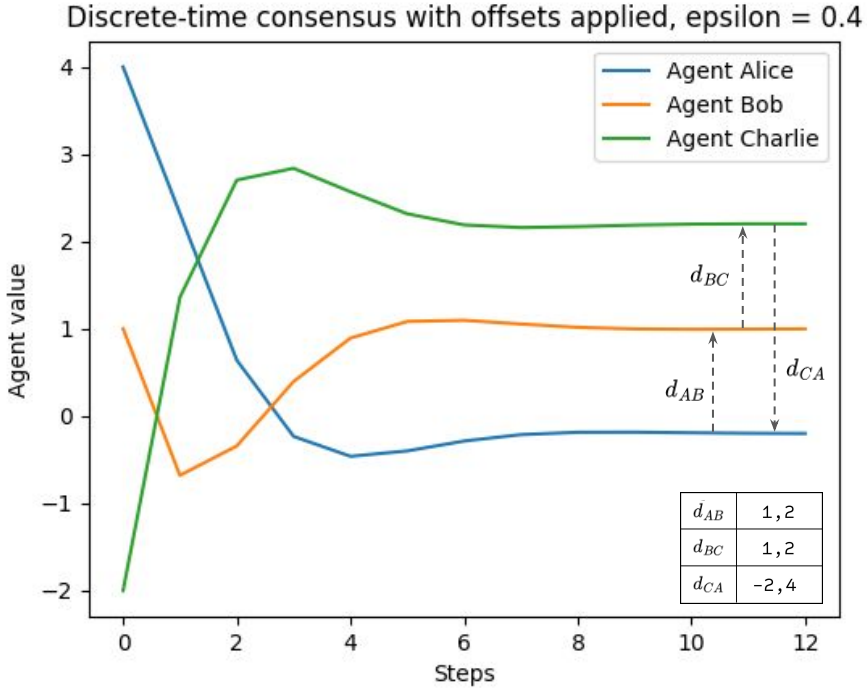
Acts like a distance condition between agents

Updated algorithm

$$x_i[k+1] = x_i[k] + \epsilon \sum_{j \in N_i} x_j[k] - x_i[k] - d_{ij}$$

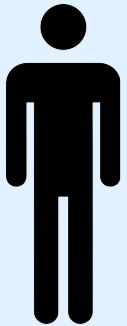
New convergence condition

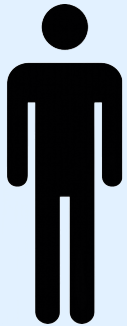
$$\sum_{\substack{i \in V \\ j \in N_i}} d_{ij} = 0$$



