

Blind Spots and Flocking Bots: Decentralized Control for Heterogeneous Robot Teams

Zixuan “Jack” Lin

May 8, 2025

Abstract

Flocking control enables multi-robot systems to exhibit cohesive, collision-free behaviors inspired by biological collectives. While classical frameworks, such as Olfati-Saber’s gradient-based method, provide strong stability guarantees in homogeneous agent groups, their extensions to heterogeneous systems and goal-driven tasks remain less explored. In this work, I extend Olfati-Saber’s potential-based flocking algorithm to heterogeneous multi-robot systems by introducing type-dependent desired distances and sensing ranges, enabling differentiated interaction rules across diverse agents. I also integrate naturalistic motion models on SE(2) and remove the artificial global alignment force of the γ -agent, instead relying on enhanced local interactions and visibility-limited neighbor selection to induce flocking. Our simulation studies investigate performance across free flocking, goal convergence, predator-prey pursuit, and collective foraging scenarios. I evaluate the system using classical flocking metrics such as cohesion, connectivity, and velocity alignment, and task-specific metrics like survival rate and food collection efficiency. Results highlight the flexibility, robustness, and scalability of the proposed method, while exposing inherent trade-offs between interaction range, local perception, and group coordination.

1 Introduction

The collective motion of animals—seen in bird flocks, fish schools, and insect swarms—offers a compelling blueprint for decentralized coordination in multi-robot systems. These biological groups achieve robust, scalable, and cohesive behavior through simple local interactions, making them ideal models for distributed control in scenarios like exploration, surveillance, and resource gathering.

This dual inspiration has led to two major threads of research: one aimed at simulating flocking behavior for animation and graphics [1], and another focused on understanding and replicating such behavior through mathematical models and control theory [2]. While early

Code and simulations available at: <https://github.com/JAAACKL/swarm-flocking>

work often assumed homogeneous agents and relied on idealized sensing or centralized coordination aids, real-world multi-robot systems must contend with heterogeneity, occlusion, and limited local perception.

In this work, I present a decentralized flocking framework designed to operate in more realistic settings. The approach supports heterogeneous agents with distinct interaction rules, simulates local sensing with visibility constraints, and eliminates artificial global alignment mechanisms. I evaluate the method across tasks ranging from free flocking to predator-prey pursuit and collective foraging, analyzing both theoretical flocking metrics and task-specific outcomes.

This paper contributes:

- An extension of Olfati-Saber’s flocking model using type-specific interaction rules.
- A visibility-aware neighbor selection rule that improves naturalism and scalability.
- A framework for simulating biologically inspired behaviors across multiple scenarios.

2 Related Work

2.1 Flocking Algorithms

Nearly 40 years ago, in the work *Flocks, Herds, and Schools: A Distributed Behavioral Model* [1], Craig Reynolds introduced a groundbreaking approach to simulate collective motion—such as flocks of birds or schools of fish—using decentralized local rules rather than centralized control or scripted paths. Each agent, or “boid,” operates as an autonomous actor making decisions based on its local environment.

The core of the model is built on three simple behavioral rules:

1. *Separation* – avoid crowding nearby agents,
2. *Alignment* – match velocity with neighbors,
3. *Cohesion* – steer toward the average position of nearby flockmates.

These rules collectively give rise to lifelike, emergent group dynamics. The influential work lays the foundation for flocking simulation and produces life-like animations. However, the work primarily focuses on producing the animation and the algorithm is largely heuristics based with intricate hand-tuned rules and priorities. The paper also does not discuss the mathematically implications of the system.

Olfati-Saber’s 2006 paper [2] builds upon Reynold’s boid algorithm and provides a rigorous and comprehensive framework for distributed flocking in multi-agent systems, transforming Reynolds’ heuristic rules into mathematically grounded control algorithms. The paper introduces three distributed control laws that formalize cohesion, separation, and alignment through potential fields and consensus-based velocity alignment, enabling agents with double-integrator dynamics to achieve coordinated group behaviors. The algorithm proposes α -agent with objective to form equal-distant lattice with neighboring α -agents and

virtual β and γ -agent serving as obstacles and moving rendezvous point. A major contribution of the paper is the use of smooth collective potential functions to enforce inter-agent spacing, which promotes the formation of structured spatial configurations called α -lattices, which is an organization such that all agents in a group are equally spaced at a desired distance. The paper also presents a formal definition of flocking, called ϵ -flocking, which includes conditions for connectivity, cohesion, velocity alignment, and energy minimization. Stability and convergence of the proposed algorithms are proven using Lyapunov methods and structural decomposition of the agent network in moving frames.

