A Bird's-Eye View of Flocking
(version 2.0)

Scott A. Smolka
Department of Computer Science

Stony Brook University
Computer Science

Venti Anni di Informatica a Urbino

Sep. 13th, 2021

Joint work with

Usama Mehmood

Educative Inc.

Shouvik Roy

Research Assistant
Stony Brook Univ.

Radu Grosu

Univ. Professor
TU Wien

Scott Stoller

Professor
Stony Brook Univ.

Ashish Tiwari

Microsoft
Research

Agenda

- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

Agenda

- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

A Room with a View

Why Do Birds Flock?

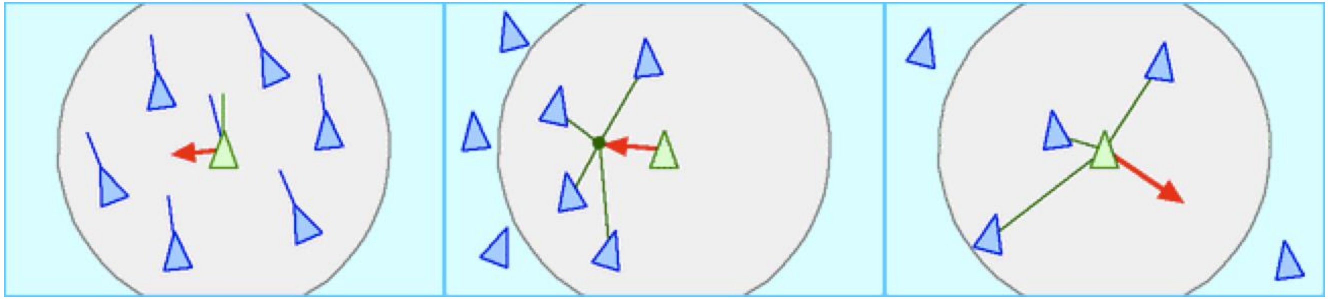
- Foraging: allows many birds to search for and take advantage of same food supply.
- Protection: a larger group of birds has a better chance of spotti[illegible]and also confusing or overwhelming it (see slide 8)
- Mating: males showing off thei[illegible]ge make themselves more visible to a greater number of f[illegible]
- W[illegible], bird flocks can share benefit of communal warmth to survive s[illegible] cold temperatures.
- Aerodynamics: V-formation!

“Birds that stay together tend to survive together.”

Drone Swarm

Extreme Flocking: Murmuration

Reynolds Flocking Model



Alignment Cohesion Separation

Reynolds Interaction Rules:

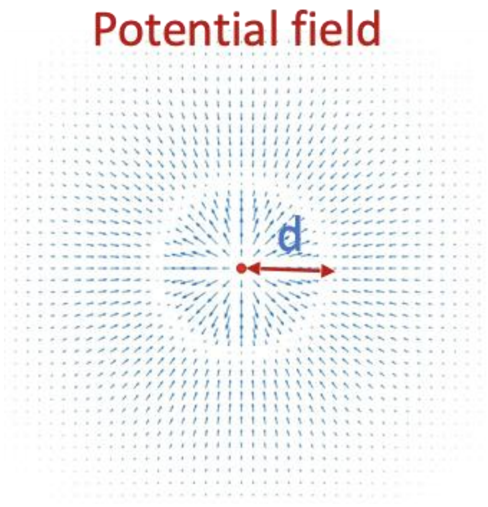
- Alignment: steer toward average heading of nearby flockmates.
- Cohesion: steer towards average position of nearby flockmates.
- Separation: steer to avoid crowding nearby flockmates.

Olfati-Saber Model

- Interaction between agents modeled as artificial potential fields.

- Potential for a pair of agents has its minimum at the desired inter-agent distance **d** of resulting **α-lattice**.

- An agent's acceleration based on
  - sum of the forces from all neighbors
  - velocity alignment term



Other α-Lattice based Models

- Centralized & Distributed MPC-based approaches; inspired by [Zhan & Li 2011, 2013]

- Use a cost function ***g*** that penalizes configurations in which inter-agent distance is not ***d***.

$$g(\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{E}(\boldsymbol{x})} \left\| x_{ji} - \frac{d \cdot x_{ji}}{\|x_{ji}\|} \right\|^2$$

Agenda

- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

Declarative Flocking

- Optimal Control used to define flocking controllers in centralized and distributed settings.
- Associated cost function has terms for cohesion and separation.
- No hard-coded rules as in Reynolds model.
- To generate agent accelerations, flocking controller seeks to minimize cost at every time-step.

Model Dynamics

The state of an agent i consists of its position x_i and velocity v_i. The state of a collection of n agents is given by:

$$s = \{x_i, v_i\}_{i=1}^{n}$$

The discrete-time equations of motion for agent i are:

$$x_i(t+1) = x_i(t) + dt \cdot v_i(t), \quad |v_i(t)| < \bar{v}$$
$$v_i(t+1) = v_i(t) + dt \cdot a_i(t), \quad |a_i(t)| < \bar{a}$$

where dt is the duration of a time-step.