New convergence condition

$$\sum_{\substack{i \in V \\ j \in N_i}} d_{ij} = 0$$

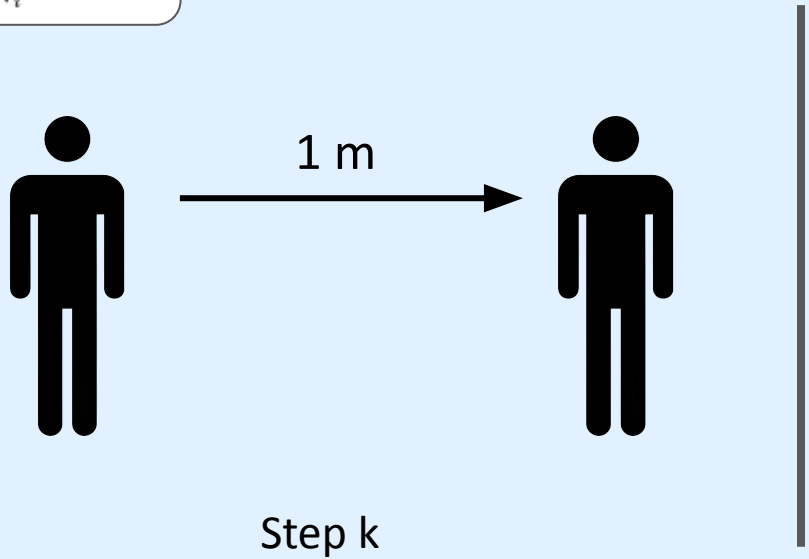




Step k

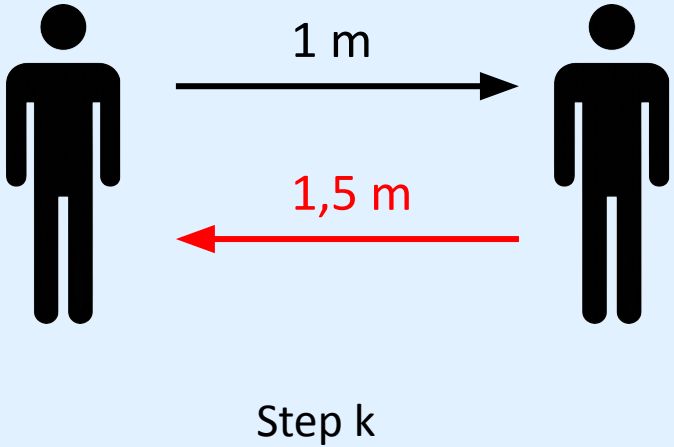
New convergence condition

$$\sum_{\substack{i \in V \\ j \in N_i}} d_{ij} = 0$$



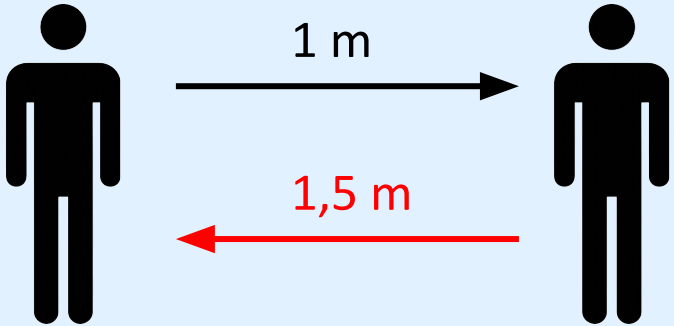
New convergence condition

$$\sum_{\substack{i \in V \\ j \in N_i}} d_{ij} = 0$$

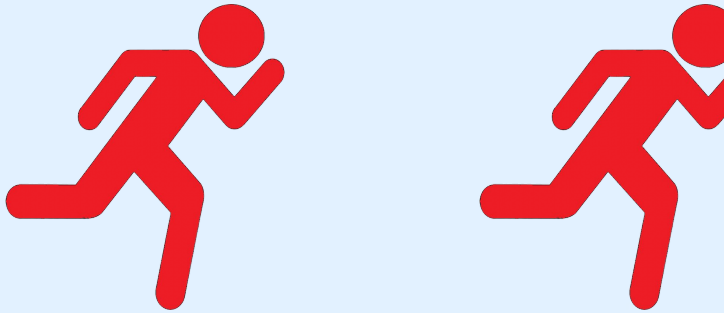


New convergence condition