More recently, Ibuki et al. [3] proposed a distributed flocking control algorithm for rigid bodies in 3D, formulated on the Special Euclidean group SE(3). Their optimization-based framework unifies Reynolds' principles of cohesion, alignment, and separation using only relative pose information, making it fully distributed and compatible with onboard sensing. The method balances the inherently conflicting goals of cohesion and separation through optimization, enforces collision avoidance via control barrier functions, and ensures pose synchronization through pose-based constraints. While the approach offers improved handling of rigid-body dynamics compared to gradient-based methods like Olfati-Saber's, it is computationally intensive—demonstrated with only 30 agents versus 150 in prior work.

2.2 Multi-Agent Coordination for Goal-Oriented Tasks

In many real-world scenarios, multi-agent systems must perform tasks beyond maintaining cohesive formations, such as reaching goals, reshaping formations, covering areas, or evading threats.

Flocking often serves as a survival mechanism in nature, particularly in predator-prey contexts. Chen and Kolokolnikov [4] developed a simplified particle-based model that reproduces several swarm behaviors through short-range repulsion and long-range attraction among prey, and a decaying attraction from predator to prey. While analytically tractable, the model does not account for agent dynamics or sensing limitations.

Foraging is another essential multi-agent task, commonly studied through ant-inspired models that use stigmergy for indirect coordination via environmental cues [5]. However, flocking can also enhance foraging. Clark and Mangel [6] proposed a mathematical model in which birds improve foraging efficiency by flocking, as it increases shared access to sparse or unpredictable food sources. Their work highlights flocking as an evolutionarily stable strategy for distributed search, connecting cohesion and goal-driven coordination.

3 Methodology

3.1 Modifications to Olfati-Saber's algorithm

My implementation largely follows the formulation of Olfati-Saber's with a few tweaks.

3.1.1 Agent Dynamics

To make the simulation look more natural, I explore the effectiveness of using Olfati-Saber's algorithm in SE2. Each agent is modeled in a 2D space with position $x_i \in \mathbb{R}^2$, orientation

$\theta_i \in \mathbb{R}$, and speed $s_i \in \mathbb{R}_{\geq 0}$. The motion model is:

$$\dot{x}_i = s_i \cdot \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}, \quad \dot{\theta}_i = \omega_i, \quad \dot{s}_i = a_i, \quad (1)$$

where ω_i is the angular velocity (rate of change of heading) and a_i is the linear acceleration.

The control input is the angular velocity and acceleration:

$$u_i = \begin{bmatrix} \omega_i \\ a_i \end{bmatrix}. \quad (2)$$

Attraction and repulsion among agents will generate the desired acceleration in \mathbb{R}^2 and then be assigned to ω and a for control. Let $\tilde{\alpha} \in \mathbb{R}^2$ be desired acceleration in \mathbb{R}^2

$$\begin{aligned} \phi &= \begin{cases} \arctan 2(u_y, u_x), & \|\tilde{\alpha}\| > 0, \\ \theta, & \|\tilde{\alpha}\| = 0, \end{cases} \\ \hat{\mathbf{v}} &= \frac{\mathbf{v}}{v + \varepsilon}, \quad a_v = \mathbf{u} \cdot \hat{\mathbf{v}}, \\ a_m &= \|\mathbf{u}\|, \quad \omega = \min\left(\frac{a_m}{a_{\max}}, 1\right) 2\pi \end{aligned}$$

The final actuation and state are then further capped by the minimum and maximum velocity, acceleration, and angular velocity.

3.1.2 Heterogeneous Multi-Agent Flocking with Goal Convergence

Consider a population of N agents labeled $i = 1, \dots, N$. Each agent i belongs to one of K distinct types (or species), denoted by

$$t_i \in \{1, 2, \dots, K\}.$$

Interactions between any two agents i and j depend on their types t_i and t_j in two complementary ways:

1. **Desired inter-agent distance.** I collect target spacing preferences in a symmetric matrix

$$D = [D_{kl}] \in \mathbb{R}^{K \times K}, \quad D_{kl} = D_{lk},$$

where D_{kl} is the “ideal” distance that a type- k agent seeks to maintain from a type- l agent.

