V-Formation as Optimal Control

Ashish Tiwari

SRI International, Menlo Park, CA, USA

BDA, July 25th, 2016

Joint work with Junxing Yang, Radu Grosu, and Scott A. Smolka

Outline

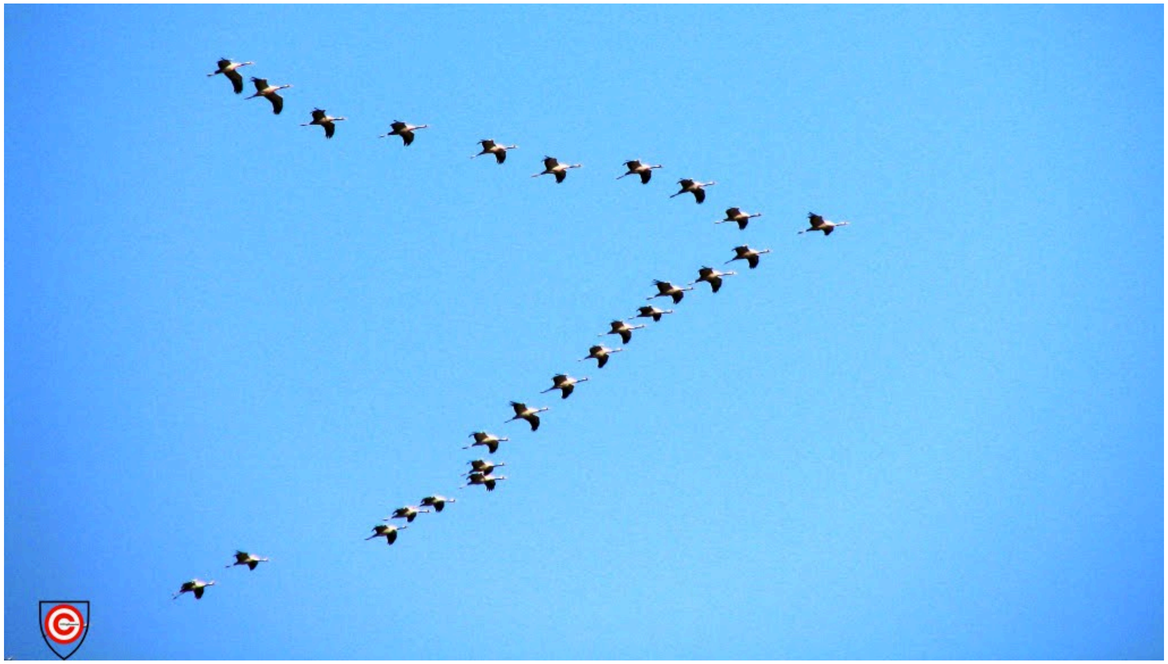
- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

Outline

- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

V-Formation

- Flocks of birds organize themselves into V-formations



Eurasian Cranes migrating in a V-formation (Hamid Hajihusseini, Wikipedia)

Reason: Saves energy as birds benefit from upwash region; provides clear visual field with visibility of lateral neighbors