Declarative Flocking Cost Function

$$J_i(t) = \underbrace{\frac{\omega_c}{|N_i|} \sum_{j \in N_i} \| x_{ij} \|^2}_{\text{Cohesion}} + \underbrace{\frac{\omega_s}{|N_i|} \sum_{j \in N_i} \frac{1}{\| x_{ij} \|^2}}_{\text{Separation}}$$

- $J_i(t)$: distributed cost function
- ω_c , ω_s: cohesion and separation weights
- $\| x_{ij} \|$: Euclidean distance between agents ***i*** and ***j***
- N_i : neighborhood of agent ***i***

As we show later in talk, DF cost function can easily be extended with terms for obstacle avoidance, leader following, predator avoidance, ...

Model Predictive Control

Goal: Find best accelerations $\boldsymbol{a_i(t)}$ at each time step that will lead to a flock formation.

- Develop a model of the plant **DONE!**
- At each time step $\boldsymbol{t}$

  Use the model and an optimization solver to determine control inputs that minimize cost function over a finite prediction horizon $\boldsymbol{T}$

  Only apply first optimal control action to the plant

Repeat at $\boldsymbol{t + 1}$ after updating model state with new measurements of the plant

Model Predictive Control (2)

At time t, we solve the following local optimization problem:

$$a_i^*(t|t), \dots, a_i^*(t+T-1|t) = \text{argmin}_{a_i(t|t),\dots,a_i(t+T-1|t)} \sum_{k=1}^{T} J_i(t+k-1)$$

$J_i(t)$: cost for agent i at time t

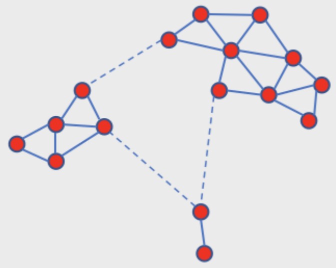
T : prediction horizon

$a_i^*(t'|t)$: optimal accel. for agent i at time t' as computed at time t

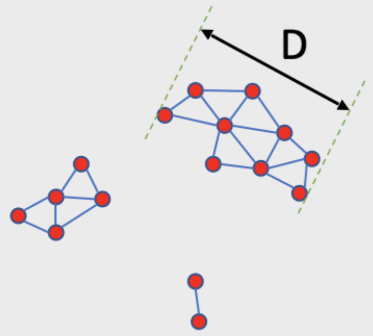
We can now set $a_i(t)$ to $a_i^*(t|t)$

Performance Measures

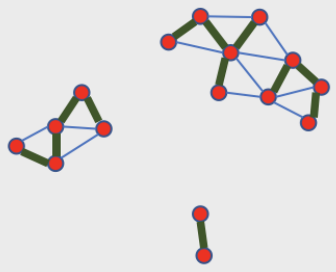
Connected Components
|CC|



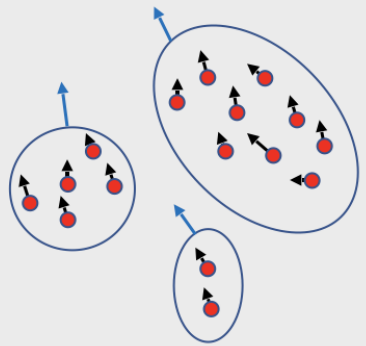
Maximum Diameter
D



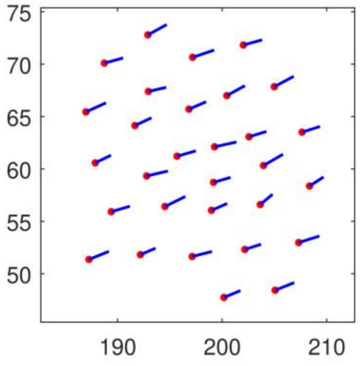
Irregularity
I



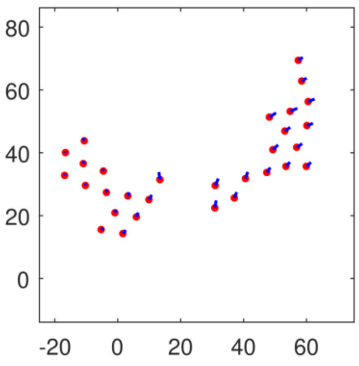
Velocity Convergence
VC



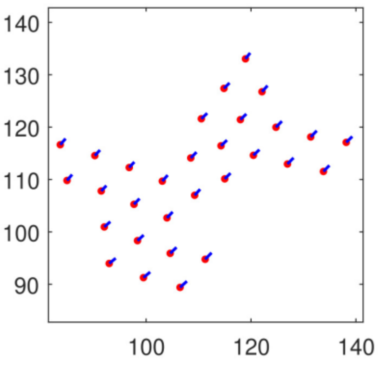
Flocking Model Formations (30 agents)