2. **Sensing (interaction) range.** We also introduce a symmetric sensing-range matrix

$$R = [R_{kl}] \in \mathbb{R}^{K \times K}, \quad R_{kl} = R_{lk},$$

where R_{kl} specifies the maximum distance at which a type- k agent can detect or respond to a type- l agent. In particular, two agents i, j only interact if

$$\|q_j - q_i\| \leq R_{t_i t_j}.$$

Thus, the pair (D, R) encodes both *how far* agents prefer to stay apart and *how far* they can sense one another. In multi-species flocking or swarming models, these matrices enter neighbor-selection rules (via R) and attraction–repulsion forces or potentials (via D).

In addition, I define agent type predator and food source for the simulation task predator-prey interaction and food source. The predator acts just like any α -agent except that its desired distance to the prey is 0, and prey evades predator by having the sensing range of to the predator the same as its desired distance to the predator so that the potential only creates repulsive force. Food sources are stationary agents which α -agents have 0 desired distance.

3.1.3 Simplified Potential Functions

In Olfati-Saber’s method, separate potential functions are employed to produce smooth control signal for α -agent to keep formation with other α -agents and β -agents from the obstacles. ϕ_β is formulated differently from ϕ_α but aims to produce similar results: a gradient that vanishes to 0 as agents move to the desired distance. I notice that having sensing distance equal to the desired distance also leads to a vanishing gradient at desired distance, therefore, I only use one potential function with different parameters to handle interactions between all pairs of agent types.

3.1.4 Removal of γ -Agent

In the paper by Olfati-Saber, the γ -agent represents a virtual agent whose role is to introduce a group objective, acting essentially as a moving rendezvous point for the flock. It guides the flock toward a desired trajectory or position, thereby preventing fragmentation and maintaining cohesion among the alpha agents α -agents, which are the actual flock members. The authors claim that without it, the swarm wonder-off separately instead of flock.

However, it is unnatural to assume that the school of fish or bird would share an imaginary goal or desired velocity in time and space. In nature, spiral shaped flocks occurs with animals revolving around a predator. In this case, agents only match locally but have different velocity when considering the entire swarm. The behavior cannot emerge if the swarm is tracking a moving rendezvous, which encourage total synchrony. The need for a γ -agent stem from short interaction range. In their experiments, the parameters of $d = 7$ and $r = 1.2d$ are used. Intuitively, the sensing range seems too small in that two agents slightly outside of the desired spacing fall outside of the range where the attractive force is effective. This might be the reason for the fragmentation.

In my experiment, increasing the sensing range in relation to the initial spacing of the agents significantly improves the effectiveness of flock formation. However, there are issues with increasing the interaction range relative to the desired distance. In the original paper, the author addresses that “the planarity of graphs induced by α -lattices imposes a restriction on maximum ratio of the interaction range to desired distance.” [2] If the ratio, $\kappa = r/d$, exceeds $\sqrt{3}$, the agents diagonal to each other in the hypercube begin to act on each other, making it impossible to converge to a formation where all agents are equally spaced. Due to the additional force, the agents converge to a stable state where the distance between agents is less than the desired distance d and the more agents are in the swarm, the closer they are.

To alleviate this problem, I only consider a fixed number of neighbors that are closest when computing the potential among α -agents. Doing so is more natural as it simulates the limited mental capacity of animals in swarms as well as obstruction of vision and therefore reduced sensing capability within dense swarms. At the same time, it also reduces the computational complexity. Ideally, the number of neighbors to consider should be greater or equal to 1 and less than 6 because 6 is the maximum number of neighbors an agent can have with equal spacing in a lattice.

3.1.5 Visibility simulation

Additionally, to replicate the effects of occlusion in dense swarm, I include calculation of visibility and obstruction. Agents occluded in line of sight by other agents are not considered neighbors regardless of distance. See figure 1 for illustration. This might add additional computational load to the simulation, in which the simulator knows the position of all nodes and computes the visibility, it is a natural behavior in real-world setting if the sensing are performed by individual agents through any directed sensor like camera, lidar, and radar.

Based on the reasons above, I do not use a γ -agent to induce flocking behavior unless the scenario involves reaching a collective destination or traveling in a predefined direction.

3.1.6 Overall Objective

The control policy aims to:

- Maintain cohesion (bounded pairwise distances)
- Align velocities across the group
- Respect heterogeneous preferred spacings via matrix D
- Ensure convergence to defined goals g_i
- Avoid static and dynamic obstacles (including adversaries)
- Enable task-specific behaviors such as foraging

These behaviors are achieved in a fully distributed fashion using only local sensing.