Reaching a V-Formation

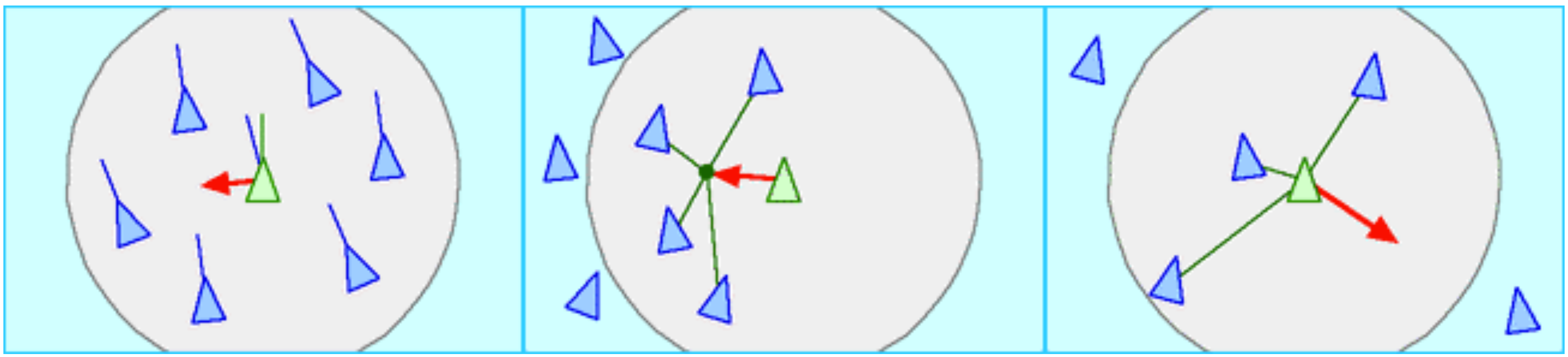
- Rule-based Approach:
  - Combinations of dynamical flight rules as driving forces
  - Not completely satisfying

- View as a Distributed Control Problem:
  - Flock wants to get into an optimal configuration that provides best view, energy benefit, and stability

- Our Approach:
  - Uses Model-Predictive Control (MPC)
  - Which uses Particle-Swarm Optimization (PSO)

Reynolds' Rules

Reynolds (1987) presented three rules for generating V-formations:



Alignment Cohesion Separation

Alignment: steer towards the average heading of local flockmates
Cohesion: steer to move toward the average position of local flockmates
Separation: steer to avoid crowding local flockmates

Extended Reynolds Model

Reynolds' model was extended by additional rules:

- A rule that forces a bird to move laterally away from any bird that blocks its view (Flake (1998)).
- Drag reduction rule: computing the induced drag gradient and steering along this gradient (Dimock & Selig (2003)).

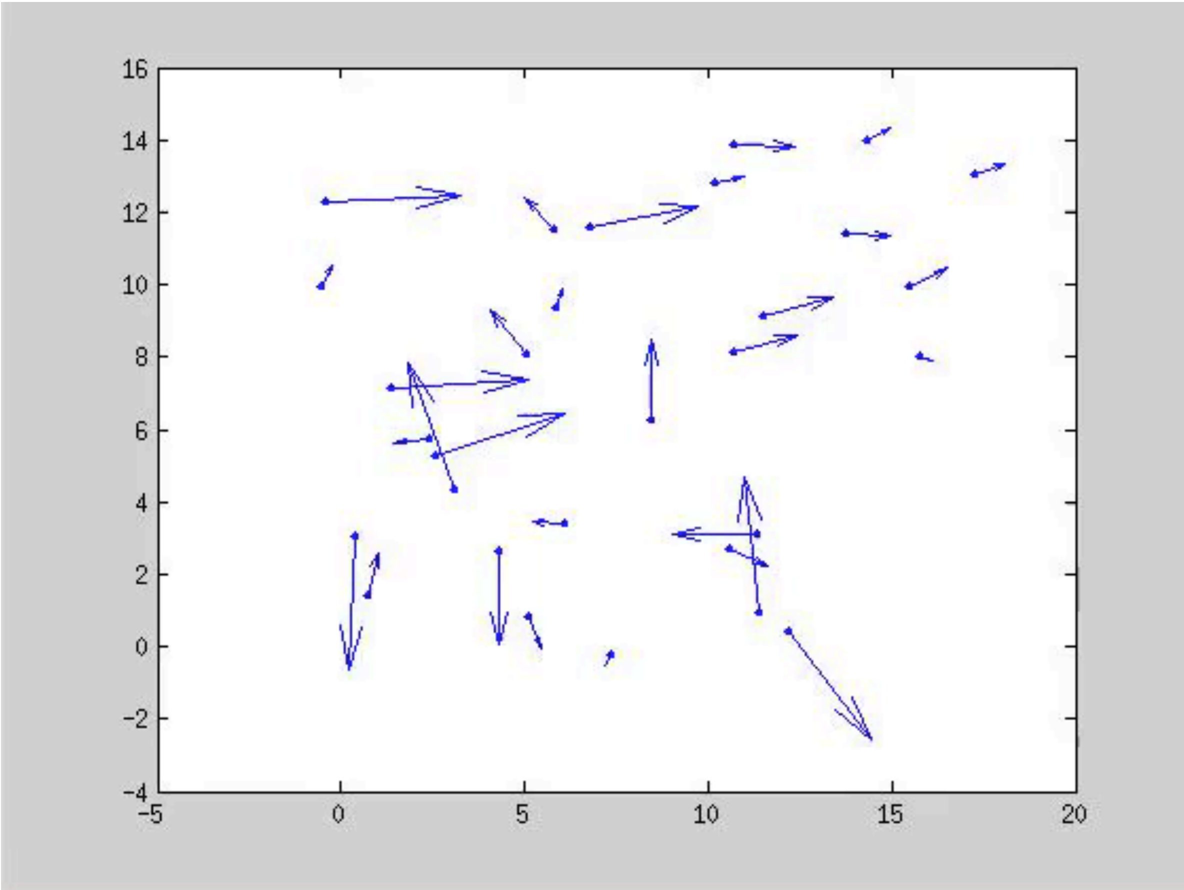
Nathan & Barbosa's model (2008):