$$\sum_{\substack{i \in V \\ j \in N_i}} d_{ij} = 0$$

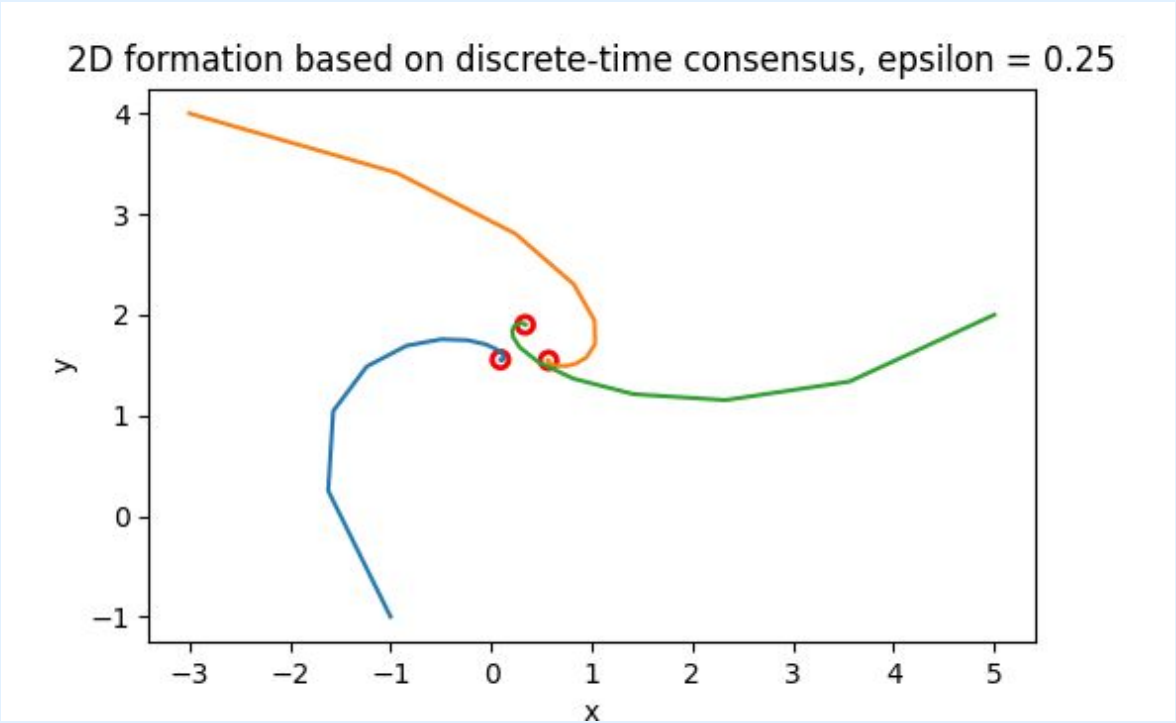


Step k



Step k+1

Consensus-based formation control

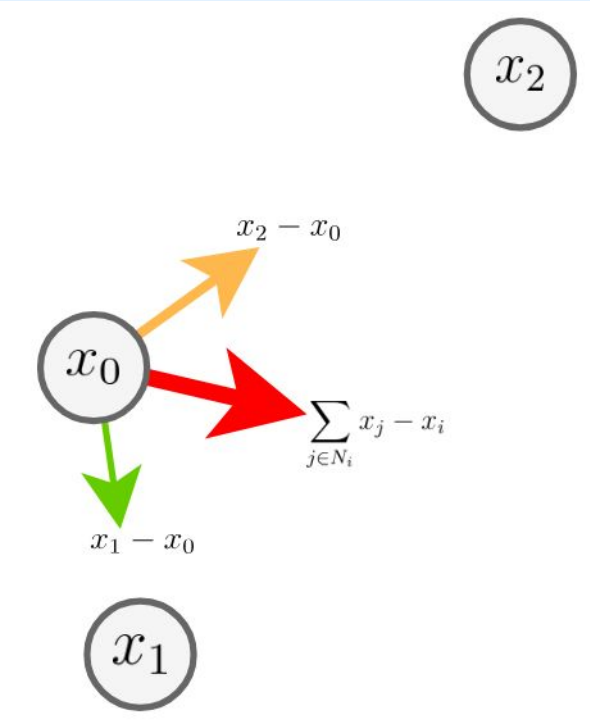


$x_i \in \mathbb{R}^2$

Same graph

Offsets for triangular formation

Applied to speed-driven systems



Consensus algorithm computes the state at time step k

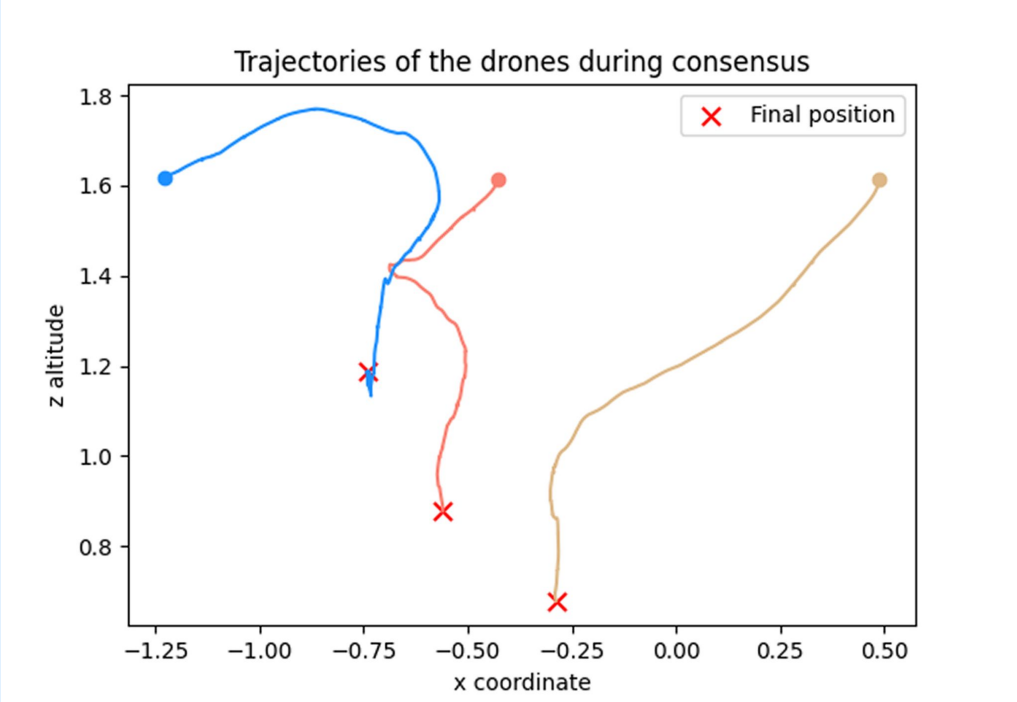
$$x_i[k+1] = x_i[k] + \epsilon \sum_{j \in N_i} x_j[k] - x_i[k] - d_{ij}$$