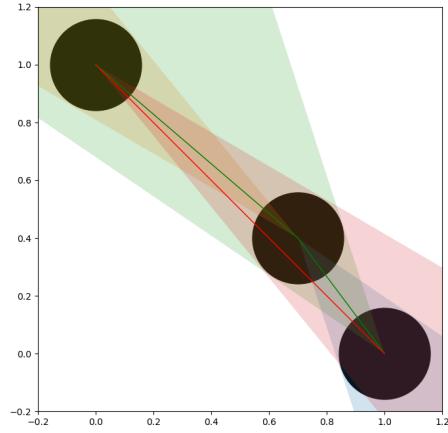


Figure 1: Occlusion among agents. The corner agents cannot see each other because they are blocked by the center agent, despite being within sensing range.

4 Evaluation and Results

4.1 Experimental Setup

All simulation experiments presented in this study share a consistent parameter configuration unless otherwise noted. The σ -norm parameter is fixed at $\epsilon = 0.1$. The scalar potential function $\phi(z)$ uses attraction and repulsion coefficients $a = 5$, $b = 15$, and smooth bump functions are employed with shape parameters $h = 0.2$ for $\phi_\alpha(z)$. The number of neighbors consider $n = 4$. The occlusion are computed so that exactly $\pi/3$ radians when the agents are d units apart, since it would suffice that the agent sees no further than its neighbor ring in a hexagonal lattice formation. Time steps range between 0.01 seconds, corresponding to update frequencies of 100 Hz. All agents update synchronously.

During simulations, each α -agent's heading angle reflects its velocity direction, and dynamic γ -agents are visually distinguished using a green dot. Every scenario has boundary half-plane obstacles on the four edges of the square simulation area and additional spherical obstacles are included based on the scenario.

The table below summarizes desired distance and sensing range parameters for pairwise interactions among the four types: prey (α -agent), predator, obstacle, and food. Each cell lists the desired distance (first) and sensing radius (second), separated by a comma.

	Prey	Predator	Obstacle	Food
Prey	20, 100	100, 100	50, 50	0, 100
Predator	0, 200	100, 100	50, 50	—

Table 1: Pairwise desired distance d and sensing radius r

To make the simulation interesting, different constraints are put on the dynamics of the predator and prey.

Agent Type	Accel. (unit/s ²)	Turn Rate (rad/s)	Vel. (unit/s)
Prey	2000	4π	200
Predator	1500	2π	300

Table 2: Dynamic constraints for prey and predator agents.

4.2 Flocking Evaluation Metrics

In the theoretical framework presented by Olfati-Saber, flocking behavior is evaluated through four primary metrics that define what is referred to as ϵ -flocking:

- **Relative Connectivity** $\mathcal{C}(t)$ — the normalized rank of the proximity net's Laplacian, capturing whether the group forms a connected graph.
- **Cohesion Radius** $\mathcal{R}(t)$ — the radius of the smallest ball that contains all agents, reflecting group cohesion.

- **Deviation Energy** $\tilde{E}(t)$ — a normalized potential energy indicating deviation from a quasi-lattice structure.
- **Velocity Mismatch** $\tilde{K}(t)$ — the average kinetic discrepancy among agents, representing alignment in motion.

Together, these criteria determine whether a system exhibits stable flocking behavior across time. In addition to the core metrics, I introduce scenario-specific performance metrics to evaluate flocking under environmental constraints and task-oriented conditions:

- **Goal Reaching Time** — for scenarios with static or dynamic obstacles, this measures the time taken by the flock to reach a specified target.
- **Survival Rate** — in predator-prey simulations, this metric records the percentage of prey agents that evade capture over time.
- **Feeding Rate** — in collective foraging tasks, this quantifies the rate at which agents successfully reach and consume food sources.