- Coalescing: seek proximity of nearest bird
- Gap-seeking: seek nearest position affording clear view
- Stationing rule: move to upwash of a leading bird

A Rule-based Attempt

Designed rules that generate a V-formation

- Drive birds towards the optimal upwash position w.r.t. the nearest bird in front; unsatisfactory solution



Outline

- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

The V-Formation Problem

Assume a generic 2-d dynamic model of a flock of birds

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$$v_i(t+1) = v_i(t) + a_i(t)$$

Goal: find best accelerations $a_i(t)$ at each time step that will finally lead to a V-formation.

This is a distributed control problem

What is a V-Formation?

We want a formation that achieves the optimum values for the following three fitness metrics:

1. Velocity Matching
2. Clear View
3. Upwash Benefit

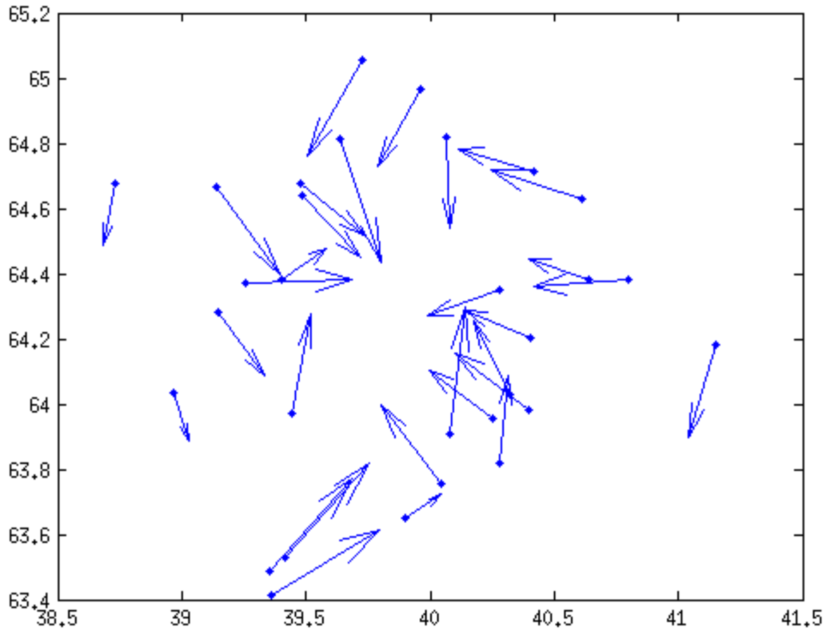
Velocity Matching (VM)