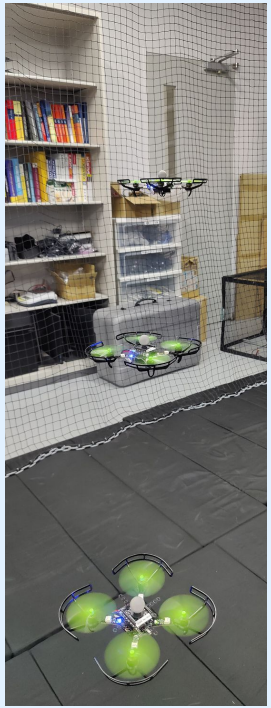
Second term ⇔ Sum of vector speeds

3. Experimentation results & conclusion

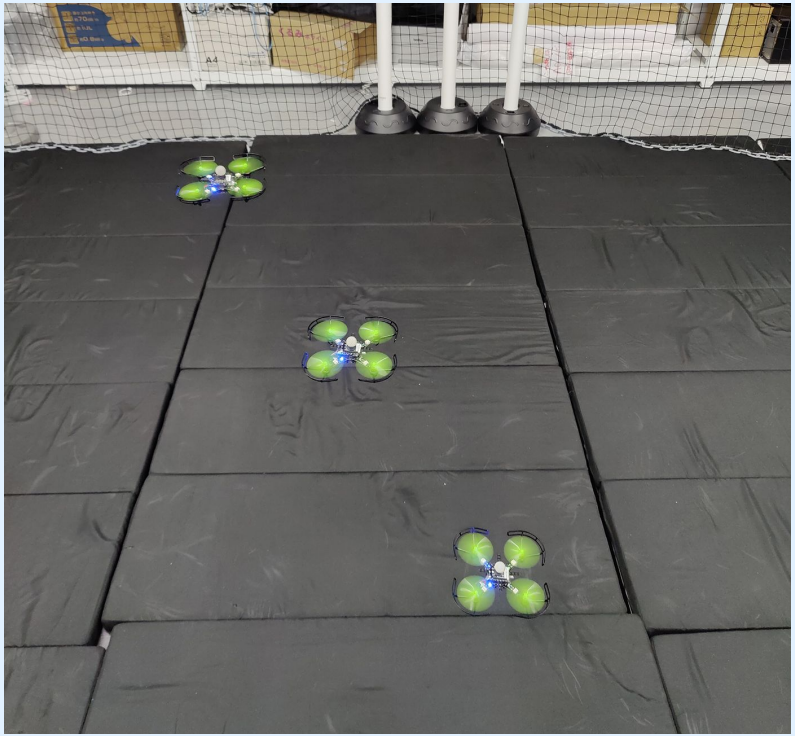
Experimentation results

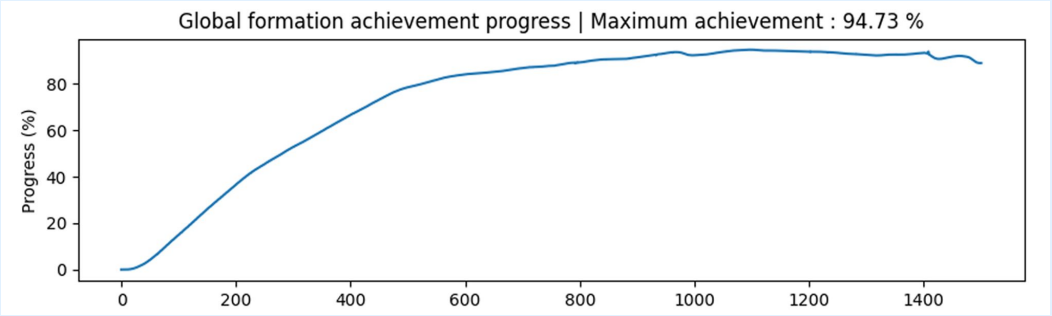
While falling, formation is still maintained

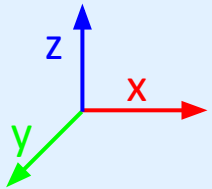




Experimentation results



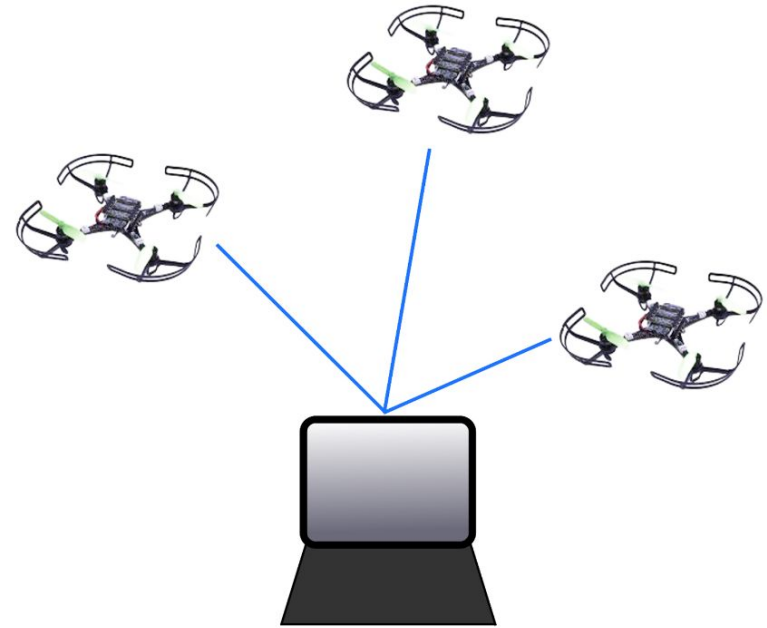