4.3 Discussion of Results

4.3.1 Convergence Experiment

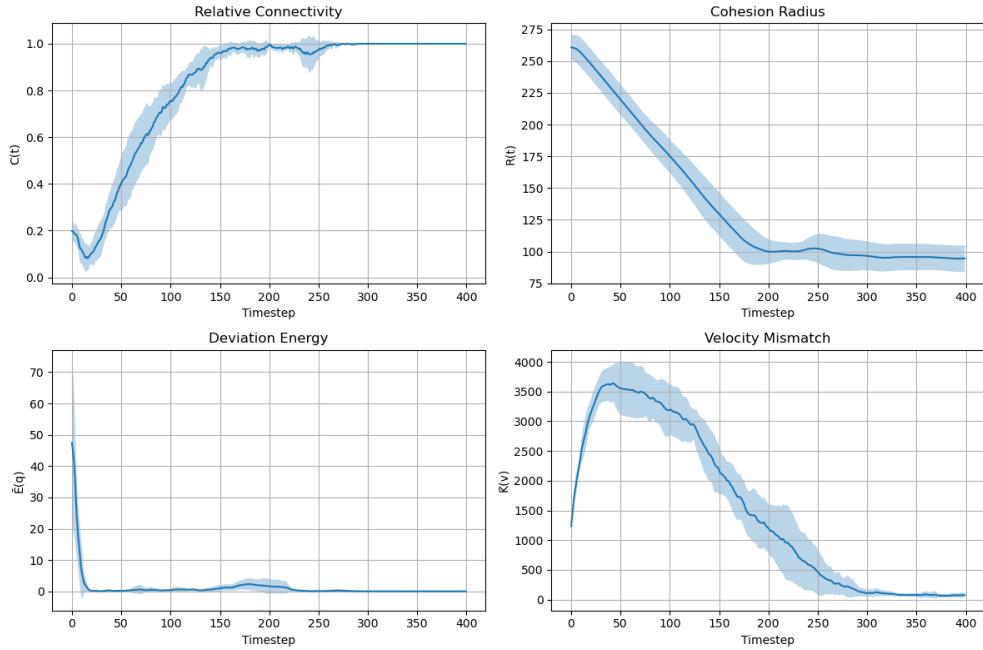


Figure 2: The graphs shows C , R , E , K of the group over time. The experiments are conducted 10 times and the shaded region shows the standard deviation. It is clear that the swarm converges to flocking formation every time. The convergence experiments are done in an unconstrained space with only α -agents.

4.3.2 Goal Reaching

Condition	Steps to Reach Goal (mean \pm std)
Non-flocking	903.5 \pm 54.87
Flocking	804.7 \pm 54.86

Table 3: Mean and standard deviation of steps to reach goal under different behaviors.

The following scenario-based experiments are setup by comparing effectiveness of reaching the objectives with and without flocking behavior. When there is no flocking, there is repulsive force between agents. Agents that are moving very slowly are essentially obstacles and they do not actively avoid other agents. All experiments are in a constraint space with half-plane obstacles at $x = \pm 250$ and $y = \pm 250$.

In the goal reaching scenario, there is a virtual *gamma*-agent at a positive $x = 200$ and $y = 0$, attracting the agents to move to the right while squeezing through 3 spherical obstacles. The objective is meet when all agent pass through a line define by $x = 100$.

From table 3, there is clearly an increase in efficiency with flocking strategy. From observation, flocking encourages order in narrow spaces. Although the individual agents might slow down to match the formation and velocity, the through put of the passage is high as the formation ensures one agent closely follows another, whereas when agents plan individually, conflicts in their paths create congestion that takes time to resolve, leaving periods when no agents are moving through the passages.

4.3.3 Predator and prey pursuit

In the predator-prey pursuit scenario, the prey tries to survive by avoiding the predator, while the predator tries to close the distance between itself and the prey. The predator's strategy is to track the center of the group that is in front of it. It does not track individual prey as I want to create the effect of predator confusion, which is when flocking swarm makes it hard for predator to single out individual prey.