s = state of the n-birds = n positions, n velocities

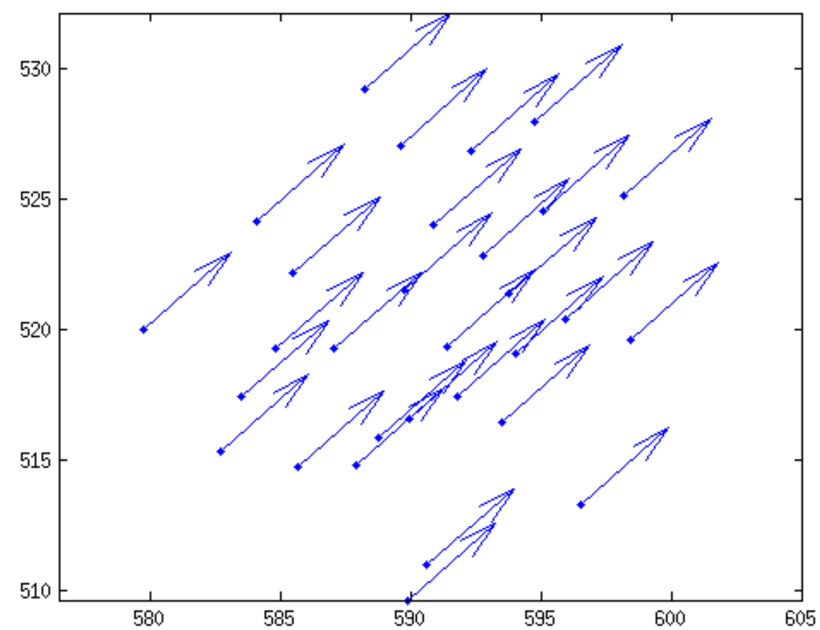
VM(s) = normalized sum of pairwise velocity difference

VM(s) = 0 if all birds have the same velocity

VM(s) increases as the velocities get more mismatched



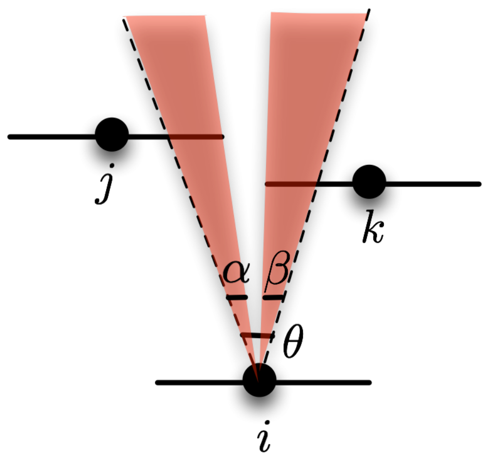
Velocity not matched



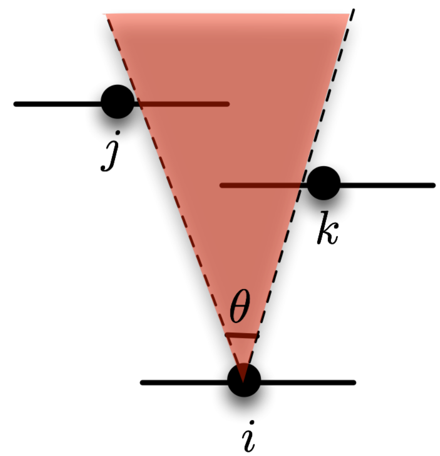
Velocity matched

Clear View (CV)

- Accumulate the percentage of the bird's view that is blocked
- CV(s) = 0 if every bird has a 100% clear view
- CV(s) increases as more of the view of any bird is blocked