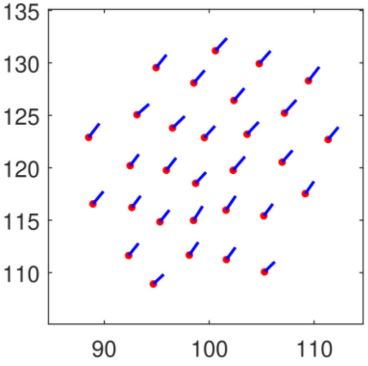
Reynolds



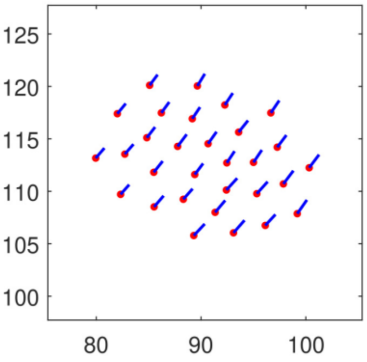
Lattice Centralized



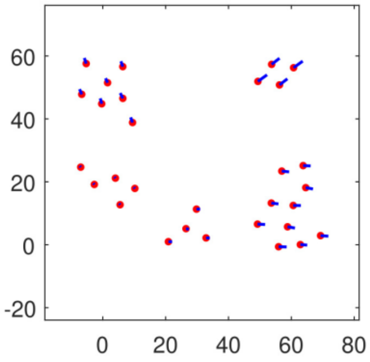
Lattice Distributed



DF Centralized

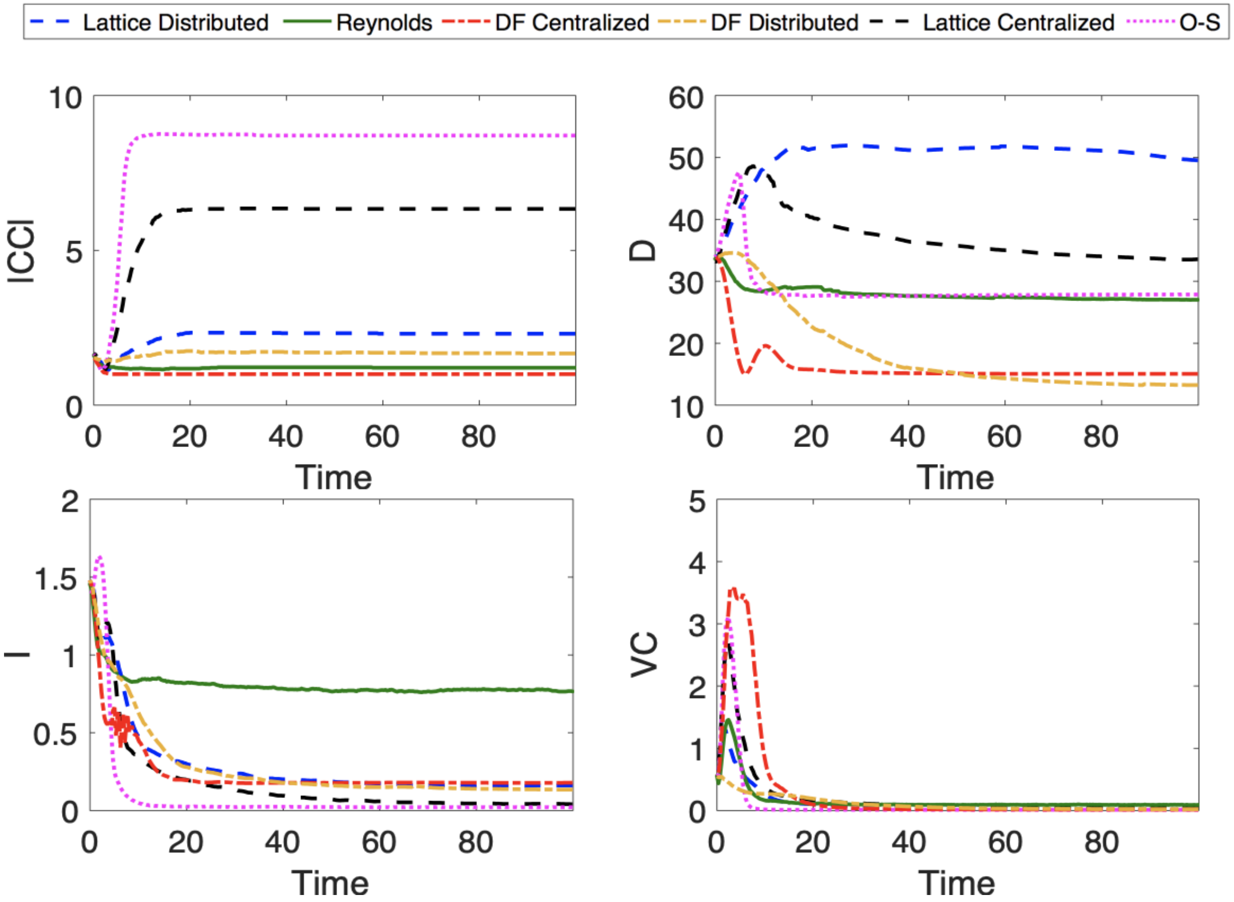


DF Distributed



Olfati-Saber

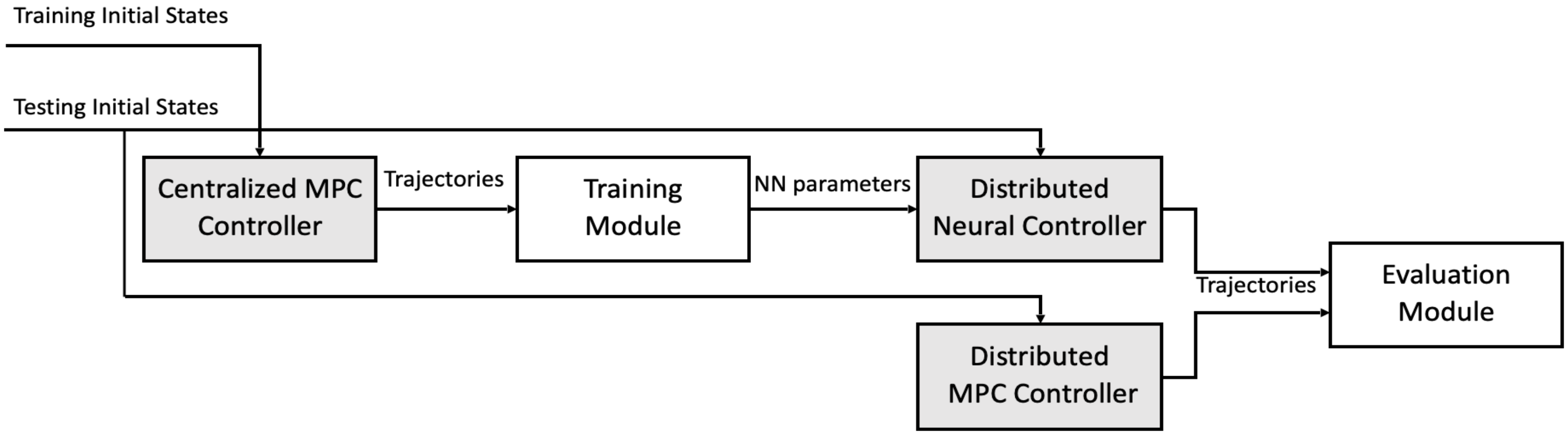
Declarative Flocking Results



Agenda

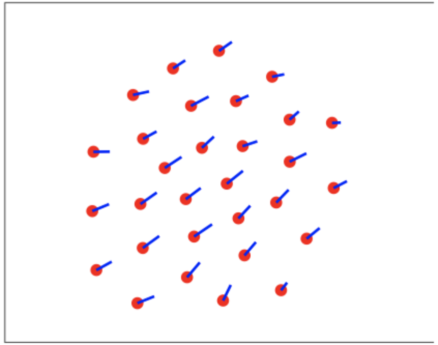
- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