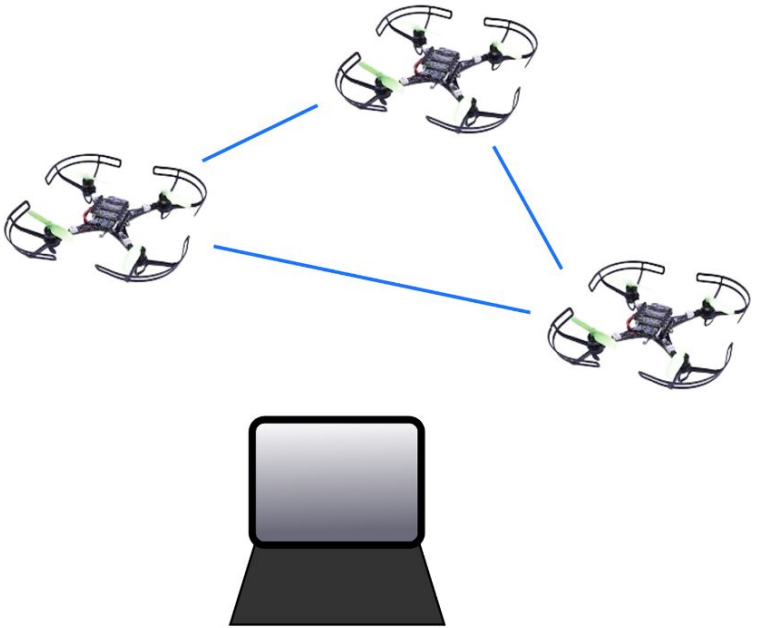
Error function : L2 norm

4. Distributed variant proof

Centralized vs distributed



Centralized



Distributed

From consensus to gradient descent

- Next step formula $x_{n+1} = x_n - \eta \nabla f(x_n)$

- For each agent, objective function is $G_i(x_i) = -\frac{1}{2} \sum_{j \in N_i} {x_j - x_i}^2$