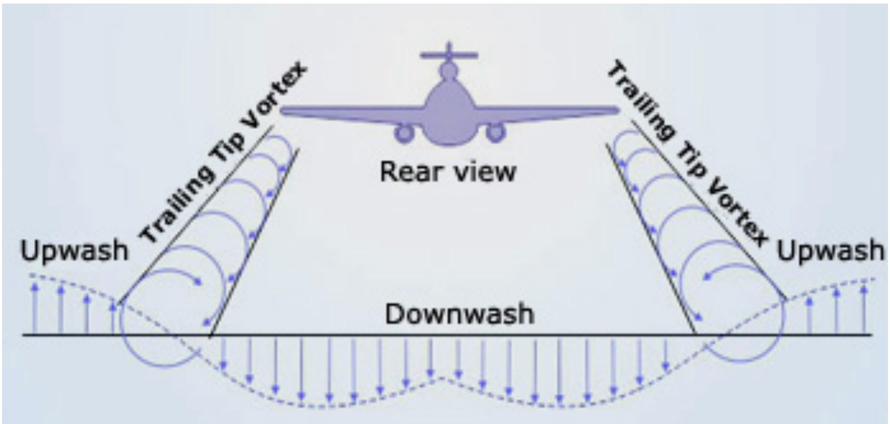


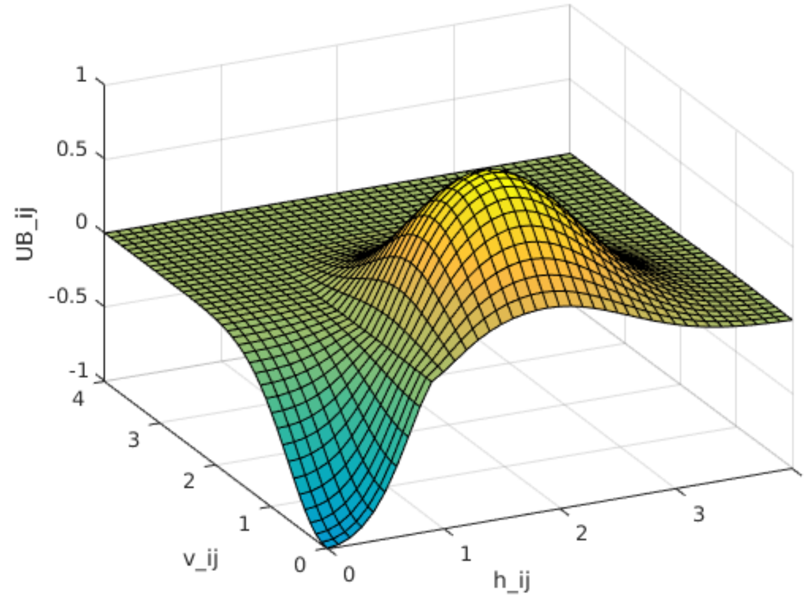
(a) i's view is partially blocked by j and k.
Clear view: $(\alpha+\beta)/\theta$



(b) i's view is completely blocked by j and k.
Clear view: 1

Upwash Benefit (UB)





- A Gaussian-like model of upwash and downwash
- UB(s) = sum of upwash benefit each bird gets from every other
- UB(s) = 1 if n-1 birds gets max possible UB benefit
- UB(s) increases as birds get lesser upwash benefit

Fitness Function

Fitness of a state is a sum-of-squares combination of VM, CV and UB

$$F(s) = (VM(s)\text{-}VM(s^*))^2 + (CV(s)\text{-}CV(s^*))^2 + (UB(s)\text{-}UB(s^*))^2$$

- state achieving optimal fitness value (i.e., a V-formation)

The V-Formation Problem

Assume a generic 2-d dynamic model of a flock of birds

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$$v_i(t+1) = v_i(t) + a_i(t)$$

Goal: find best accelerations $a_i(t)$ at each time step that will finally lead to a state with minimum F(s)

This is a distributed control problem

Outline

- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

Model Predictive Control (1)

At each time *t*, consider how the model will behave in the next T steps under different choices for the control inputs