Neural Flocking [FoSSaCS 2020]

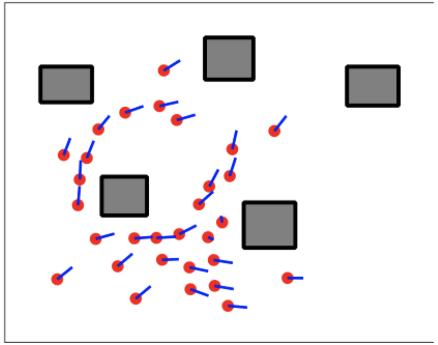


- New approach to flocking using Supervised Learning.
- Centralized MPC controller provides labeled training data to learning agent: a symmetric distributed neural controller (DNC).

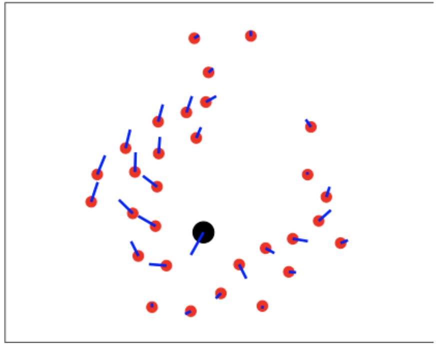
NF Control Objectives



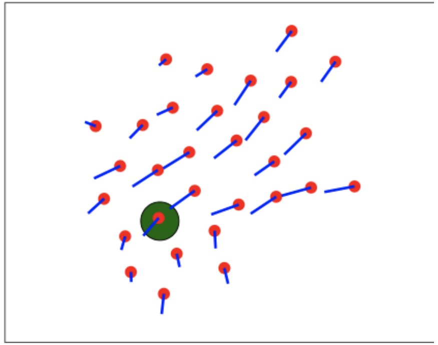
Basic Flocking (BF)



Obstacle Avoidance (OA)