  By defining $\eta = -\epsilon$ we fall back on the discrete consensus algorithm

- Common objective of all nodes $\min \sum_{i \in V} G_i(x_i)$

5. Conclusion & future work

Conclusion & Future work

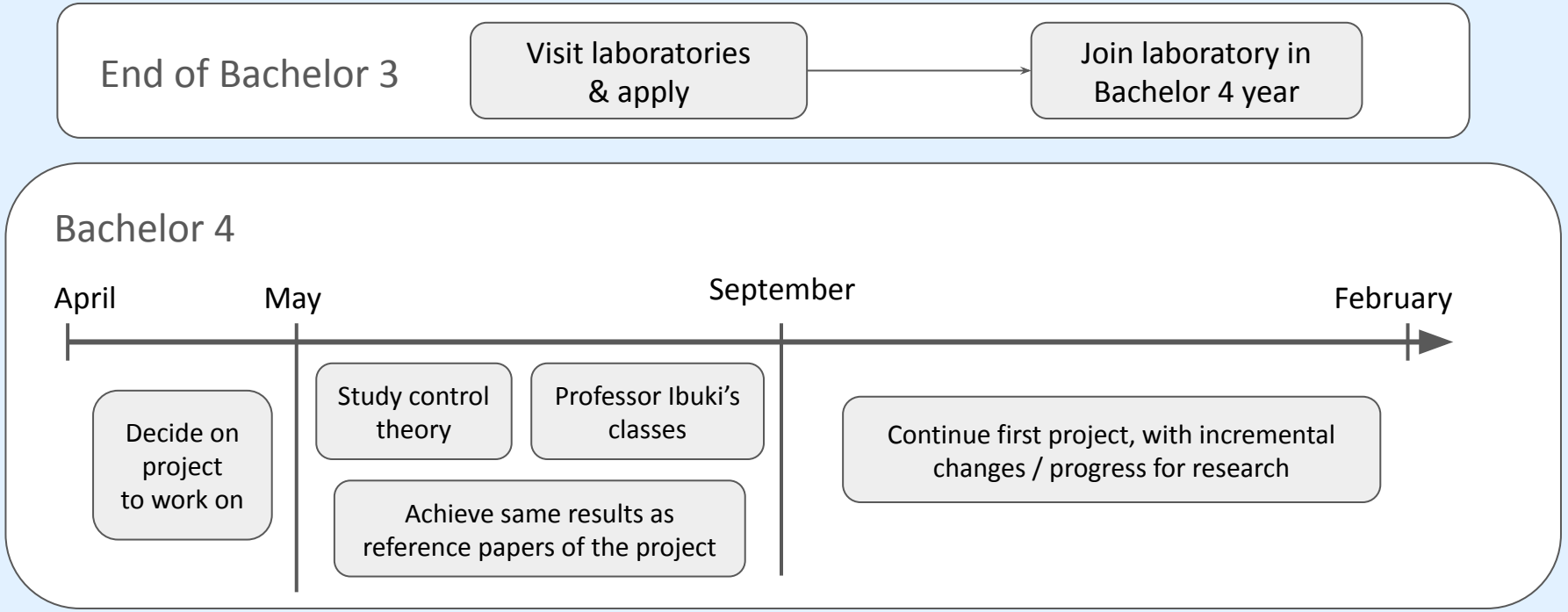
Conclusion

- Consensus-based formation implemented & validated
- Distributed variant convergence acknowledged by Professor Ibuki
- Satisfying results during simulation & experiments

Future work

- Convergence in a weakly connected graph
- Directed vs Undirected network consensus convergence speed
- Modifying offsets assignment to achieve formation faster

Addendum : Project lifecycle at Ibuki laboratory



Thank you for listening

This presentation was focused on my contributions and not everything could be mentioned
Let’s follow up with the questions