- Use a model that represents the behavior of the plant

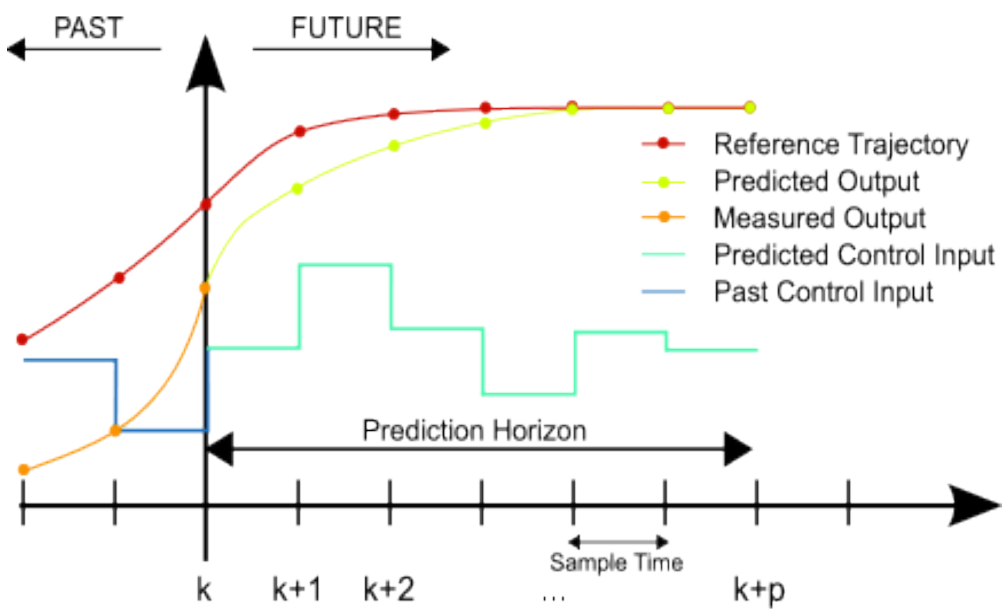
Use an optimization solver to find the best control inputs over this finite prediction horizon

Only apply the first optimal control action

Repeat at t+1

Model Predictive Control (2)

- At time *t+1*, update model state with new measurements of the plant.
- Repeat the optimization with new states.



A discrete MPC scheme (Wikipedia): horizon=p, current time=k

Model Predictive Control for V-Formation (1)

Bird *i* at time *t* solves the following optimization problem:

$$a^*_i(t), \ldots, a^*_i(t+T-1) = \operatorname{argmin}_{ai(t),\ldots,ai(t+T-1)} F(\, s_{Ni}(t+T-1)\,)$$

- $s_{Ni}(t)$: state at time *t* consisting of positions and velocities of bird 's neighbors
- Centralized control if Ni includes all birds
- F : fitness function.
- T: prediction horizon.

Model Predictive Control for V-Formation (2)

- Subject to constraints:
  - Model dynamics: State updates of each bird are governed by the model dynamics
  - Bounded velocities and accelerations: The velocities are upper-bounded by a constant, and the accelerations are upper-bounded by a factor of the velocities
- Finally, bird i uses the optimal acceleration for bird it found for time .

Particle Swarm Optimization (1)

The optimization problem is solved using PSO

- Inspired by social behavior of bird flocking or fish schooling.
- Initialize a population (swarm) of candidate solutions (particles) that move around in the search-space.
- Each particle keeps track of the best solution it has achieved so far (pbest) and the best solution obtained so far by any particle in the neighbors of the particle (gbest).

Particle Swarm Optimization (2)

- Repeatedly update the particle's velocity and position by:

  $$v_i(t+1) = w\, v_i(t) + c_1\, r_1\, (pbest_i - x_i(t)) + c_2\, r_2\, (gbest_i - x_i(t))$$

  $$x_i(t+1) = x_i(t) + v_i(t+1)$$

  where

  w : inertia weight

  r_1, r_2 : random numbers in (0, 1) sampled every iteration

  c_1, c_2 : constant learning factors

- Terminate when maximum iterations or desired fitness criteria is attained.