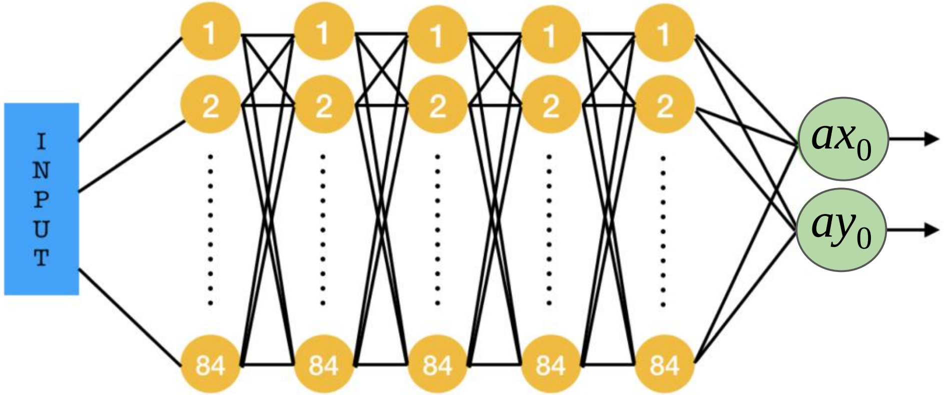


Predator Avoidance (PA)



Target Seeking (TS)

DNN for Neural Flocking



DNN with 5 hidden layers, each with 84 neurons

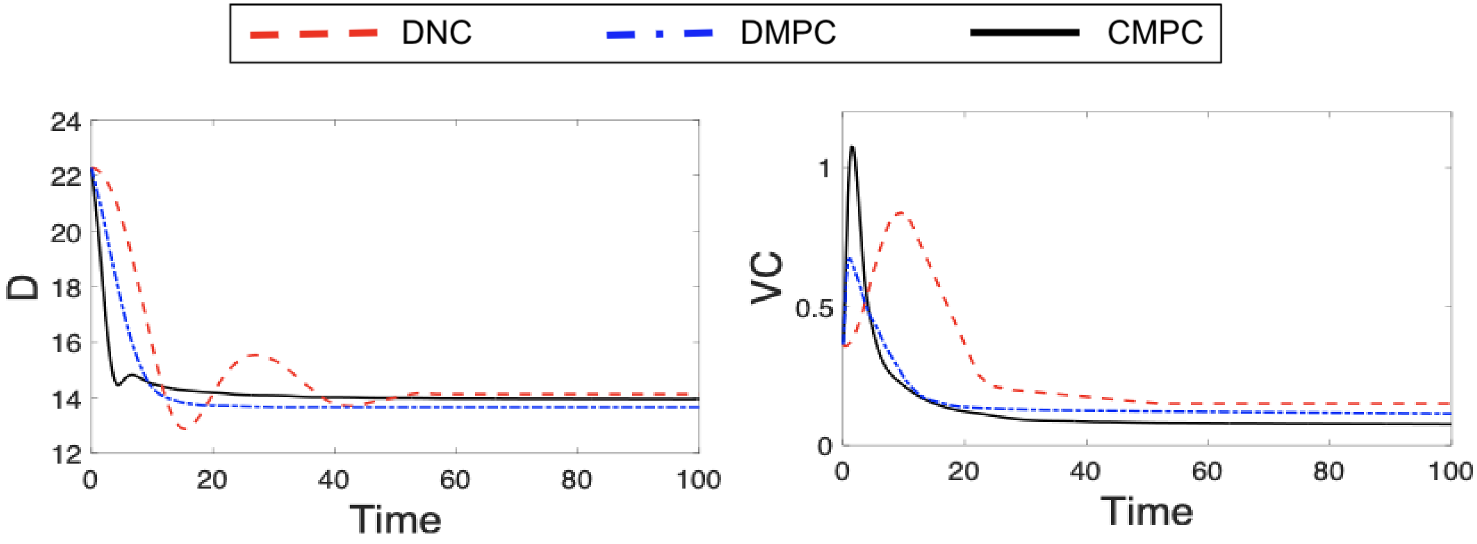
- x_i and v_i are position and velocity of agent ***i***
- ***o*** and ***g*** are closest obstacle and target location
- x_{pred} and v_{pred} are position and velocity of predator
- DNN outputs accelerations ax_0 and ay_0 for agent **0**

DNN Inputs		
BF	**OA + TS**	**PA**
x_o	x_o	x_o
v_o	v_o	v_o
.	o_o	.
.	.	.
.	.	x_{14}
x_{14}	x_{14}	v_{14}
v_{14}	v_{14}	x_{pred}
	o_{14}	v_{pred}
	g	

DNC Training Parameters

- Training data: CMPC trajectory data for 15 agents:
  - Initial position distribution = $[-15, 15]^2$
  - Initial velocity distribution = $[0, 1]^2$
- # training samples = 200 trajs $\times$ 1,000 time-steps $\times$ 15 agents = 3M
- Batch size = 2,000 , # training epochs = 1,000
- Optimizer = Adam
- # training parameters = 33,854 (for Basic Flocking)
- Training software = Keras (runs on top of TensorFlow)

Neural Flocking Results (15 agents)



Average (over 100 runs) *execution times* per time-step:

Centralized MPC: 80.6 ms / agent

Distributed MPC: 58 ms / agent

DNC: 1.6 ms / agent

DNC 36x faster than DMPC!