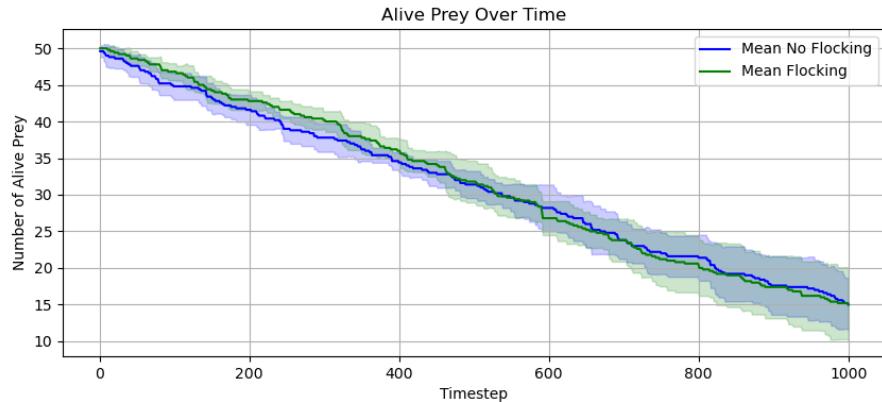


Figure 3: Comparison of survival over time of flocking versus individual

The experiment is conducted 5 times for each scenario. The mean surviving prey can be seen in figure 3. There does not seem to a difference between flocking and non-flocking strategy for prey. However, the experiment does not accurately reflect the effectiveness of flocking strategy as the predator's strategy is difficult to recreate. The best conclusion I draw from the experiment is that occlusion-aware flocking model creates natural looking predator-prey interaction at times.

4.3.4 Flocking for foraging

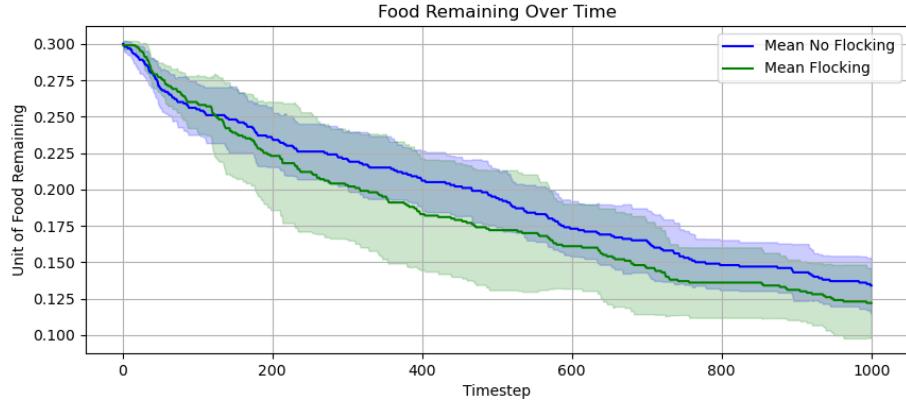


Figure 4: Comparison of food remaining over time of flocking versus individual foraging adjusted by average velocity of agents

In the foraging experiment, the simulation has 25 agents. 3 food sources each with 10 units of food is randomly scatter in the area. When an agent is hungry, it will be drawn to the food if it is within the sensing distance. Once an agent is fed, it will not be attracted to another food within 1000 iterations. Figure 4 shows the comparative results of flocking versus non-flocking foraging. The food remaining is adjusted by the average velocity of the agents, which is assumed to be proportional to the energy consumption. Adjustment is needed as flocking algorithm tend to lead to lower velocity, resulting an overall lower feeding rate without adjustment. After the adjustment, the result shows that flocking leads to better foraging efficiency.

4.4 Complexity and robustness

The simulation run in $O(n^2 \log n)$ due to having to find the nearest neighbor by sorting and compute visibility. The terms of interaction with obstacles, predators, and food are dropped since the number of α -agents dominate. However, since this is a distributed algorithm, the complexity is $O(n \log n)$ for individual robot with all-to-all communication overhead. If neighbors are determined through sensing, then the algorithm runs in $O(1)$ time with no communication overhead as a fixed number of neighbors are used to compute control.

The algorithm scales well with number of agents. Because all agents make decision based on local information, the flocking behavior remains consistent regardless of the number of

agents. New agents can be added to the group at any time and the swarm will absorb them seamlessly.

4.5 Limitations

The algorithm produces natural looking flocking behavior, but more comprehensive experiments and comparative analysis to real world swarms are needed to further validate its fidelity. As mentioned in the discussion, the algorithm has draw backs such as slowing down the swarm. Improvement such as only considering agents ahead as neighbors might alleviate the issue. Additionally, the potential and gradients by Olfati-Saber are in \mathbb{R}^2 while the control are done in $SE(3)$. I did not provide rigorous proof of soundness of my conversion and further work is needed to ensure the smoothness of the conversion.

5 Conclusion

This work presents a decentralized control framework for flocking in heterogeneous multi-agent systems, inspired by biological collectives and grounded in an extension of Olfati-Saber’s potential-based method. By introducing type-dependent interaction rules, modeling agent dynamics on $SE(2)$, and removing artificial global alignment mechanisms in favor of visibility-limited local coordination, the proposed system enables realistic and scalable flocking behaviors.