Distributed MPC Procedure

At every time step:

- Each bird looks at its neighbors

➢ Plays several scenarios in its head to find the best configuration that the neighborhood can reach in 3 steps

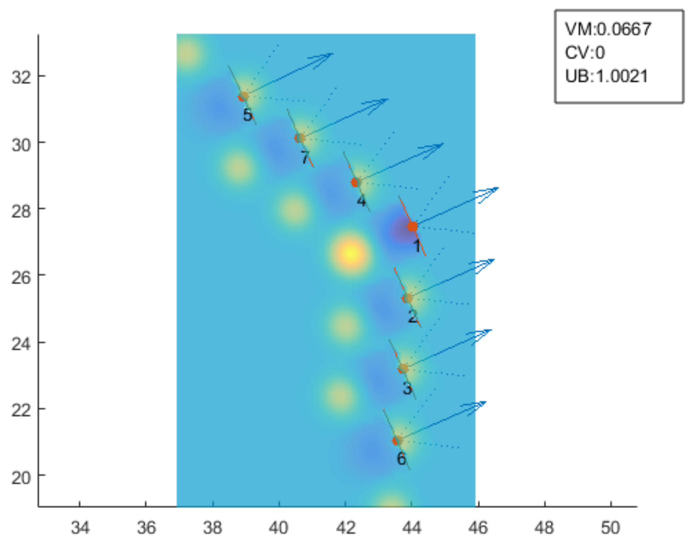
➢ The bird then applies the first move of that solution to update its position

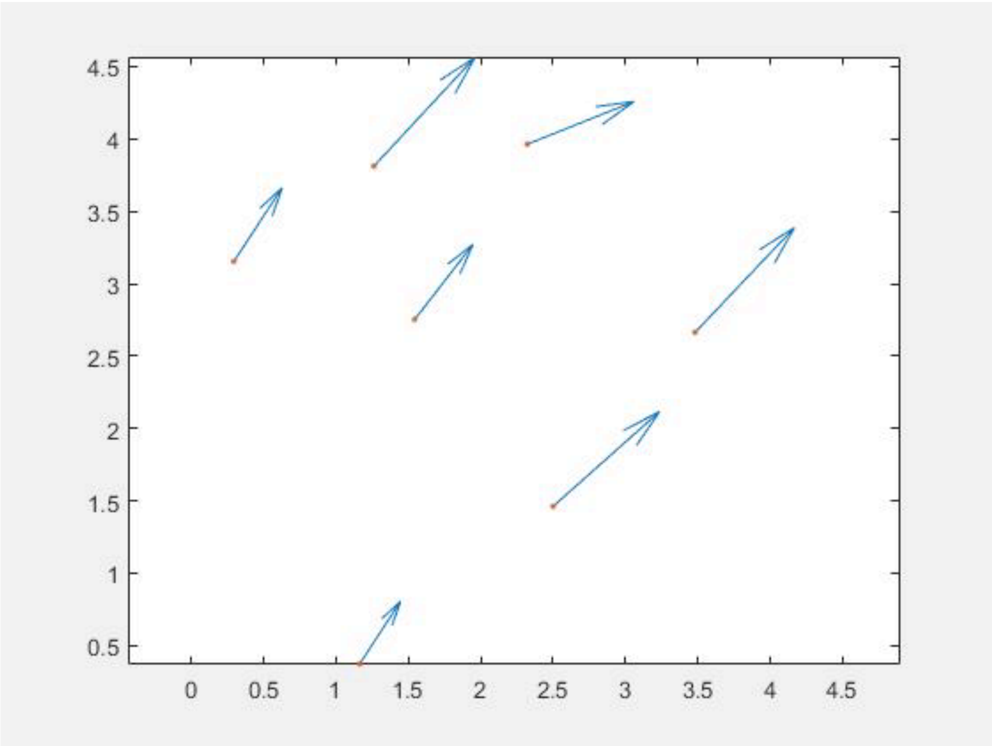
In the next time step, each bird updates its knowledge of the neighbors (positions and velocities), which may not be the same of what that bird predicted for its neighbors