Neural Flocking Generalization

Agents	Avg. Conv. Diameter	Conv. Rate (%)	Avg. Conv. Time	ICR
15	14.13	100	52.15	0
20	16.45	97	58.76	0
25	19.81	94	64.11	0
30	23.24	92	72.08	0
35	30.57	86	83.84	0.008
40	38.66	81	95.32	0.019

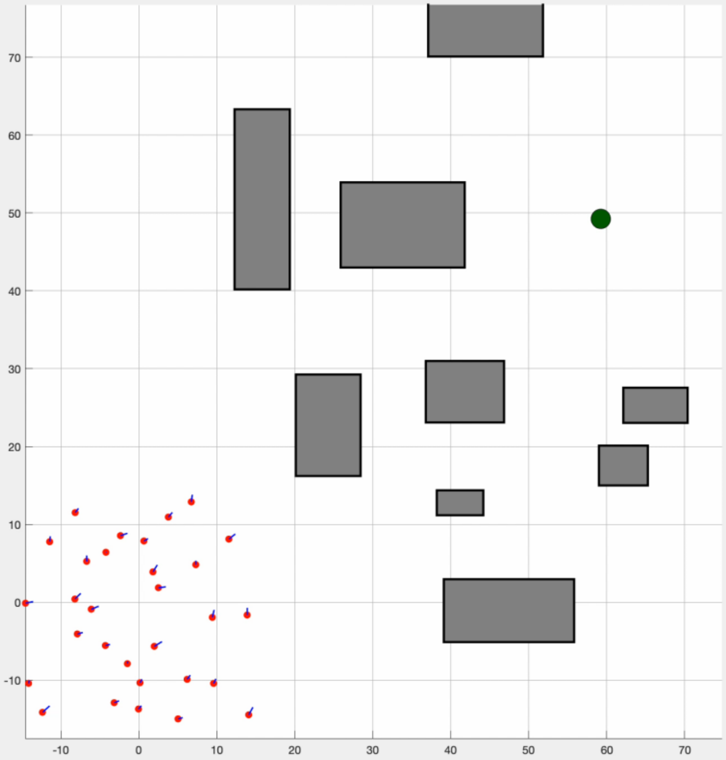
Generalization Performance for Basic Flocking

	OA		PA	
Agents	ICR	OCR	ICR	PCR
15	0	0	0	0
20	0	0	0	0
25	0	0	0	0
30	0	0	0	0
35	0.011	0.009	0.013	0.010
40	0.021	0.018	0.029	0.023

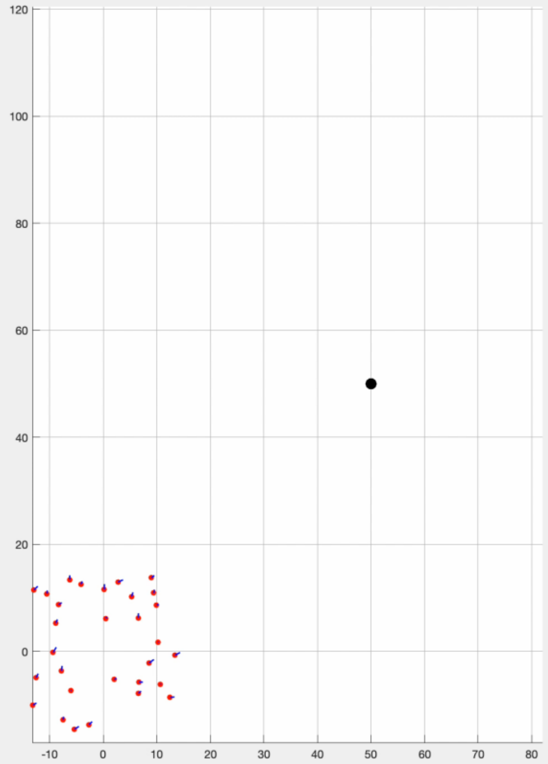
Generalization Performance for Obstacle Avoid. & Predator Avoid.

- ICR: Inter-agent collision rate
- OCR: Obstacle collision rate
- PCR: Predator collision rate