Simulation studies across multiple scenarios—free flocking, goal convergence, predator-prey pursuit, and collective foraging—demonstrate the effectiveness of the method in achieving cohesion, velocity alignment, and task-oriented objectives under real-world constraints such as occlusion and agent diversity. Evaluation through both classical flocking metrics and scenario-specific performance measures confirms the system’s adaptability and robustness.

The algorithm shows promising scalability and natural behavior, but also reveals trade-offs such as reduced throughput in tight formations and potential inefficiencies in pursuit scenarios. Future work will aim to improve directional awareness in neighbor selection, validate the control mappings more rigorously in $SE(2)$, and compare simulated behaviors with real-world biological and robotic swarms. Additional research may also explore decentralized learning mechanisms for adaptive flocking strategies in dynamic environments.

Acknowledgments

- Generative AI is used for polishing the language and format of the report and getting a quick start on the code to be iterated upon.

References

- [1] C. W. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. Anaheim, CA, USA: ACM, 1987, pp. 25–34.

- [2] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [3] T. Ibuki, S. Wilson, J. Yamauchi, M. Fujita, and M. Egerstedt, "Optimization-based distributed flocking control for multiple rigid bodies," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1891–1898, 2020.
- [4] Y. Chen and T. Kolokolnikov, "A minimal model of predator–swarm interactions," *Journal of The Royal Society Interface*, vol. 11, no. 94, p. 20131208, 2014. [Online]. Available: <https://doi.org/10.1098/rsif.2013.1208>
- [5] Q. Lu, G. M. Fricke, J. C. Erickson, and M. E. Moses, "Swarm foraging review: Closing the gap between proof and practice," *Current Robotics Reports*, vol. 1, no. 4, pp. 215–225, 2020. [Online]. Available: <https://doi.org/10.1007/s43154-020-00018-1>
- [6] C. W. Clark and M. Mangel, "Foraging and flocking strategies: Information in an uncertain environment," *The American Naturalist*, vol. 123, no. 5, pp. 626–641, 1984. [Online]. Available: <https://www.jstor.org/stable/2461242>

Appendix

Simulation Screenshots

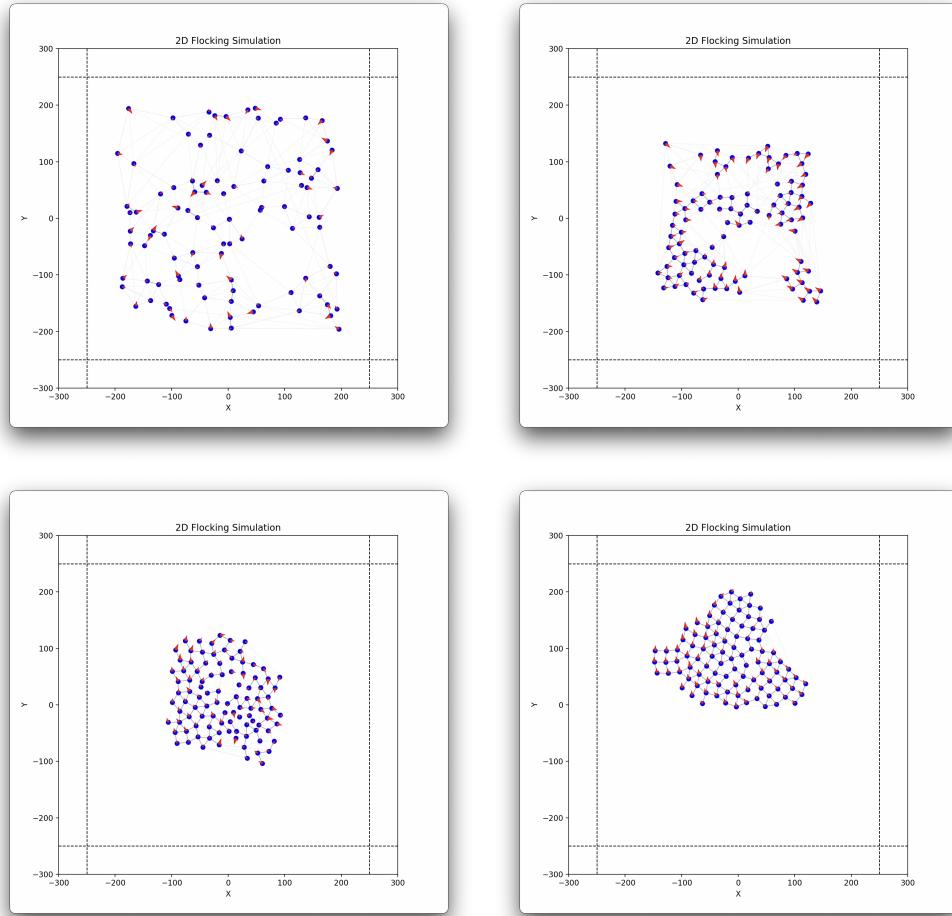


Figure 5: Swarm converging to a flocking state from a random start.