Outline

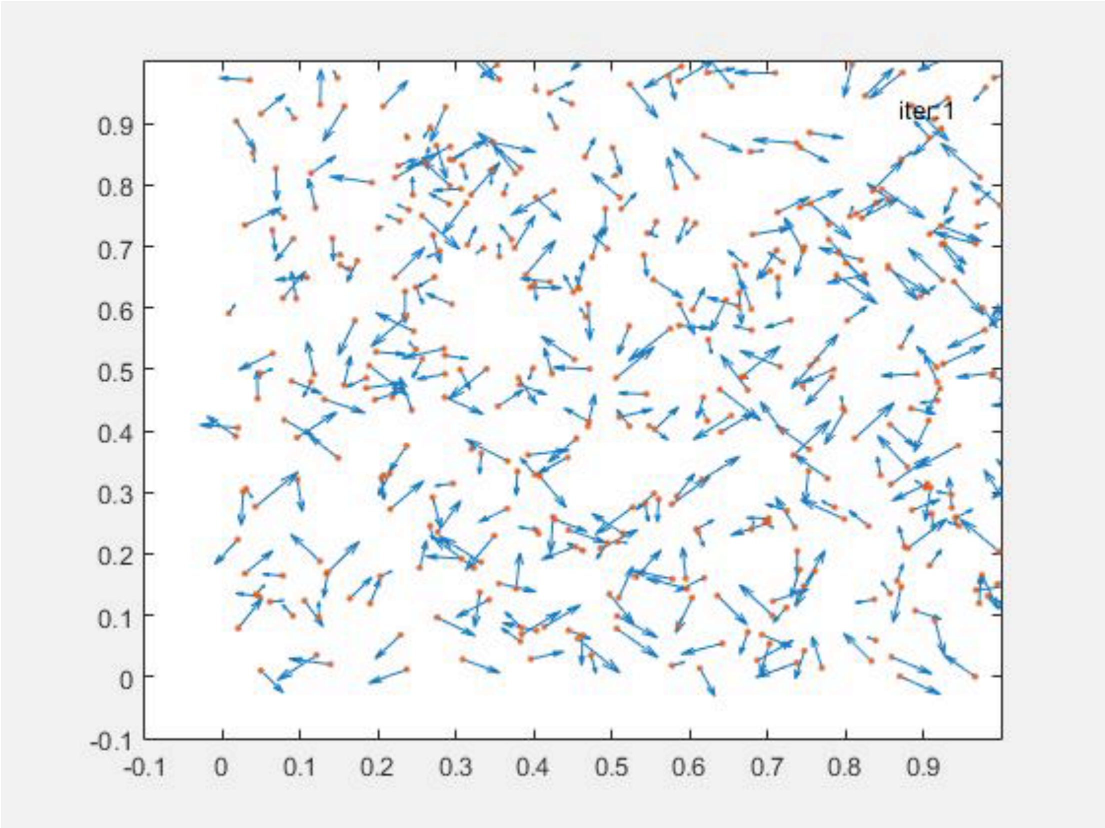
- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

Experimental Results (1)





Experimental Results (2)



Outline

- Introduction
- The V-Formation Problem
- Model Predictive Control for V-Formation
- Experimental Results
- Conclusions & Future Work

Conclusions

- Use distributed control instead of behavioral rules to achieve V-formation.
- Integrate MPC with PSO to solve the optimization problem.

Ongoing and Future Work

- Deploy the approach to actual plants (drones).
- Collision avoidance.
- Improve success rate of converging to V-formation.
- Use SMC to quantify the probability of success.
- Energy consumption and leader selection.

Thank you!