Obstacle Avoidance Video



Predator Avoidance Video

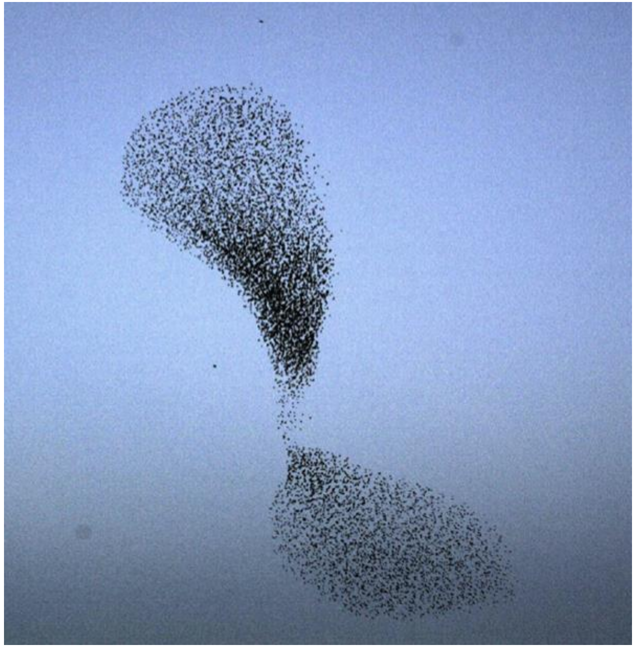


Agenda

- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

Flocking Maneuvers

- Flocking maneuvers employed by starlings for a spontaneous change in travel direction.
- Such turns initiated by a few individual birds & rapidly propagate throughout the flock.
- Propagation follows a linear dispersion law with negligible attenuation.



Flocking Maneuvers [ACC 2021]

- Distributed MPC with Acceleration-Weighted Neighborhooding (AWN-DMPC) used to synthesize a controller for **high-speed** flocking maneuvers.
- AWN exploits imbalance in agent accelerations during a turning maneuver to ensure actively turning agents are prioritized.
- Only a few agents (initiators) are aware of maneuver objective. AWN-DMPC controller ensures this local information is propagated throughout the flock in a scale-free manner.

AWN-DMPC Cost Function

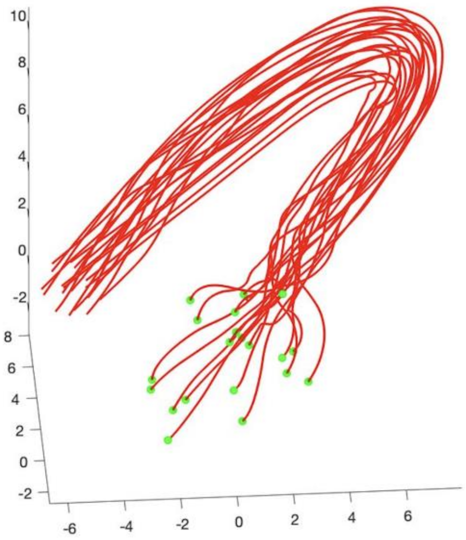
$$J_i(t) = \frac{\omega_c}{|N_i|} \sum_{j \in N_i} \| x_{ij} \|^2 + \frac{\omega_s}{|N_i|} \sum_{j \in N_i} \frac{1}{\| x_{ij} \|^2} + \sum_{j \in N_i} \gamma_{ij} \cdot \| v_{ij} \|$$

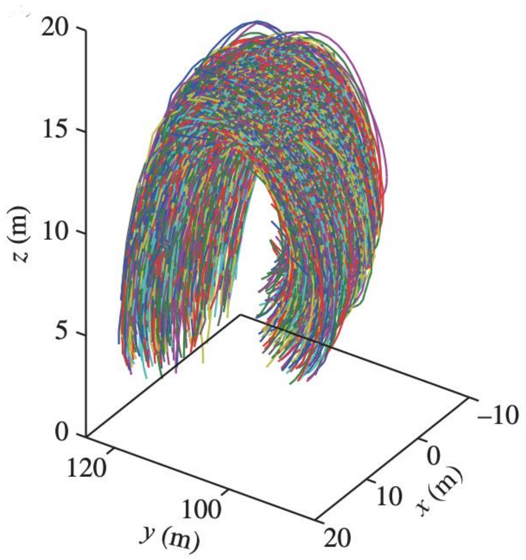
where γ_{ij} is the preferential weight term

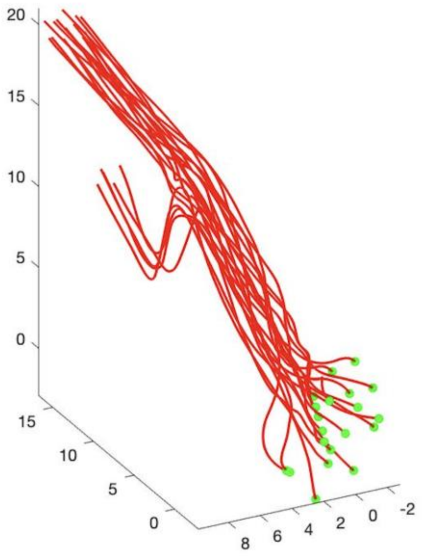
$$\gamma_{ij} = \frac{e^{\eta \cdot \Delta v_j(k)}}{\sum_{j \in N_i} e^{\eta \cdot \Delta v_j(k)}}$$

- $\Delta v_j(k) = \| v_j(k) - v_j(k-1) \|$ is change in agent j 's velocity between the two consecutive time-steps.
- η is a large constant used to stabilize the softmax function.