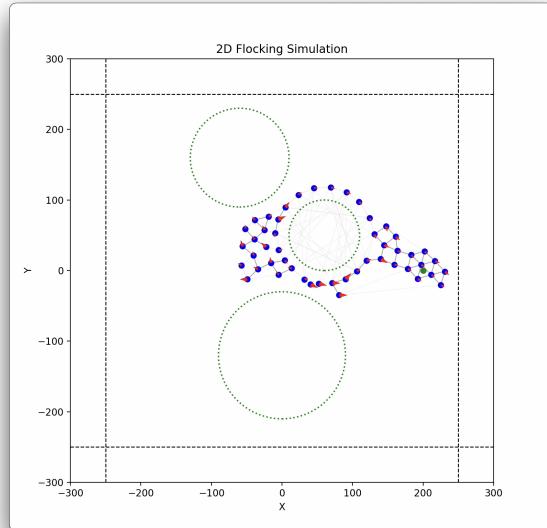


Figure 6: Flocking behavior while squeezing through narrow obstacles.

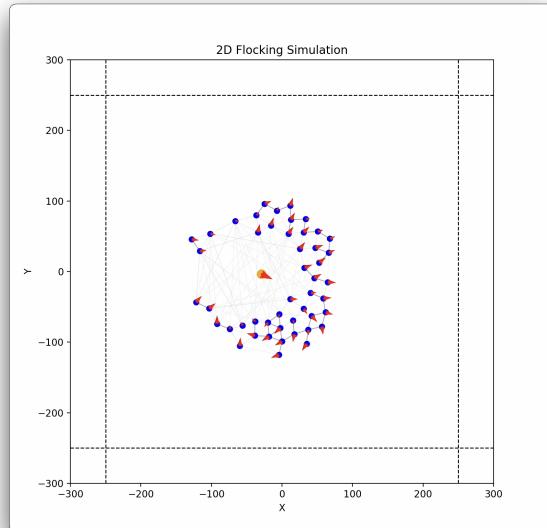


Figure 7: Flocking behavior in the presence of predator.

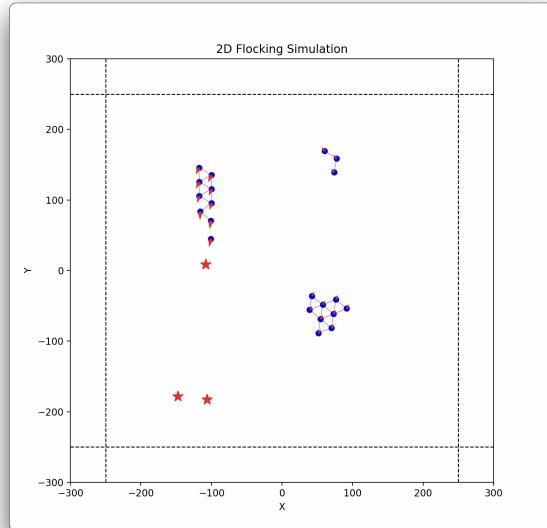


Figure 8: Flocking behavior for foraging.

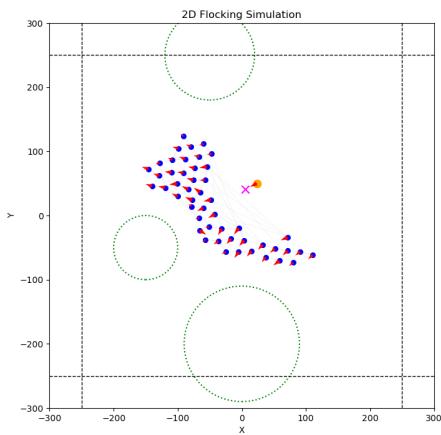


Figure 9: Flocking behavior in the presence of predator and obstacles.