AWN-DMPC Results





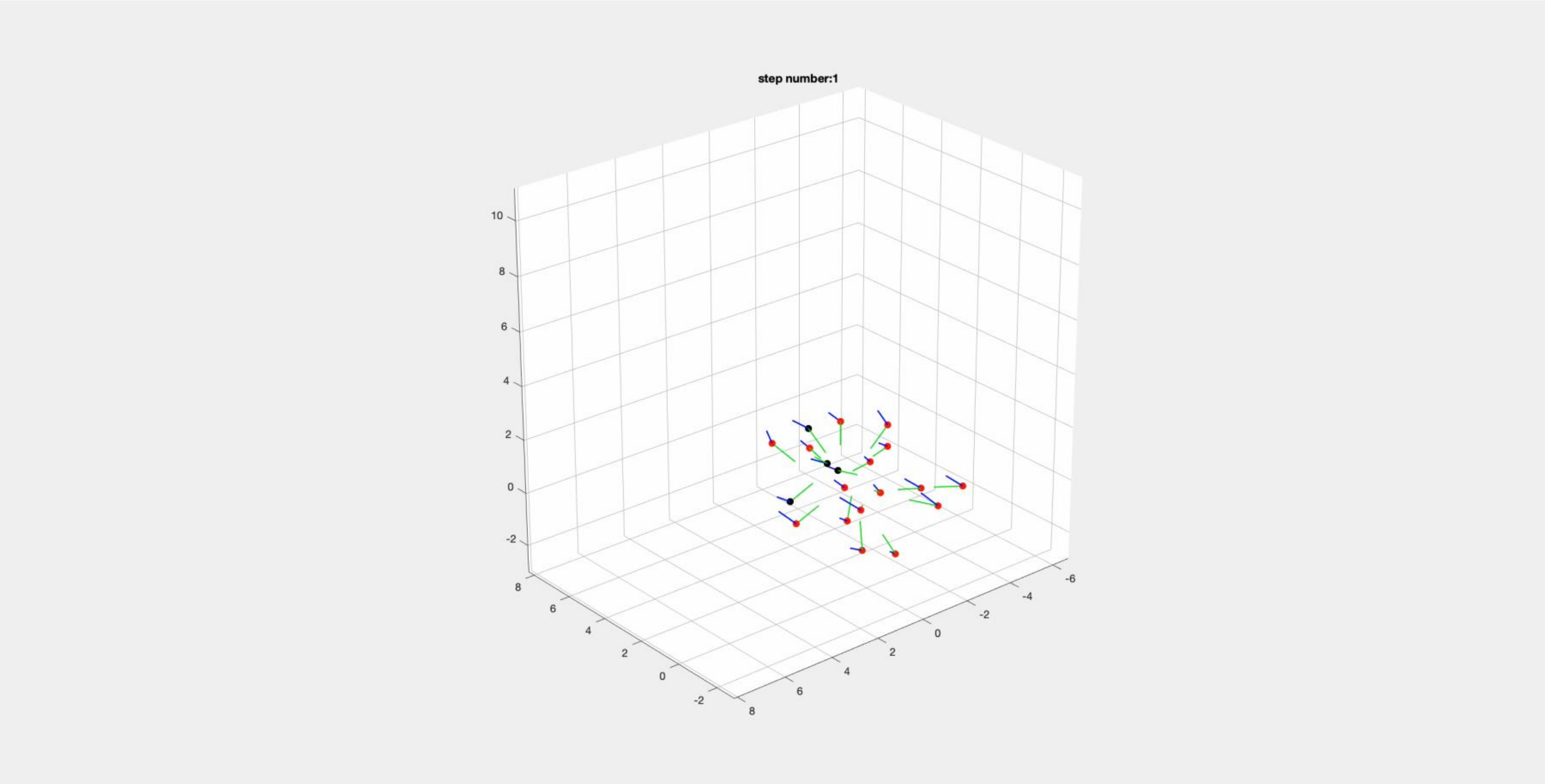


Trajectories with AWN
(20 agents, 170° turn, 4 initiators)

Starling Trajectories
[Attanasi et al. 2015]

Trajectories without AWN

Flock Maneuver using AWN



Agenda

- Introduction
- Declarative Flocking
- Neural Flocking
- Flocking Maneuvers
- Experimental Results + 1
- Conclusions & Future Work

Conclusions

- Declarative Flocking optimal-control framework (cost functions).
- Centralized / Distributed MPC flocking controllers.
- Extended Declarative Flocking to various flight control objectives.
- Distributed Neural Controller for real-time flocking + generalization.
- Distributed MPC+AWN controller for high-speed flocking maneuvers.
- Experimental validation using SPC and Crazyflie drones.

Ongoing & Future Work

- WebGL for high-speed, large-scale flocking on your phone!
- Outdoor demonstrations
  - Crazyflies 2.1 (only 27 grams & fits in palm of your hand)

- Collision Avoidance can be established using our Distributed Simplex Architecture for runtime assurance of distributed systems.
- Learn explainable, biophysical Neural Network for flocking (Liquid-Time Constant NNs)

Research supported in part by NSF awards CPS-144683, CCF-1954837, CCF-191822, DCL-2040599 and ONR award N00014-20-1-2751.

https://www3.cs.stonybrook.edu/~sas/